

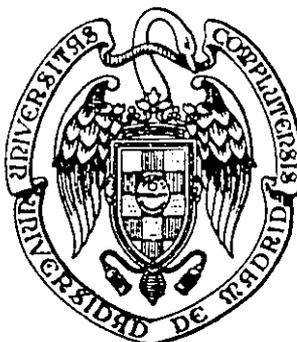
UNIVERSIDAD COMPLUTENSE DE MADRID

FACULTAD DE MATEMATICAS

Departamento de Estadística e Investigación Operativa

DE CORTES LAGRANGIANOS FUERTES EN LA RESOLUCION
DEL PROBLEMA DEL ORDENAMIENTO SECUENCIAL

M. Teresa Ortuño Sánchez



20.285

UNIVERSIDAD COMPLUTENSE DE MADRID
FACULTAD DE CIENCIAS MATEMATICAS
Departamento de Estadística e Investigación Operativa

SOBRE CORTES LAGRANGIANOS FUERTES EN LA RESOLUCION
DEL PROBLEMA DEL ORDENAMIENTO SECUENCIAL

M. Teresa Ortuño Sánchez



* 5 3 0 9 5 4 2 0 4 8 *

UNIVERSIDAD COMPLUTENSE

Memoria para optar al grado de Doctor
en Ciencias Matemáticas, realizada
bajo la dirección del Dr. D. Laureano
Escudero Bueno.

Madrid, Marzo 1995

D. LAUREANO ESCUDERO BUENO, PROFESOR DEL DEPARTAMENTO DE ESTADISTICA E INVESTIGACION OPERATIVA DE LA UNIVERSIDAD COMPLUTENSE DE MADRID.

CERTIFICO:

Que la presente memoria titulada:

SOBRE CORTES LAGRANGIANOS FUERTES EN LA RESOLUCION DEL PROBLEMA DEL ORDENAMIENTO SECUENCIAL

ha sido realizada bajo mi dirección por la Licenciada en Ciencias Matemáticas Dña. M. Teresa Ortuño Sánchez, y constituye su Tesis para optar al grado de Doctor en Ciencias Matemáticas.

Y para que conste, en cumplimiento de la legislación vigente y a los efectos que haya lugar, firmo el presente en Madrid a 28 de Marzo de 1995.

Laureano Escudero Bueno

A mis padres.

AGRADECIMIENTOS

No quisiera comenzar esta memoria sin expresar mi gratitud al profesor L. Escudero, bajo cuya dirección ha sido realizado este trabajo, por haber hecho posible la elaboración del mismo con sus orientaciones, enseñanzas y su inestimable ayuda. La posibilidad de trabajar a su lado ha sido una fuente de enriquecimiento constante durante los años de elaboración de esta memoria.

Agradezco también al Departamento de Estadística e Investigación Operativa, a sus directores P. Ibarrola y L. Pardo, y a todos sus miembros el haberme brindado la posibilidad de desarrollar mi trabajo en un ambiente realmente propicio. Mención especial merecen los que, además de excelentes compañeros, han sabido ser amigos y fuente constante de apoyo y estímulo a lo largo del camino, no siempre fácil, recorrido en estos años.

Por último, he de agradecer la gran comprensión y el apoyo moral recibido por parte de mis padres, mis hermanos, de los amigos, que han seguido paso a paso, sin entender a veces pero siempre desde el cariño, el trabajo que ha dado lugar a esta memoria.

Madrid, Marzo 1995

INDICE

PROLOGO

CAPITULO 1: DESCRIPCION DEL PROBLEMA Y PRELIMINARES

1.1	Introducción	1
1.2	Descripción del problema	2
1.3	Campos de aplicación	4
1.4	Notación	4
1.5	Definiciones básicas	6
1.6	Problemas de Optimización Combinatoria relacionados con el SOP	8
1.6.1	El problema de asignación	8
1.6.2	El problema del árbol generador de peso mínimo	9
1.6.3	El problema del camino mínimo	10
1.6.4	El problema del flujo máximo	10
1.6.5	El problema del viajante	12
1.7	Cotas superiores y cotas inferiores en optimización combinatoria	14
1.7.1	Complejidad	14
1.7.2	Procedimientos aproximativos (heurísticos)	18
1.7.3	Procedimientos de relajación	20

CAPITULO 2: METODOLOGIA BASADA EN EL PRIMAL

2.1	Introducción	23
2.2	Modelos	25
2.2.1	Introducción. Equivalencia y bondad de modelos	25
2.2.2	Diferentes modelos 0-1 equivalentes para el SOP con relaciones de precedencia	28
2.2.3	Modelo 0-1 para el SOP con relaciones de precedencia y acotaciones	33
2.3	Preproceso	37
2.3.1	Eliminación de arcos del conjunto A	37
2.3.2	Reducción del intervalo (r_1, d_1)	38
2.3.3	Eliminación de arcos del conjunto Π	39
2.3.4	Adición de arcos al conjunto Π	40
2.3.5	Transformación de desigualdades en igualdades	40
2.3.6	Test de infactibilidad	41

2.4 Obtención de cotas inferiores. SOP con relaciones de precedencia	42
2.4.1 Algoritmos de separación. Metodología	42
2.4.2 Identificación de condiciones de eliminación de subciclos	43
2.4.3 Identificación de condiciones que preservan relaciones de precedencia	49
2.4.4 Algoritmo de planos de corte. Líneas generales	52
2.5 Obtención de cotas inferiores. SOP con relaciones de precedencia y acotaciones	57
2.6 Algoritmo exacto	57

CAPITULO 3: METODOLOGIA BASADA EN EL DUAL

3.1 Introducción	61
3.1.1 Motivación	61
3.1.2 Desarrollo de la metodología	66
3.2 Modelo matemático	67
3.2.1 Modelización	67
3.2.2 Preproceso	68
3.3 Relajación Lagrangiana	68
3.3.1 Resultados fundamentales	68
3.3.2 Estimación de los multiplicadores de Lagrange. Método del subgradiente	71
3.3.3 Relajación Lagrangiana del SOP. Método de generación de filas y columnas para la estimación de multiplicadores	74
3.4 Algoritmo general de planos de corte utilizando metodología dual. Líneas generales	82

CAPITULO 4: CORTES VALIDOS PARA EL SOP

4.1 Introducción	85
4.2 Cortes válidos inducidos por las relaciones de precedencia	87
4.2.1 Obtención de cortes válidos	87
4.2.2 Identificación de restricciones violadas. Algoritmos de separación	91
4.2.3 Ejemplo de identificación de restricciones violadas	103

4.3 Cortes validos inducidos por las acotaciones superiores al peso acumulado de cada nudo	107
4.3.1 Obtención de cortes válidos	107
4.3.2 Identificación de permutaciones. Búsquedas locales	115
4.3.3 Obtención del peso de un camino permutación	117
CAPITULO 5: REFORZAMIENTO DE CORTES VALIDOS PARA EL SOP	
5.1 Introducción	119
5.2 Fundamentos teóricos	120
5.3 Reforzar restricciones tipo cubrimiento	129
5.3.1 Caso 1: Una restricción tipo cubrimiento como soporte	130
5.3.2 Caso 2: Una restricción tipo mochila como soporte	134
5.4 Reforzar restricciones tipo mochila	143
5.4.1 Caso 1: Una restricción tipo cubrimiento como soporte	143
5.4.2 Caso 2: Otra restricción tipo mochila como soporte	145
5.4.3 Caso 3: Misma restricción como soporte	147
5.5 Identificación de cliques maximales	151
5.6 Ejemplo ilustrativo	152
APENDICE 1: DESCOMPOSICION DE BENDERS	157
REFERENCIAS	163

Prólogo

El problema del viajante, TSP (del inglés, Traveling Salesman Problem), problema que, planteado sobre una red, consiste en obtener un circuito hamiltoniano a coste mínimo, es quizá el problema sobre el que más se ha escrito de entre los problemas no resueltos dentro de la Optimización Combinatoria (es decir, problemas con complejidad NP para la obtención de la solución óptima). Esto es así no sólo por la riqueza de sus aplicaciones, sino porque ha sido un problema paradigmático, que ha influido en el desarrollo de numerosos conceptos de optimización y algoritmos a lo largo de su historia.

La aparición del artículo de Dantzig, Fulkerson y Johnson (1954), "Solution of a large-scale traveling-salesman problem" en la revista *Journal of the Operations Research Society of America* fue uno de los principales eventos en la historia de la Optimización Combinatoria. En esa época ya Dantzig había desarrollado el método Simplex para programación lineal, y distintos problemas de optimización combinatoria, como los problemas de asignación y de transporte, se habían resuelto en este contexto, mediante adaptaciones del método simplex que dieron lugar a algoritmos con complejidad polinomial.

El problema de asignación consiste en elegir n elementos de una matriz $C = (c_{ij})$, uno por fila y por columna, de forma que la suma de los elementos elegidos sea mínima. Este problema puede resolverse utilizando técnicas de programación lineal de la siguiente manera:

sea el politopo P formado por todas las matrices de la forma $X = (x_{ij})$ que satisfagan las condiciones $x_{ij} \geq 0$, $\sum_j x_{ij} = 1 \quad \forall i$, $\sum_i x_{ij} = 1 \quad \forall j$.

Se plantea el problema de programación lineal consistente en encontrar la matriz $X \in P$ tal que se minimice la función $\sum_{i,j} c_{ij} x_{ij}$.

Ambos problemas son equivalentes, ya que en este caso los vértices de P son precisamente las $n!$ matrices que cumplen las condiciones (es decir, matrices con un 1 en cada fila y columna y 0 en todas las demás posiciones), y el óptimo del problema de programación lineal se alcanza siempre en uno de los vértices.

Al intentar aplicar el mismo procedimiento para resolver el problema del viajante, donde $x_{ij} = 1$ significa que nos movemos del nudo i al nudo j en el camino hamiltoniano elegido, se observa que, aunque similar en su planteamiento, el problema cambia substancialmente. Sea la siguiente identificación del politopo Q como el formado por todas las matrices $X = (x_{ij})$ que satisfacen

$$\sum_j x_{ij} = 1 \quad \forall i, \quad \sum_i x_{ij} = 1 \quad \forall j, \quad x_{ij} \geq 0,$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1 \quad \forall S / 1 < |S| < n,$$

donde la nueva condición introducida evita la formación de subciclos, (i.e., caminos hamiltonianos de cardinal inferior a $|S|$), situación que el problema de asignación no excluía en su planteamiento. El número de condiciones así introducidas para problemas de medianas dimensiones es exponencialmente grande.

Se advierte por tanto que ahora el número de condiciones necesarias para definir el politopo se incrementa de manera considerable, pero; sobre todo, que en este caso ya no es cierto que todos los vértices de Q tengan

coordenadas enteras 0-1. Es por eso necesario añadir una nueva condición al problema que asegure la integralidad de las soluciones, pasando así de un problema resoluble por técnicas de programación lineal a un problema de programación lineal entera, cuyas técnicas de solución no estaban desarrolladas en absoluto en esos momentos.

Si, dadas todas las matrices X del politopo Q con todos sus valores 0 ó 1 (todas los posibles caminos hamiltonianos), fuera posible determinar un nuevo politopo, sea R , cuyos vértices fueran exactamente estas soluciones, entonces el problema podría resolverse utilizando programación lineal. El politopo R deseado es la envoltura convexa de todas las soluciones factibles. Pronto se observó que las desigualdades que caracterizan este politopo no son de fácil obtención y, en cualquier caso, el número requerido haría infactible su utilización para fines prácticos.

Así, Dantzig, Fulkerson y Johnson especularon con la posibilidad de, partiendo de una solución óptima o casi óptima, probar la optimalidad introduciendo en el problema caracterizado por el politopo Q un número adicional de desigualdades (denominadas *cortes*) que, aún no definiendo por completo el politopo R , eliminaran soluciones fraccionales sin eliminar ninguna solución entera. De esta forma lograron resolver un problema del viajante con 49 ciudades, pero, sobre todo, mostraron que el concepto de *planos de corte* era relevante en general para resolver otros tipos de problemas de programación lineal entera. Además en ese artículo, quizá por primera vez, se utilizó el concepto de Branch-and-Bound, técnica cuyo desarrollo ha sido fundamental para la solución de la mayoría de los problemas de optimización combinatoria planteados en el marco de la programación lineal entera.

Otro procedimiento importante que determina cotas sobre las que aplicar algoritmos de tipo Branch-and-Bound es la *relajación Lagrangiana*, que se describe en el capítulo 3. Held y Karp (1970, 1971) utilizaron la primera versión de este procedimiento para diseñar un algoritmo para el TSP más eficiente que todos los obtenidos hasta entonces utilizando relajaciones lineales. De nuevo el problema del viajante servía de base para el desarrollo de técnicas de optimización cuyo uso se extendería a

numerosos problemas combinatorios.

En cuanto a sus aplicaciones, el problema del viajante, tanto en su forma simétrica, TSP, (con $c_{ij} = c_{ji} \forall i, j$) como asimétrica, ATSP (del inglés Asymmetric Traveling Salesman Problem), es el modelo clásico para secuenciar de forma óptima un conjunto de elementos. Normalmente, el valor óptimo de la secuenciación se expresa como la suma de los costes de secuenciar pares de elementos sucesivos. Aunque multitud de situaciones reales pueden modelizarse de esta forma, en gran número de casos la secuenciación buscada debe satisfacer otro tipo de condiciones adicionales, tal que modifican notablemente la naturaleza del problema. Aparecen así nuevos problemas combinatorios, que deben ser estudiados independientemente.

Una de las condiciones adicionales que aparece con frecuencia es la presencia de *relaciones de precedencia*; ciertos elementos deben aparecer antes que otros en la secuencia. Otra condición frecuente, especialmente en problemas de secuenciación de trabajos, donde cada elemento lleva asociado un peso, es la presencia de *fechas tope* antes de las cuales cada elemento debe haber sido secuenciado.

Escudero (1988) introdujo el ATSP con relaciones de precedencia, al que llamó *Problema del Ordenamiento Secuencial*, SOP (del inglés, Sequential Ordering Problem).

Escudero y Sciomachen (1993) han extendido el problema introduciendo pesos en los nudos del grafo del problema y acotaciones en el peso de los subcaminos desde el origen hasta cada nudo o elemento. Este es el problema objeto de estudio en esta monografía, que está dividida en cinco capítulos, estructurados de la forma siguiente:

En el capítulo 1 se describe formalmente el problema del ordenamiento secuencial así como otros problemas combinatorios relacionados; se establece su complejidad algorítmica y se describen los métodos generales de resolución para problemas de este tipo que se aplican actualmente en optimización combinatoria.

El capítulo 2 recoge el estado actual de la investigación referente al problema que nos ocupa, centrada en la metodología que denominaremos *primal*, consistente en trabajar con sucesivas relajaciones lineales del problema. Se estudian diferentes modelizaciones para el SOP y se resumen los algoritmos de separación desarrollados para detectar restricciones violadas e introducirlas en forma de planos de corte válidos que refuerzan la relajación lineal actual.

Para el SOP (i.e., ATSP con relaciones de precedencia y acotaciones) los resultados computacionales obtenidos a partir de esta metodología muestran que el modelo no proporciona buenas cotas inferiores, ya que en general, a pesar de identificar y añadir a la relajación del problema nuevos cortes detectados, la función objetivo no mejora, o lo hace muy lentamente. Por tanto, parecía necesario enfrentarse al problema desde un nuevo enfoque, que permitiera la obtención de una mejor cota inferior a la solución óptima desde la que iniciar la metodología branch-and-bound. A este propósito está dedicada el resto de la monografía.

El capítulo 3 está dedicado a establecer las bases de la metodología que denominaremos *dual*. En el se recogen los resultados fundamentales referentes a la relajación lagrangiana, se desarrolla el algoritmo básico para la aplicación de esta relajación a la resolución del SOP, y se obtiene un nuevo método para la estimación de los multiplicadores de Lagrange. Este nuevo método está basado en la aplicación de la metodología de descomposición de Benders y en la generación de cortes para el problema relajado que, en nuestro caso, es el problema de la arborescencia de mínimo peso. Así, el problema relajado pasa de ser un problema lineal a ser un problema combinatorio, conservandose siempre la integralidad de las soluciones obtenidas.

A la identificación de cortes válidos para el problema está dedicado el capítulo 4. En las secciones 4.2 y 4.3, respectivamente, se obtienen cortes válidos inducidos por las relaciones de precedencia y por las acotaciones (condiciones que no han sido introducidas en el problema

relajado), y se desarrollan diversos algoritmos de separación que permiten detectar y reforzar cortes que sean violados por la solución actual de dicho problema. La inclusión de estos cortes válidos no destruye la estructura de arborescencia del problema relajado, pues estos cortes se utilizan para modificar los coeficientes de la función objetivo.

Finalmente, en el capítulo 5, tomando como base algunos trabajos sobre reforzamiento de condiciones de tipo cubrimiento vía incremento de coeficientes, se propone una metodología para reforzar los cortes válidos obtenidos en el capítulo 4 antes de ser introducidos en el problema, así como aquellos cortes ya detectados e introducidos, sean restricciones tipo cubrimiento o, si sus coeficientes ya han sido incrementados en reforzamientos anteriores, restricciones tipo mochila. Se pretende de esta forma deteriorar al máximo la función objetivo, para acelerar así su convergencia al óptimo.

Capítulo 1

Descripción del problema y Preliminares

1.1.- INTRODUCCION

El Problema del Viajante, TSP (del inglés, Traveling Salesman Problem), tanto en su forma simétrica como asimétrica, es el modelo clásico para secuenciar de forma óptima un conjunto de elementos. Normalmente, el valor óptimo de la secuenciación se expresa como la suma de los costes de secuenciar pares de elementos sucesivos. Aunque multitud de situaciones reales pueden modelizarse de esta forma, en gran número de casos la secuenciación buscada debe satisfacer otro tipo de condiciones adicionales, tal que modifican notablemente la naturaleza del problema. Aparecen, por tanto, nuevos problemas combinatorios, uno de los cuales es el problema que nos ocupa, el Problema del Ordenamiento Secuencial, SOP (del inglés, Sequential Ordering Problem).

Este capítulo está organizado de la forma siguiente. En la sección 1.2 se define formalmente el Problema del Ordenamiento Secuencial. En la sección 1.3 se recogen algunas de sus aplicaciones prácticas. En la sección 1.4 se introduce la notación que se utilizará a lo largo de todo el trabajo. A continuación, en las secciones 1.5 y 1.6, se definen los

conceptos básicos necesarios para la exposición posterior, así como aquellos problemas combinatorios relacionados con el SOP o necesarios en los algoritmos que se desarrollan posteriormente. Por último, en la sección 1.7, se encuadra el problema de acuerdo a su complejidad algorítmica y se exponen los métodos generales de resolución de problemas de este tipo. En esta sección se presentan la teoría y el tipo de técnicas que se desarrollarán en los capítulos posteriores, indicando, por tanto, el marco de los trabajos objeto de esta monografía.

1.2.- DESCRIPCION DEL PROBLEMA

Sea un grafo dirigido o digrafo $G = (V,A)$, donde V es el conjunto de nudos y A es el conjunto de arcos. Sea c_{ij} el peso del arco $(i,j) \forall (i,j) \in A$. Un camino es una secuencia de nudos y arcos $i_1, (i_1, i_2), i_2, (i_2, i_3), \dots, (i_{r-1}, i_r), i_r$ tal que $(i_k, i_{k+1}) \in A \quad \forall k=1, \dots, r-1$. El Problema del Viajante consiste en obtener un camino Hamiltoniano \mathcal{H} en G , es decir, un camino que contenga todos los vértices de G sin repetir ninguno, tal que su peso sea mínimo. El problema se denomina *Problema del Viajante Simétrico* (TSP) si $c_{ij} = c_{ji} \quad \forall (i,j) \in A$. En caso contrario se denomina *Problema del Viajante Asimétrico* (ATSP). En este trabajo se supone que $\exists (i,j) \in A$ tal que $c_{ij} \neq c_{ji}$.

Sea 0 el nudo ficticio, tal que $A = A \cup \{(0,j) \quad \forall j \in V\}$ con $c_{0j} \geq 0$, desde el que se supone que comienza cualquier camino, y sea u el último nudo secuenciado. Sea (k, \dots, l) la secuencia ordenada de nudos desde el nudo k hasta el nudo l en un camino Hamiltoniano dado. Así $\mathcal{H} \equiv (0, \dots, u)$. Sea $i \in (k, \dots, l)$ si el nudo i pertenece al camino desde el nudo k hasta el nudo l , y sea $(i,j) \in (k, \dots, l)$ si el arco (i,j) forma parte del camino. Se denota por $i < j$ el hecho de que el nudo i se secuencie en \mathcal{H} antes que el nudo j .

Sea el grafo dirigido $P = (N, \Pi)$ con $N \subseteq V$, tal que un camino \mathcal{H} factible debe satisfacer las relaciones de precedencia dadas por el conjunto de arcos Π , de forma que $(i,j) \in \Pi$ exige que $i < j$ en todo camino Hamiltoniano factible en G . Se supone que Π tiene la propiedad transitiva

cerrada (i.e., si $(i,k) \in \Pi$ y $(k,j) \in \Pi$ entonces $(i,j) \in \Pi$). Escudero (1988) introdujo el ATSP con relaciones de precedencia, al que llamó *Problema del Ordenamiento Secuencial*.

Escudero y Sciomachen (1993) han extendido el problema introduciendo pesos en los nudos del grafo y acotaciones en el peso de los subcaminos del camino \mathcal{H} factible en G . Sea p_j el peso del nudo j , $\forall j \in V$, con $p_0 = 0$. Dado un vértice i , sea y_i el peso acumulado del nudo i en un camino hamiltoniano \mathcal{H} factible en G , tal que

$$y_i = \sum_{(a,b) \in (0, \dots, i)} c_{ab} + \sum_{b \in (0, \dots, i)} p_b. \quad (1.1)$$

Por tanto el peso total del camino hamiltoniano coincide con el peso acumulado del nudo u .

Sea el vector n -dimensional d tal que d_i proporciona una cota superior al peso acumulado del vértice i , $\forall i \in V$ en cualquier camino factible \mathcal{H} .

El PROBLEMA DEL ORDENAMIENTO SECUENCIAL (SOP) consiste en, dado un grafo dirigido con pesos en los vértices y en los arcos, obtener un camino hamiltoniano con peso total mínimo satisfaciendo las relaciones de precedencia y con acotaciones superiores en el peso de cada subcamino hamiltoniano.

Formalmente, dados los digrafos $G=(V,A)$ y $P=(N,\Pi)$, y los atributos p_i , $d_i \forall i \in V$ y $c_{ij} \forall (i,j) \in A$, el SOP consiste en obtener el camino Hamiltoniano \mathcal{H} en G tal que

- a. $\exists (j,i) \in \Pi$ para $i < j$ en \mathcal{H} .
- b. $y_i \leq d_i \forall i \in V$, donde la expresión para y_i viene dada en (1.1).
- c. El peso $\sum_{(i,j) \in \mathcal{H}} c_{ij}$ es mínimo.

Si $\Pi = \emptyset$ y $\exists j \in V / d_j$ es finito, el SOP se convierte en el ATSP. (Lawler et al. (1985) recogen numerosas referencias sobre el TSP/ATSP.)

1.3.- CAMPOS DE APLICACION

El SOP tiene gran número de aplicaciones prácticas. Una aplicación obvia es el ATSP con ciudades origen y destino fijadas, en el que visitar una ciudad requiere una cantidad dada de tiempo y la partida de las ciudades debe satisfacer unas fechas dadas.

El SOP aparece, por ejemplo, y es una aplicación interesante, en problemas de planificación de la producción (ver Escudero, 1989). El problema básicamente es el siguiente: Una máquina tiene que procesar un conjunto de trabajos, sólo puede ejecutar uno cada vez y los trabajos no pueden ser interrumpidos después de comenzados. Se conoce el tiempo de proceso para cada trabajo. Además existe un tiempo de ajuste entre el proceso de dos trabajos distintos, que depende de ambos trabajos. (La secuenciación de trabajos que pertenecen a la misma familia no precisa de ningún tiempo de ajuste). Cada trabajo puede tener una fecha límite para ser entregado. Además pueden existir relaciones de precedencia entre los trabajos, de forma que algunos no pueden comenzar hasta que otros estén finalizados. El objetivo consiste en obtener la secuencia de trabajos que satisfaga todas estas condiciones, tal que se minimice el tiempo de finalización del último trabajo que se procese.

1.4.- NOTACION

Sea el grafo dirigido $G=(V,A)$, donde V es el conjunto de vértices y A el conjunto de arcos, y sean W y W' dos subconjuntos de V , F un subconjunto de A , y los vectores $x \in \mathbb{R}^{|A|}$, $z, a \in \mathbb{R}^{|V|}$. Para un camino hamiltoniano \mathcal{H} en G , se define:

1) $A(W)$ es el conjunto de arcos del subgrafo generado por W ;

$$A(W) \equiv \{ (i,j) \in A \mid i,j \in W \}$$

2) $x(F) \equiv \sum_{(i,j) \in F} x_{ij}$

3) Por simplicidad de notación nos referiremos a veces a un camino como una secuencia de nudos (i_1, i_2, \dots, i_r) , enumerando implícitamente los arcos que los unen, o como una secuencia de arcos sin mencionar los nudos. Así, (i, \dots, j) representa el conjunto ordenado de nudos desde i hasta j en el camino \mathcal{H} .

4) Dado el conjunto $W = \{i, a, b, j\}$ y el subcamino (i, a, b, j) ,

$$x(i, \dots, j) = x_{ia} + x_{ab} + x_{bj}$$

$$x(W) \equiv x(A(W))$$

5) $(i, \dots, j) \cup (a, \dots, b)$ representa el conjunto ordenado de nudos $(i, \dots, j, a, \dots, b)$.

6) $i < j$ significa que el nudo i se secuencia antes (no necesariamente inmediatamente antes) que el nudo j en un camino Hamiltoniano dado.

$$7) \delta^+(j) \equiv \{ (j, k) \in A \}$$

$$\delta^-(j) \equiv \{ (i, j) \in A \}$$

8) $w^+(W)$ es el conjunto de todos los arcos con nudo origen en W ,

$$(i.e. (i, j) \in A / i \in W, j \notin W)$$

$w^-(W)$ es el conjunto de todos los arcos con nudo destino en W ,

$$(i.e. (j, i) \in A / j \notin W, i \in W)$$

$$9) W:W' \equiv w^+(W) \cap w^-(W') = \{ (i, j) \in A / i \in W, j \in W' \}$$

$$\text{Por tanto, } W:W \equiv A(W)$$

$$10) j:W \equiv \{ \{j\}:W \} \quad \forall j \in V$$

$$W:j \equiv \{ W:\{j\} \} \quad \forall j \in V$$

11) $N(F)$ es el conjunto de nudos de V incidente en el conjunto de arcos F .

- 12) $z(i)$ es el nudo que se secuencia inmediatamente después que el nudo i en un camino \mathcal{H} dado.
- 13) $a(i)$ es el nudo que se secuencia inmediatamente antes que el nudo i en un camino \mathcal{H} dado.
- 14) Dado el camino Hamiltoniano (i, \dots, j) , $R(i, \dots, j)$ es el camino $(j, a(j), a(a(j)), \dots, z(z(i)), z(i), i)$, es decir, la secuenciación de los nudos de i a j en orden inverso.
- 15) $P(j)$ es el conjunto de nudos tal que $i \in P(j) \Leftrightarrow (i, j) \in \Pi$.
- 16) $T_{ab} \equiv c_{ab} + p_b$, $\forall (a, b) \in A$

De acuerdo con esta notación, el peso acumulado del camino Hamiltoniano $(0, \dots, i)$ será

$$y_i = \sum_{(a, b) \in (0, \dots, i)} T_{ab} \quad (1.2)$$

1.5.- DEFINICIONES BASICAS

Definición 1.1: Se llama *árbol* a un grafo conexo y sin ciclos. Se llama *bosque* a un grafo sin ciclos.

Definición 1.2: Se llama *arborescencia* a un digrafo conexo y sin ciclos, tal que cada nudo es vértice destino de un único arco, exceptuando un nudo al que no llega ningún arco, llamado nudo *raiz*.

Definición 1.3: Dado un grafo (digrafo) $G = (V, E)$ ($G = (V, A)$) y un conjunto $V' \subset V$, se llama *subgrafo generado por V'* al grafo (digrafo) $G_{V'} = (V', E_{V'})$ ($G_{V'} = (V', A_{V'})$) donde

$$E_{V'} = \{ (x, y) \in E / x \in V', y \in V' \}$$

$$(A_{V'} = \{ (x, y) \in A / x \in V', y \in V' \}).$$

Definición 1.4: Dado un grafo (digrafo) $G = (V,E)$ ($G = (V,A)$) y un conjunto $F \subseteq E$ ($F \subseteq A$) se llama *grafo parcial generado por F* al grafo (digrafo) $G_F = (V_F, F)$, donde $V_F \subseteq V$ recoge sólo los nudos que definen (nudos origen o destino) algún arco en F.

Definición 1.5: Dado un grafo $G = (V,E)$ se llama *árbol generador de G* a un grafo parcial conexo y sin ciclos de G que contiene a todos sus nudos. Dado un digrafo $G = (V,A)$ se llama *arborescencia generadora de G* a un grafo parcial de G, sea $G_C = (V,C)$ con $C \subseteq A$, tal que G_C es una arborescencia.

Definición 1.6: Dado un digrafo $G = (V,A)$ se llama *camino Hamiltoniano* en G a un camino en G que contiene a todos sus nudos. Si el nudo inicial y el nudo final del camino coinciden, entonces se denomina *circuito Hamiltoniano*.

Definición 1.7: Dado un digrafo $G = (V,A)$ se llama *corte* en G a un conjunto de arcos $Q \subseteq A$ tal que verifica que el grafo $G' = (V, A \setminus Q)$ no es conexo y ningún subconjunto de Q verifica esta propiedad.

Definición 1.8: Dado un digrafo $G = (V,A)$ y un digrafo $P = (V,\Pi)$ de precedencias, se llama *politopo del ordenamiento secuencial* a la envoltura convexa del conjunto de puntos que son vectores de incidencia de caminos hamiltonianos factibles en G, es decir, soluciones factibles del SOP. Se denotará por $SOP(n,P)$, donde $n \equiv |V|$.

Definición 1.9: Dado un politopo $P = \text{conv} \{ x \in \{0,1\}^n / Ax \leq b \}$, donde A es una matriz real $m \times n$ y b un vector m-dimensional, una desigualdad de la forma $\alpha x \leq \beta$ es una *desigualdad válida* para P si se satisface para todo $x \in P$.

Definición 1.10: Se llama *cara* de un politopo P correspondiente a la desigualdad $\alpha x \leq \beta$ al conjunto $F = \{ x \in P / \alpha x = \beta \}$. Si $\emptyset \neq F \neq P$ la cara se llama *propia*. En este caso, se dice que $\alpha x \leq \beta$ define una cara.

Definición 1.11: Un politopo P es de *dimensión* k ($\dim P = k$) si el máximo número de puntos en P afínmente independientes es $k+1$.

Definición 1.12: Se llama *faceta* de un politopo P a toda cara propia de P con dimensión máxima; es decir, una cara F del politopo P es una faceta si y sólo si $\dim F = \dim P - 1$.

1.6.- PROBLEMAS DE OPTIMIZACION COMBINATORIA RELACIONADOS CON EL SOP Y CON SU RESOLUCION.

1.6.1.- El problema de asignación (Assignment Problem, AP).

Dada una matriz cuadrada C de dimensión n , donde c_{ij} es el elemento correspondiente a la fila i -ésima y la columna j -ésima, el problema de asignación consiste en encontrar una permutación (i_1, \dots, i_n) de las columnas de C tal que se minimice $\sum_{j=1}^n c_{ji}$.

Muy frecuentemente, el problema descrito tiene su aplicación en el caso en que sea preciso asignar recursos a tareas a coste mínimo, tal que cada recurso ha de asignarse a una y sólo una tarea y cada tarea sólo puede tener asignado un recurso. En estos casos c_{ij} recoge el coste asociado a la asignación del recurso i a la tarea j .

Este problema se puede modelizar como problema de programación 0-1 de la siguiente forma:

$$\text{Sea } x_{ij} = \begin{cases} 1, & \text{si se asigna la fila } i \text{ a la columna } j \\ 0, & \text{en caso contrario} \end{cases}$$

Entonces, el problema de asignación consiste en

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

s.a.

$$(1) \quad \sum_{l=1}^n x_{lj} = 1 \quad \forall j=1, \dots, n$$

$$(2) \quad \sum_{j=1}^n x_{ij} = 1 \quad \forall i=1, \dots, n$$

$$(3) \quad x_{ij} \in \{0,1\} \quad \forall i,j=1, \dots, n$$

Existen excelentes algoritmos para resolver este problema, como el famoso método Húngaro, de complejidad $O(n^3)$ y, por tanto, fuertemente polinomial. Hoy día los algoritmos más utilizados tienen complejidad $O(\sqrt{n} m \log(nC))$, y, por tanto, así llamada débilmente polinomial, donde m es el número de arcos (i,j) con $c_{ij} > 0$, y $C = \max_{(i,j)} \{ c_{ij} \}$. Así el algoritmo de Bertsekas (1991) y otros. Ver también Ahuja, Magnanti y Orlin (1993).

1.6.2.- El problema de la arborescencia generadora de peso mínimo (Min-Sum Arborescence Problem).

Sea $G = (V,A)$ un digrafo conexo en el que cada arco $u \in A$ tiene asignado un peso $w(u)$, donde $u \equiv (i,j)$ para $i,j \in V$. El problema de la arborescencia generadora de peso mínimo consiste en encontrar la arborescencia generadora de G con menor suma de los pesos de sus arcos. La solución de este problema para el caso de grafos no dirigidos (*problema del árbol generador de peso mínimo, minimum spanning tree problem*) fue uno de los primeros éxitos de la optimización combinatoria, siendo su complejidad polinomial. (Ver Kruskal (1956)). La complejidad del problema dirigido es también polinomial; ver el algoritmo de Edmonds (1967). En el desarrollo de nuestro trabajo utilizaremos la implementación debida a Fischetti y Toth (1993).

Este problema se puede plantear como un problema de programación 0-1, tal que se define la siguiente variable 0-1:

$$x_{ij} = \begin{cases} 1, & \text{si el arco } (i,j) \text{ pertenece a la arborescencia} \\ 0, & \text{en caso contrario} \end{cases}$$

Sea 0 el nudo origen de la arborescencia generadora tal que $\exists(0,j) \in A$ y $|\exists(j,0) \in A, \forall j \in V$.

$$z = \text{Min } c^t x$$

s.a.

$$(1) x(A) = n-1$$

$$(2) x(\delta^-(j)) = 1 \quad \forall j \in V \setminus \{0\}$$

$$(3) x(W) \leq |W| - 1 \quad \forall W \subset V, \quad 2 \leq |W| \leq n-1$$

$$(4) x \in B^n$$

1.6.3.- El problema del camino mínimo (Shortest Path Problem).

Dado un digrafo $G = (V,A)$, donde cada arco $u=(i,j) \in A$ tiene asociado el parámetro $l(u)$, llamado *longitud* de u , el problema del camino mínimo desde un vértice a hasta otro vértice b consiste en encontrar un camino de la forma (a, \dots, b) tal que minimice

$$\sum_{u \in (a, \dots, b)} l(u)$$

Las longitudes de los arcos pueden ser positivas o negativas, mientras no exista ningún ciclo dirigido con longitud negativa en G .

La complejidad de este problema es polinomial, $O(n^2)$ para redes tales que $l(u) \geq 0 \quad \forall u \in A$ y $O(nm)$ ($O(n^3)$ para grafos completos) cuando no hay restricción en el signo de las longitudes de arco. Ver Gallo y Pallottino (1988); y Ahuja, Magnanti y Orlin (1993).

1.6.4.- El problema del flujo máximo (Maximum Flow Problem).

Dado un digrafo $G = (V,A)$, con $|A| = m$, donde cada arco $u=(i,j) \in A$ tiene asociado el parámetro real no negativo $c(u)$ denominado *capacidad* del arco, sean las siguientes definiciones:

Definición 1.13: Un *flujo* a través de G es un vector $\varphi = (\varphi_1, \dots, \varphi_m)$ de forma que se verifica la siguiente *ley de conservación de flujo*:

$$\forall i \in V \quad \sum_{u \in \delta^+(i)} \varphi_u = \sum_{u \in \delta^-(i)} \varphi_u$$

Definición 1.14: Un *flujo de s a t* es un vector φ tal que verifica la ley de conservación de flujo $\forall i \neq s, t$, y además se verifica que

$$\sum_{u \in \delta^+(s)} \varphi_u = \sum_{u \in \delta^-(t)} \varphi_u$$

Definición 1.15: Un flujo φ es *compatible* con G si $\varphi_u \leq c(u) \forall u \in A$.

Definición 1.16: Se llama *valor de un flujo φ de s a t* a

$$v(\varphi) = \sum_{u \in \delta^+(s)} \varphi_u$$

Dado un digrafo $G = (V, A)$, el *problema de flujo máximo de s a t* para $s, t \in V$ consiste en encontrar un flujo compatible a través de A tal que su valor sea máximo.

Definición 1.17: Dado un digrafo $G = (V, A)$, un conjunto $Q \subseteq A$ es un (s, t) -*corte* si su eliminación separa a G en dos componentes conexas y s y t no pertenecen a la misma componente conexa. Un (s, t) -corte, por tanto, divide a V en dos conjuntos disjuntos, sean W y \bar{W} , tales que $s \in \bar{W}$ (i.e., $s \notin W$) y $t \in W$. Así, $Q = \delta^-(W)$.

Definición 1.18: Se define la *capacidad* de un corte, $c(\delta^-(W))$, como

$$c(\delta^-(W)) = \sum_{i \in \bar{W}, j \in W} c(i, j)$$

Teorema 1.1: (Ford y Fulkerson 1956,1962) Sea φ un flujo compatible en un digrafo G . Si $\delta^-(W)$ es un corte en G , entonces

$$v(\varphi) \leq c(\delta^-(W)).$$

Teorema 1.2: (Flujo máximo-corte mínimo) (Ford y Fulkerson (1956)) Dado un digrafo G se verifica:

$$\max \{ v(\varphi) / \varphi \text{ compatible} \} = \min \{ c(\delta^-(W)) / \delta^-(W) \text{ corte en } G \}$$

Por tanto el problema del flujo máximo también se puede formular como sigue: *Dado un digrafo G con capacidades asociadas a sus arcos, encontrar un corte tal que su capacidad sea mínima.*

La complejidad de los algoritmos utilizados para resolver este problema, tales como el algoritmo de Goldberg y Tarjan (1988) y Orlin (1993), también es polinomial, siendo $O(nm^2 \log^2 n)$ para el primero y $O((n+m)(\log n)(m+n \log n))$ para el segundo. Para una panorámica sobre algoritmos de flujo, ver Ahuja, Magnanti y Orlin (1989, 1993).

1.6.5.- El problema del viajante (Traveling Salesman Problem, TSP).

Dado un digrafo $G = (V,A)$, con pesos c_{ij} asociados a los arcos (i,j) , para $(i,j) \in A$, $i,j \in V$, el problema, propiamente denominado, del viajante consiste en encontrar un circuito hamiltoniano \mathcal{H} en G tal que su longitud sea mínima, donde la longitud está expresada por la suma de los pesos c_{ij} de los arcos en el camino.

En la sección 1.2 se efectúa la distinción del TSP en simétrico y asimétrico. El TSP asimétrico tiene una gran utilidad para los objetivos de este trabajo.

Sea x_{ij} una variable 0-1 que indica si en arco (i,j) está en el circuito hamiltoniano o no, tal que

$$x_{ij} = \begin{cases} 1, & \text{si } (i,j) \in \mathcal{H} \\ 0, & \text{en caso contrario} \end{cases}$$

La modelización 0-1 del ATSP más apropiada para este trabajo es la siguiente:

$$z = \text{Min } c^t x$$

s.a.

$$(1) x(A) = n$$

$$(2) x(\delta^-(j)) = 1 \quad \forall j \in V$$

$$(3) x(\delta^+(j)) = 1 \quad \forall j \in V$$

$$(4) x(W) \leq |W| - 1 \quad \forall W \subset V, \quad 2 \leq |W| \leq n-1$$

$$(5) x \in B^n$$

La versión de este modelo para la obtención del camino Hamiltoniano en G de peso mínimo, (donde la condición (1) es sustituida por $x(A) = n-1$), será ampliamente utilizada a lo largo de este trabajo. (Ver, por ejemplo, sección 2.2.2).

El TSP es el problema de Optimización Combinatoria por antonomasia, muy utilizado en la literatura. El problema objeto de nuestro estudio puede considerarse como una extensión del ATSP. A diferencia de los problemas combinatorios anteriores, no es polinomial, sino NP -duro (ver sección 1.7.1); por tanto, no se conocen algoritmos polinomiales para su resolución.

Los algoritmos desarrollados para todos los problemas descritos en esta sección son útiles para la resolución del problema que nos ocupa (SOP).

1.7.- COTAS SUPERIORES Y COTAS INFERIORES EN OPTIMIZACION COMBINATORIA

La importancia del problema que nos ocupa, además de por sus aplicaciones prácticas, viene dada por el hecho de que es un problema típico de *Optimización Combinatoria*. En él tratamos de minimizar un peso total, luego es un problema de optimización; pero el conjunto de soluciones posibles no es un conjunto continuo, sino un conjunto finito: el conjunto de todos los posibles caminos que verifiquen una serie de condiciones. Se trata por tanto de un problema combinatorio. Así, las técnicas desarrolladas en el estudio de este problema y la filosofía subyacente a las metodologías que expondremos es válida para el estudio de otros tipos de problemas combinatorios.

1.7.1. Complejidad

Existen diferentes formas de medir la complejidad en los algoritmos en los que el número de iteraciones no está fijado determinísticamente por el tamaño del problema. Elegimos de entre ellas el análisis del peor de los casos, que, aunque presenta el inconveniente de permitir que sean casos patológicos los que determinen la complejidad, proporciona la teoría de la \mathcal{NP} -completitud. No es el objeto de esta monografía desarrollar con precisión esta teoría; por tanto, sólo se definirán brevemente los términos fundamentales utilizados en este trabajo. Para una exposición detallada sobre el tema ver Garey y Johnson (1979).

Definición 1.19: Se dice que un problema tiene *complejidad polinomial* si existe un algoritmo que lo resuelva tal que su complejidad en el peor de los casos está acotada por una función polinomial de n , el tamaño del problema. A un algoritmo de este tipo se le denomina *algoritmo polinomial*. En caso contrario se dice que el problema tiene *complejidad no polinomial*, (también denominada complejidad exponencial). Un algoritmo *no polinomial* es un algoritmo cuyo tiempo de ejecución es una función no polinomial en n .

Las tablas 1.1 y 1.2 ilustran numéricamente la importancia de disponer de algoritmos polinomiales para resolver un problema.

$t \setminus n$	10	20	30	40	50	60
n	.00001 secs.	.00002 secs.	.00003 secs.	.00004 secs.	.00005 secs.	.00006 secs.
n^2	.0001 secs.	.0002 secs.	.0009 secs.	.0016 secs.	.0025 secs.	.0036 secs.
n^3	.001 secs.	.008 secs.	.027 secs.	.064 secs.	.125 secs.	.216 secs.
n^5	.1 secs.	3.2 secs.	24.3 secs.	1.7 min.	5.2 min.	13.0 min.
2^n	0.01 secs.	1.0 secs.	17.9 min.	12.7 días	35.7 años	366 siglos
3^n	.059 secs.	58 min.	6.5 años	3855 siglos	2×10^8 siglos	1.3×10^{33} siglos

Tabla 1.1 Tiempo de cómputo de la solución óptima en función del tamaño del problema.

$t=f(n)$	Ordenador actual	100 veces más rápido	1000 veces más rápido
n	N_1	$100N_1$	$1000N_1$
n^2	N_2	$10N_2$	$31.6N_2$
n^3	N_3	$4.64N_3$	$10N_3$
n^5	N_4	$2.5N_4$	$3.98N_4$
2^n	N_5	$N_5 + 6.64$	$N_5 + 9.97$
3^n	N_6	$N_6 + 4.19$	$N_6 + 6.29$

Tabla 1.2: Tamaño máximo de un problema a optimizar en una hora. (En función de tecnología superior)

Formalmente la clasificación de problemas según su complejidad en esta teoría se reduce a *problemas de decisión*, es decir, problemas que exigen una respuesta de la forma "sí" ó "no". Sin embargo esto no restringe la aplicabilidad de la teoría, ya que cualquier problema de optimización se puede reformular como un problema de decisión y, en la mayoría de los casos, se verifica que el problema de optimización tiene complejidad polinomial si y sólo si el problema de decisión asociado la tiene.

Definición 1.20: Se dice que el problema A es *reducible en tiempo polinomial* al problema B, si existe un algoritmo para A que utiliza un algoritmo que resuelve el problema B, la utilización del algoritmo B es un subalgoritmo polinomial y el resto de las operaciones del algoritmo para A son también polinomiales.

Lema 1.1: Si existe una reducción en tiempo polinomial de A a B, y existe un algoritmo polinomial para B, entonces existe un algoritmo polinomial para A.

Definición 1.19: Se dice que el problema A es *transformable en tiempo polinomial* al problema B si A es reducible en tiempo polinomial a B, A y B son problemas de decisión y la llamada al algoritmo para B se realiza exactamente una vez.

Las principales clases en las que se agrupan los problemas de decisión son las siguientes:

Definición 1.22: La clase \mathcal{P} (del inglés, *Polynomial time*) es la clase de todos los problemas de decisión para los que existe un algoritmo polinomial.

Existe una clase de problemas que no pertenecen a la clase \mathcal{P} , pero no se ha podido probar que no existe un algoritmo polinomial para su resolución. Algunos de estos problemas se engloban dentro de la clase \mathcal{NP} (del inglés, *Nondeterministic Polynomial time*). Durante las últimas décadas se han dado tres diferentes formulaciones de las características

de los problemas en esta clase, todas ellas equivalentes. Para un desarrollo de estas formulaciones ver Garey y Johnson (1979) y Papadimitriou y Steiglitz (1982) entre otros. Se introduce a continuación una de ellas:

Definición 1.23: La clase NP es la clase de todos los problemas transformables en tiempo polinomial al problema de programación entera.

Definición 1.24: Un problema de decisión A es NP -completo si:

a) $A \in NP$ y b) cualquier problema de la clase NP es transformable en tiempo polinomial a A .

Por tanto, el problema de programación entera es NP -completo. Si se demostrara que un problema NP -completo pertenece a la clase P , todos los problemas de la clase NP podrían ser resueltos mediante algoritmos en tiempo polinomial.

Definición 1.25: Un problema (de decisión o no de decisión) es NP -duro si cualquier problema de la clase NP es reducible (no necesariamente transformable) en tiempo polinomial a dicho problema.

Se verifica que $P \subseteq NP$. Como se ha indicado anteriormente, no está probado el hecho de que $P \neq NP$ (es este uno de los grandes problemas abiertos de las matemáticas hoy día), pero la hipótesis general es que $P \neq NP$ y, por tanto, que los problemas NP -completos pertenecen a $NP \setminus P$.

La mayoría de los problemas que se engloban dentro de la Optimización Combinatoria son problemas que pertenecen a la clase NP -dura. (Problemas polinomiales son el problema de asignación, el problema del árbol generador, el problema del camino mínimo, y el problema del flujo máximo entre otros).

Es bien sabido que el problema del viajante es NP -duro. Por tanto también el SOP lo es. Es, por tanto, improbable que haya algún algoritmo eficiente para obtener recorridos óptimos, con esfuerzo computacional

aceptable, cuando el cardinal de V es grande. Hoy día problemas de grandes dimensiones de los que se sospecha que no tienen complejidad polinomial se abordan de dos formas: bien por procedimientos aproximativos, bien por medio de relajaciones del problema.

1.7.2. Procedimientos aproximativos (heurísticos)

Un enfoque para resolver el problema es diseñar algoritmos heurísticos eficientes, que, aunque no garanticen la obtención de la solución óptima, normalmente, encuentren soluciones aceptables. Un algoritmo heurístico se mueve de una solución factible a otra no necesariamente con mejor valor en la función objetivo, hasta que llega a una solución bloqueada (a un óptimo local del problema). Por tanto, para un problema de minimización, un algoritmo heurístico proporciona cotas superiores al valor óptimo del problema. Ver por ejemplo Reeves (1993).

Algoritmos de búsqueda local

Uno de los tipos de algoritmos heurísticos más utilizados son los algoritmos de *búsqueda local*.

Un algoritmo de búsqueda local se basa en procedimientos de búsqueda en entornos: a partir de una solución dada (donde la solución inicial se obtiene utilizando otra heurística si es necesario) el algoritmo examina (implícita o explícitamente) todas las posibles soluciones en dicho entorno y, si existe, encuentra una con mejor valor en la función objetivo.

La noción de entorno varía en cada heurística y en cada problema específico. En nuestro caso, dado un camino hamiltoniano, los caminos pertenecientes a su entorno serán aquellos caminos que pueden obtenerse por medio de un número limitado de intercambios de arcos del camino por arcos no pertenecientes al mismo.

El tiempo de ejecución de una heurística de búsqueda local depende del tiempo necesario para la aplicación del procedimiento de búsqueda en el entorno y del número de iteraciones que se requieran. En general, para el problema de búsqueda de caminos de peso mínimo, se verifica el siguiente teorema:

Teorema 1.3: (Papadimitriou & Steiglitz, 1982) Si A es un algoritmo de búsqueda local con tiempo de búsqueda en entornos acotado por un polinomio, entonces, asumiendo que $\mathcal{P} \neq \mathcal{NP}$, no se puede garantizar que A encuentre un camino de peso acotado por un múltiplo constante de la solución óptima del problema, incluso si se permitiera un número exponencial de iteraciones.

Por tanto es posible que una heurística de búsqueda local llegue a una solución bloqueada muy lejos de la solución óptima del problema.

Sin embargo, excepto en casos patológicos, la experiencia computacional demuestra que este tipo de heurísticas suelen proporcionar aceptables cotas superiores a la solución óptima (en un problema de minimización).

Se han desarrollado distintas heurísticas para el SOP que proporcionan aceptables cotas superiores a la solución óptima del problema. Escudero (1988) y (1989) y Escudero y Sciomachen (1993) describen un conjunto de algoritmos para obtener una solución inicial factible para el SOP, así como algoritmos heurísticos para mejorar la solución inicial mediante intercambios en los que intervienen k nudos para $k = 3$ y 4 . A lo largo de este trabajo supondremos en todo momento que puede encontrarse una solución factible al problema, y, por tanto, que la solución óptima está acotada superiormente. Pero el desarrollo y alcance de las heurísticas queda fuera del propósito de esta monografía.

1.7.3. Procedimientos de relajación

Dado un problema (P), sea $F(P)$ su conjunto de soluciones factibles, y z_P el valor en la función objetivo $z_P(x)$ del punto óptimo x^* .

Definición 1.26: Dado un problema (P) de minimización, un problema (R) de minimización es una relajación de (P) si y sólo si:

- (i) $F(P) \subseteq F(R)$
- (ii) $z_R \leq z_P$ en $F(P)$

Existen diferentes métodos para obtener relajaciones de un problema dado. Uno de los más utilizados consiste en considerar no todas, sino sólo algunas de las condiciones del problema, de forma que se satisfice la condición (i). Dado que la función objetivo para ambos problemas es la misma, resulta que $z_R(x) \leq z_P(x)$.

Ejemplo 1.1: Dado un problema IP de programación entera

$$z_{IP} = \text{Min} \{ z_{IP}(x) : x \in S, x \text{ entero} \}$$

la *relajación lineal* de IP es $z_{LP} = \text{Min} \{ z_{IP}(x) : x \in S \}$.

Otra forma de conseguir relajaciones consiste en eliminar algunas restricciones del problema e introducirlas en la función objetivo de forma que la condición (ii) se satisfaga. Esta técnica se conoce con el nombre de *relajación Lagrangiana*.

Los procedimientos de relajación se basan en la resolución de problemas relajados (que se suponen más fáciles de resolver) en lugar del problema original. Como se verifica siempre que el valor en la función objetivo del punto óptimo del problema relajado no es superior (si se trata de un problema de minimización) al óptimo del problema original, estos métodos proporcionan cotas inferiores para el óptimo de este problema. Se les exige para ser eficientes que tengan complejidad polinomial.

Mediante los procedimientos de relajación el objetivo consiste en obtener el intervalo de optimalidad apropiado para el problema original, en lugar de su solución óptima. Llamando z^H al valor en la función objetivo de la mejor solución factible obtenida heurísticamente, entonces el cociente $(z^H - z_R)/z_R$ proporciona el intervalo de optimalidad y, por definición, mide la bondad de la solución heurística obtenida.

Definición 1.27: Dado un problema (P), un punto $x^* \in F(P)$ y un $\epsilon > 0$, se dice que x^* es una solución ϵ -óptima de (P) si se verifica la siguiente condición (para un problema de minimización):

$$z_P(x^*) \leq z_R + \epsilon$$

La obtención de la cota z_R puede no ser suficiente para probar optimalidad, pero los procedimientos de relajación permiten la identificación de soluciones ϵ -óptimas, ya que, si un punto x^* factible satisface $z_P(x^*) \leq z_R + \epsilon$, entonces x^* es una solución ϵ -óptima.

La idea básica del método es relajar algunas condiciones del problema de forma que el problema relajado pueda resolverse en tiempo polinomial, obteniendo así una primera cota inferior al valor óptimo. A continuación se pretende mejorar el valor de la cota obtenida. Para ello es necesario identificar (en tiempo polinomial) alguna condición relajada que sea violada por la solución actual, y, de alguna manera, introducirla en el problema. Se consigue así un nuevo problema relajado más fuerte que el anterior, que debe poder resolverse también en tiempo polinomial.

La forma usual de introducir en el problema relajado condiciones violadas es hacerlo en forma de planos de corte. Esta metodología, aplicada al SOP, se desarrolla en el capítulo 2, donde se describen distintos modelos a relajar y diversas técnicas para identificar e introducir en el problema cortes violados que refuercen la cota inicial. En este contexto tiene gran importancia el estudio del polítopo de soluciones factibles, en orden a identificar aquellas restricciones que sean facetas del mismo.

Un método alternativo puede consistir en introducir en el problema relajado condiciones violadas en forma de cortes que modifiquen linealmente los coeficientes de la función objetivo. Esta es la metodología elegida en este trabajo, que se desarrollará en los restantes capítulos de la monografía.

Capítulo 2

Metodología Basada en el Primal

2.1.- INTRODUCCION

Sea el modelo general de programación 0-1 IP

Modelo 2.1:

$$\begin{aligned} z &= \text{Max } cx \\ \text{s.a.} \\ Ax &\leq b \\ x &\in B^n \end{aligned}$$

donde x es el vector columna del conjunto J de variables 0-1, $n \equiv |J|$, c es el vector fila de la función objetivo, A es la matriz de coeficientes de las restricciones y b el vector del lado derecho.

Denominaremos *Metodología Primal* al tipo de técnicas y modelos de búsqueda de cotas inferiores a la solución óptima de un problema de programación 0-1 (2.1) y, en general, programación con variables enteras, tal que tengan las siguientes características:

- 1) Se relaja la condición de integralidad de las variables binarias;

estas variables en el problema relajado pasan a ser variables acotadas entre 0 y 1. Así, el problema a resolver en cada iteración es un problema de programación lineal (PPL).

- 2) Pueden también relajarse otras condiciones del problema original.

- 3) Resuelto en una iteración un problema de programación lineal, se trata de identificar condiciones que sean violadas por la solución óptima del problema relajado actual, y que, o bien son condiciones relajadas del problema original, o bien son cortes válidos para el problema original. Esta información se utiliza para añadir nuevos cortes al PPL, y, por tanto, para reducir el conjunto de soluciones factibles lineales, por supuesto sin reducir el conjunto de soluciones enteras factibles del problema original.

Esta metodología requiere para su aplicabilidad que la complejidad de los algoritmos para identificar condiciones violadas sea polinomial.

Así mismo requiere que la resolución del problema relajado en cada iteración tenga también una complejidad polinomial. Esta última condición se satisface ya que la programación lineal tiene este tipo de complejidad. Ver los trabajos pioneros de Khachian (1979) y Karmarkar (1984).

Utilizando esta metodología se han diseñado algoritmos que proporcionan cotas inferiores a un gran número de problemas combinatorios. (Ver Grötschel, Lovasz y Schrijver, 1988). Veremos a lo largo de este capítulo como esta metodología se ha utilizado para la obtención de algoritmos para el SOP.

En primer lugar, en la sección 2.2, se plantea el problema de la modelización más eficiente del SOP para ser utilizada por un algoritmo que se base en la metodología que hemos denominado *primal*. Se plantean diversas modelizaciones equivalentes en el sentido que se definirá más adelante, y se analiza la bondad de cada una desde el punto de vista computacional.

A continuación, en la sección 2.3, se desarrollan los aspectos fundamentales de la técnica conocida como *preproceso*, que tratará de reducir el número de puntos lineales factibles del problema sin eliminar ninguna solución factible entera.

En las secciones 2.4 y 2.5 se estudian los algoritmos para la obtención de cotas inferiores, algoritmos que siguen el esquema general de los algoritmos de planos de corte conocidos en la literatura, haciendo especial incapie en los algoritmos de separación desarrollados para algunas de las restricciones específicas del problema que nos ocupa. En la sección 2.4 se estudia el caso en el que no existen acotaciones al peso acumulado (i.e., $d_j = \infty \forall j \in V$), y en la sección 2.5 el caso en el que este peso sí está acotado.

Por último, en la sección 2.6, se describe brevemente el procedimiento exacto (exponencial, basado en técnicas "branch-and-bound") a seguir cuando la única condición violada por una solución obtenida es la condición de integralidad, condición que en todo momento aparece relajada en esta metodología primal para obtener cotas inferiores al valor óptimo del problema.

2.2.- MODELOS

2.2.1.- Introducción. Equivalencia y bondad de modelos

Sea IP el problema de programación entera 2.1. (Se supone que el problema está propiamente dimensionado).

Definición 2.1: Se llama *relajación lineal* del problema IP el mismo sistema (2.1) donde cada x_j puede tomar cualquier valor del intervalo $[0,1]$. Sea LP la relajación lineal de IP.

Definición 2.2: Los sistemas $Ax \leq b$ y $A'x \leq b'$ son sistemas 0-1 *equivalentes* si tienen las mismas soluciones para $x \in \{0,1\}^n$.

Definición 2.3: Dados los sistemas 0-1 equivalentes $Ax \leq b$ y $A'x \leq b'$, se dice que el sistema $A'x \leq b'$ es *tan fuerte como* $Ax \leq b$ si se verifica que $\{x / A'x \leq b'\} \subseteq \{x / Ax \leq b\}$ para $x \in [0,1]^n$.

Definición 2.4: Dados los sistemas 0-1 equivalentes $Ax \leq b$ y $A'x \leq b'$, se dice que el sistema $A'x \leq b'$ es *más fuerte que* $Ax \leq b$ si se verifica que $\{x / A'x \leq b'\} \subset \{x / Ax \leq b\}$ para $x \in [0,1]^n$.

Por definición, dos sistemas 0-1 equivalentes $Ax \leq b$ y $A'x \leq b'$ tienen las mismas soluciones enteras. Pero cuanto más fuerte sea uno de los dos sistemas más se restringe el espacio de soluciones lineales factibles y, por tanto, más se acerca el sistema a la envoltura convexa del conjunto de soluciones enteras. De hecho, el objetivo final al obtener un sistema fuerte, sea $Ax \leq b$ con $x \in [0,1]^n$, es que las restricciones del sistema definan facetas y, por tanto, definan la envoltura convexa del problema original. La solución óptima del problema LP será, en este caso, la solución óptima del problema original IP.

Ejemplo 2.1:

Sea el sistema siguiente:

$$\begin{aligned} 5x + 6y + 7z &\leq 11 \\ x, y, z &\in \{ 0;1 \} \end{aligned}$$

La relajación lineal del problema es:

$$\begin{aligned} 5x + 6y + 7z &\leq 11 \\ 0 &\leq x, y, z \leq 1 \end{aligned}$$

que geoméricamente representa el conjunto factible representado en la figura 2.1a.

Un sistema 0-1 equivalente al sistema anterior es el siguiente,

$$\begin{aligned} x + z &\leq 1 \\ y + z &\leq 1 \\ 0 &\leq x, y, z \leq 1 \quad x, y, z \in \{ 0;1 \} \end{aligned}$$

ya que no elimina ninguna solución entera del problema. Su relajación lineal geoméricamente representa el conjunto factible (representado superpuesto al anterior) que se muestra en la figura 2.1b:

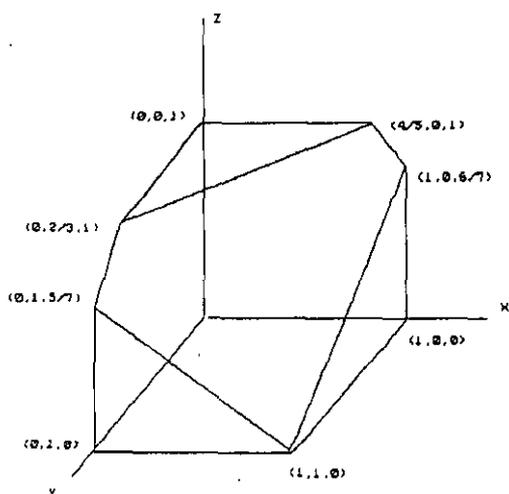


Figura 2.1a

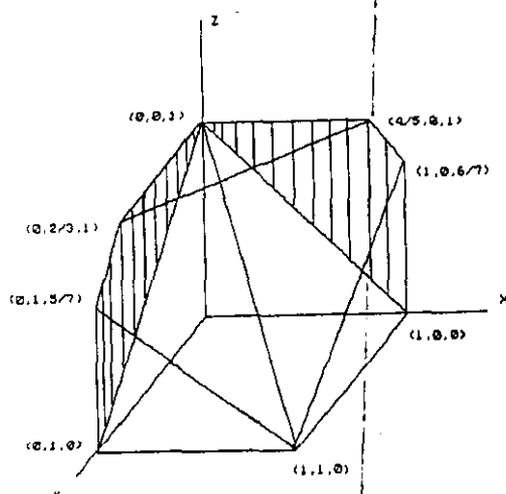


Figura 2.1b

Figura 2.1

Tenemos así dos formulaciones 0-1 equivalentes del mismo problema. Estas representaciones son tales que la segunda es más fuerte que la primera, ya que elimina soluciones lineales factibles para el primer problema relajado.

Además, en este caso, la segunda representación es tal que las restricciones introducidas definen facetas del politopo entero y, por tanto, los puntos extremos del conjunto factible relajado tienen coordenadas enteras.

Podemos observar que un problema 0-1 puede tener (y, muy frecuentemente, éste es el caso) varias formulaciones equivalentes según la definición 2.2. Tiene sentido, pues, analizar dichas formulaciones, así como la bondad de cada modelización en términos de "fortaleza" en relación

con las otras formulaciones (equivalentes) del problema.

2.2.2.- Diferentes modelos 0-1 equivalentes para el SOP con relaciones de precedencia

En un primer momento (ver Escudero, 1988), el SOP fue formulado como el problema de encontrar un camino hamiltoniano \mathcal{H} con peso total mínimo en un grafo de forma que se satisfagan las relaciones de precedencia. Estas relaciones de precedencia vienen dadas por el digrafo $P=(N,\Pi)$, donde $(i,j)\in\Pi$ significa que $i < j$ en todo camino hamiltoniano factible. Claramente, este grafo no puede contener ningún circuito y, además, si $(i,j),(j,k)\in\Pi$ entonces i tiene que preceder a k . Por tanto, se supone que P es un digrafo sin circuitos y cerrado transitivamente.

Como se ha mencionado anteriormente, existen diferentes modelos 0-1 para el SOP, todos ellos 0-1 equivalentes. A continuación se describen algunos modelos que se han planteado en la literatura (ver Ascheuer et al., 1993). Así mismo se resume el resultado de la experiencia computacional obtenida por lo que respecta a la bondad de cada modelo.

Modelo 2.2

Para esta primera formulación del SOP sea la siguiente notación:

- ←
 (a) $\Pi \equiv \{ (j,i)\in V\times V / (i,j)\in\Pi \}$
 →
 (b) $\Pi \equiv \{ (i,k)\in V\times V / \exists j \text{ con } (i,j),(j,k)\in\Pi \}$
 ← →
 (c) $\bar{A} \equiv A \setminus (\Pi \cup \Pi)$

Ningún camino factible \mathcal{H} puede contener un arco de $\bar{\Pi}$, pues, en caso contrario, se violaría una relación de precedencia, ni un arco de $\bar{\Pi}$, ya que si $(i,j),(j,k)\in\Pi$ entonces $i < j$ y $j < k$, y esto no sería posible si el arco (i,k) perteneciera al camino. \bar{A} será, por tanto, el conjunto de arcos cuya infactibilidad todavía no ha podido ser demostrada, si existe. Se considera pues el subgrafo $D=(V,\bar{A})$. Se consideran también los siguientes parámetros:

$$(d) \alpha_k \equiv 1 + |\{ i / (i,k) \in \Pi \}| \quad k \in V$$

$$(e) \beta_k \equiv n - |\{ j / (k,j) \in \Pi \}| \quad k \in V$$

$\alpha_k - 1$ (respectivamente $n - \beta_k$) es el mínimo número de nudos predecesores (resp. sucesores) que ha de tener el nudo k en cualquier camino hamiltoniano \mathcal{H} factible.

Se introducen los dos tipos siguientes de variables: Para cada arco $(i,j) \in \bar{A}$, x_{ij} es una variable 0-1 que indica si el arco (i,j) está en el camino hamiltoniano \mathcal{H} o no, tal que

$$x_{ij} = \begin{cases} 1, & \text{si } (i,j) \in (0, \dots, u) \\ 0, & \text{en caso contrario} \end{cases}$$

Es decir, $x_{ij} = 1$ si y sólo si $j = z(i)$

El segundo tipo de variables 0-1 son variables auxiliares ξ_{hk} para todo $h, k \in V$. Estas variables se utilizarán para modelizar las restricciones de precedencia, de forma que

$$\xi_{hk} = \begin{cases} 1, & \text{si el nudo } k \text{ se secuencia a nivel } h, \text{ con } \alpha_k \leq h \leq \beta_k \\ 0, & \text{en caso contrario} \end{cases}$$

El modelo queda, pues, formulado de la siguiente manera:

$$z^* = \text{Min } c^t x$$

s.a.

$$(1) x(\bar{A}) = n-1$$

$$(2) x(\delta^-(j)) \leq 1 \quad \forall j \in V$$

$$(3) x(\delta^+(j)) \leq 1 \quad \forall j \in V$$

$$(4) x_{ij} \geq 0 \quad \forall (i,j) \in \bar{A}$$

$$(5) x(W) \leq |W| - 1 \quad \forall W \subset V, 2 \leq |W| \leq n-1$$

$$(6) x_{ij} \in \{0,1\} \quad \forall (i,j) \in \bar{A}$$

$$(7) \sum_{k/\alpha_k \leq h \leq \beta_k} \xi_{kh} = 1 \quad \forall h \in V$$

$$(8) \sum_{\alpha_k \leq h \leq \beta_k} \xi_{kh} = 1 \quad \forall k \in V$$

$$(9) \sum_{\alpha_1 \leq h \leq \beta_1} h \xi_{1h} + 1 \leq \sum_{\alpha_j \leq h \leq \beta_j} h \xi_{jh} \quad \forall (i,j) \in \Pi \overset{\rightarrow}{\Pi}$$

$$(10) \xi_{1h} + \xi_{jh+1} \leq 1 \quad \forall (i,j) \in A \setminus \bar{A},$$

$$\max \{\alpha_1, \alpha_j - 1\} \leq h \leq \min \{\beta_1, \beta_j - 1\}$$

$$(11) \xi_{kh} \in \{0,1\} \quad \forall k \in V, \alpha_k \leq h \leq \beta_k$$

$$(12) \xi_{1h} + \xi_{jh+1} \leq 1 + x_{1j} \quad \forall (i,j) \in \bar{A},$$

$$\max \{\alpha_1, \alpha_j - 1\} \leq h \leq \min \{\beta_1, \beta_j - 1\}$$

Las restricciones (1)-(6) dan la formulación en programación 0-1 del problema de encontrar un camino hamiltoniano en el grafo $D=(V, \bar{A})$, análogamente al modelo 0-1 del ATSP. Las inecuaciones tipo (5) se llaman *restricciones de eliminación de subciclos*, SEC's (del inglés, *Subtour Elimination Constraints*, ver Dantzig, Fulkerson y Johnson, 1954).

Las restricciones (7)-(12) aseguran que se respeten las relaciones de precedencia. La restricción (7) obliga a que se secuencie un solo nudo en cada nivel, y la restricción (8) a que exista un solo nivel para cada nudo. La restricción (9) evita que se secuencien en orden inverso pares de

nudos unidos por relaciones de precedencia directas, y la restricción (10) hace que si $(i,j) \in \Pi$ y $(j,k) \in \Pi$ ó $(k,j) \in \Pi$, entonces el nudo k no se secuencie inmediatamente después que el nudo i . Finalmente, las restricciones (12) integran los submodelos (1)-(6) y (7)-(11).

Crítica al modelo

Después de trabajar computacionalmente con este modelo se observó que en un gran número de casos el intervalo de optimalidad, es decir, la desviación relativa $(z^H - z_{LR}) / z_{LR}$ era bastante grande, donde z^H representa el valor en la función objetivo de la solución que obtiene el algoritmo heurístico utilizado, y z_{LR} la cota inferior al valor óptimo z^* obtenida mediante la relajación lineal del modelo anterior.

Este intervalo se debe a que, o bien z^H o bien z_{LR} , o ambos, están lejos del verdadero valor óptimo del problema. Debido a la experiencia computacional con otro tipo de modelos se postula que el valor z^H es cercano, mientras que z_{LR} no. Surge de ahí la necesidad de encontrar formulaciones equivalentes para el problema, que sean más fuertes que la formulación anterior.

Modelo 2.3

En este segundo modelo se utilizan dos tipos de variables 0-1. Sean las variables x_{ij} tales que

$$x_{ij} = \begin{cases} 1, & \text{si } (i,j) \in (0, \dots, u) \\ 0, & \text{en caso contrario} \end{cases}$$

Sean las variables 0-1 y_{ij} para cada $(i,j) \in A$ variables auxiliares que se utilizarán para modelizar las relaciones de precedencia, tales que

$$y_{ij} = \begin{cases} 1, & \text{si } i < j \text{ en el camino factible } \mathcal{H} \\ 0, & \text{en caso contrario} \end{cases}$$

El modelo es el siguiente:

$$z^* = \text{Min } c^t x$$

s.a.

x satisface las restricciones (1)-(6) del modelo 2.2

$$(7) y_{ij} = 1 \quad \forall (i,j) \in \Pi$$

$$(8) y_{ij} + y_{ji} = 1 \quad \forall (i,j) \in A$$

$$(9) y_{ij} + y_{jk} + y_{ki} \leq 2 \quad \forall i,j,k \in V, i \neq j \neq k$$

$$(10) y_{ij} \geq 0 \quad \forall (i,j) \in A$$

$$(11) x_{ij} - y_{ij} \leq 0 \quad \forall (i,j) \in \bar{A}$$

Considerando las restricciones (8)-(10) y añadiendo la condición de integralidad se obtiene la formulación 0-1 del problema del ordenamiento lineal. (Ver Grötschel, Junger y Reinelt (1984)). En este caso las restricciones $y_{ij} \in \{0,1\}$ no son necesarias, ya que la restricción (11) las implica debido a la integralidad de las variables x_{ij} . Las restricciones (7)-(11) aseguran que se respetan las relaciones de precedencia.

Critica al modelo

Una ventaja del modelo 2.3 es que combina de una forma natural dos problemas de optimización combinatoria bien conocidos: el problema del camino hamiltoniano y el problema del ordenamiento lineal. Una desventaja es la utilización de las variables auxiliares y_{ij} .

Este modelo no parece ser bueno computacionalmente, ya que en gran número de casos el intervalo de optimalidad, utilizando las técnicas que se describen en este capítulo, sigue siendo grande. En el siguiente modelo se sustituirán las restricciones donde aparecen las variables y_{ij} por un

nuevo conjunto de restricciones, de tamaño exponencial en n .

Modelo 2.4

$$z^* = \text{Min } c^t x$$

s.a.

x satisface las restricciones (1)-(6) modelo 2.2

$$(12) \ x(j;W) + x(W) + x(W:i) \leq |W|$$

$$\forall (i,j) \in \Pi \text{ y } \forall W \text{ tal que } \emptyset \neq W \subseteq V \setminus \{i,j\}$$

Llamaremos a las inecuaciones de tipo (12) *restricciones que fuerzan la precedencia*, PFC's; (del inglés *Precedence Forcing Constraints*, ver Ascheuer et al. (1993) y Balas et al. (1994)).

Crítica al modelo

Este modelo tiene la ventaja de no utilizar variables auxiliares, y es un planteamiento más natural para el SOP. La experiencia computacional confirma este punto de vista. Los modelos 2.3 y 2.4 son más fuertes que el modelo 2.2. Sin embargo algoritmos implementados utilizando el modelo 2.4 proporcionan mejores cotas inferiores a la solución óptima que aquellos que utilizan el modelo 2.3. En la sección 2.4.1 se estudia el tratamiento del conjunto exponencial de condiciones de tipo (12).

2.2.3.- Modelo 0-1 para el SOP con relaciones de precedencia y acotaciones

En esta sección se considera el problema de obtener un camino hamiltoniano de peso mínimo de forma que se satisfagan las relaciones de precedencia, cuando existen acotaciones superiores al peso de cada subcamino hamiltoniano. Sea d el vector n -dimensional que proporciona estas cotas.

Para la resolución del problema es conveniente que el espacio de

soluciones lineales factibles sea la más restringido posible. Para ello, como se verá en la sección 2.3, puede ser conveniente introducir un nuevo vector n -dimensional, r , que proporciona cotas inferiores al peso de cada subcamino. En principio, en el problema que nos ocupa, $r_i = 1 \forall i=1, \dots, n$.

Se considera el nudo ficticio 0, con $p_0 = 0$, $d_0 = 0$, $\forall j \in V$, desde el que ha de comenzar cualquier camino. Se supone que $c_{0j} \geq 0$ recoge el costo de acondicionamiento en el caso de que el nudo j sea el primer nudo del camino hamiltoniano después del nudo 0.

Modelo 2.5

En este modelo se consideran las variables 0-1 x_{ij} , tal que

$$x_{ij} = \begin{cases} 1, & \text{si } (i, j) \in (0, \dots, u) \\ 0, & \text{en caso contrario} \end{cases}$$

El segundo tipo de variables a utilizar son las variables continuas y_i , que dan el peso acumulado del vértice i en el camino hamiltoniano \mathcal{H} .

En el caso que se está considerando, donde el peso acumulado de un vértice puede estar acotado inferiormente, este peso, que depende del conjunto de nudos secuenciados antes que él en \mathcal{H} , viene dado por:

$$y_i = \max \{r_i - 1, y_{a(i)} + c_{a(i)i}\} + p_i, \quad \text{con } y_0 = 0 \quad (2.1)$$

El modelo es el siguiente:

$$z^* = \text{Min } c^t x$$

s.a.

$$(1) \ x(\bar{A}) = n-1$$

$$(2) \ x(\delta^-(j)) \leq 1 \quad \forall j \in V$$

$$(3) \quad x(\delta^+(j)) \leq 1 \quad \forall j \in V$$

$$(4) \quad 0 \leq x_{ij} \leq 1 \quad \forall (i,j) \in \bar{A}$$

$$(5) \quad x(W) \leq |W|-1 \quad \forall W \subset V, 2 \leq |W| \leq n-1$$

$$(6) \quad x_{ij} \in \{0;1\} \quad \forall (i,j) \in \bar{A}$$

$$(7a) \quad m_{ij}(1-x_{ij}) \leq (y_i + T_{ij}) - y_j \leq M_{ij}(1-x_{ij}) \quad \forall (i,j) \in \bar{A}$$

si $r_i = 1 \quad \forall i \in V$

$$(7b) \quad (y_i + T_{ij}) - y_j \leq M_{ij}(1-x_{ij}) \quad \forall (i,j) \in \bar{A}$$

si $\exists i \in V / r_i > 1$

$$(8) \quad y_i + S_{ij} \leq y_j \quad \forall (i,j) \in \Pi$$

$$(9) \quad e_i \leq y_i \leq d_i \quad \forall i \in V$$

donde

$$e_i = r_i - 1 + p_i$$

$$M_{ij} = (d_i + T_{ij}) - e_j$$

$$m_{ij} = (p_i + T_{ij}) - d_j$$

S_{ij} es la longitud del camino mínimo desde el nudo i hasta el nudo j en el digrafo $G=(V,A)$, considerando T_{ij} como la distancia de i a $j \quad \forall (i,j) \in A$.

Las restricciones (1)-(6) son las mismas que en los modelos anteriores, y proporcionan un camino hamiltoniano \mathcal{H} . La restricción (7a) obliga a que $y_j = y_i + T_{ij}$ si $x_{ij} = 1$, es decir, si el arco (i,j) pertenece al camino \mathcal{H} , en el caso en que todas las cotas inferiores sean iguales a 1. Esta restricción es redundante para $x_{ij} = 0$. La restricción

(7b) obliga a que $y_j \geq y_i + T_{ij}$ si $x_{ij} = 1$ en el caso de que alguna cota inferior sea superior a 1, y es redundante para $x_{ij} = 0$. La restricción (7b) es necesaria dada la definición (2.1) de y_j .

Los valores de M_{ij} y m_{ij} son valores suficientemente grandes en valor absoluto como para no eliminar ninguna solución entera del problema, pero tales que utilizan la información disponible sobre el problema para acotar al máximo el problema lineal relajado y, por tanto, hacerlo más fuerte.

Las restricciones (8) son las PFC's, ya que obligan a secuenciar el nudo i antes que el nudo j si $(i,j) \in \Pi$. Las restricciones (9) acotan el peso acumulado de cada nudo.

Llamaremos a las inecuaciones de tipo (7) y (9) *restricciones que fuerzan las acotaciones*, DFC's; (del inglés *Due date Forcing Constraints*).

Este modelo puede ser planteado matemáticamente sin necesidad de utilizar el vector r , ya que hasta el momento se considera que $r_j = 1 \forall j \in V$ y, por tanto, la condición (7b) es vacía. Basta para ello con eliminar la condición (7b), y reescribir las condiciones (7) y (9) en la forma

$$(7') \quad m_{ij}(1-x_{ij}) \leq (y_i + T_{ij}) - y_j \leq M_{ij}(1-x_{ij}) \quad \forall (i,j) \in \bar{A}$$

$$(9') \quad p_i \leq y_i \leq d_i \quad \forall i \in V$$

donde $M_{ij} = (d_i + T_{ij}) - p_j$

$$m_{ij} = (p_i + T_{ij}) - d_j$$

Sin embargo, como veremos en la sección 2.3, utilizando la primera formulación es posible reforzar las restricciones de forma que se disminuya el conjunto de soluciones lineales factibles sin eliminar ninguna posible solución factible 0-1, lo cual es muy conveniente si se trata de resolver el problema utilizando la metodología que en este

capítulo se desarrolla.

2.3.- PREPROCESO

Una fase importante de un procedimiento de optimización basado en planos de corte consiste en analizar el problema para intentar descubrir alguna estructura que ayude a descomponerlo, o a reducir su tamaño, o a hacer más fuerte la formulación del IP convirtiendo algunas desigualdades en igualdades o fijando algunas variables a un valor determinado, etc. Esto es igualmente necesario para reducir el esfuerzo computacional requerido para la obtención de una solución factible para el problema. Se llama *preproceso* a la realización de todas estas operaciones.

Utilizando como base la formulación IP para el SOP dada en el modelo 2.5 de la sección 2.2.3, se proponen las siguientes operaciones:

2.3.1.- Eliminación de arcos del conjunto A

Dado el grafo $G = (V,A)$, con V conjunto de nudos y A conjunto de arcos, se debe eliminar del conjunto A el arco (i,j) si se ha detectado que i no puede ser el predecesor inmediato de j en ningún camino factible \mathcal{H} . (Es decir, si $a(j) \neq i \quad \forall \mathcal{H}$ factible). Las líneas maestras del procedimiento son las siguientes:

- a. Fijar $x_{ij} = 0$ (es decir, eliminar el arco (i,j) del conjunto de arcos A) si se verifica alguna de las condiciones siguientes:

$$(j,i) \in \Pi \quad (2.2)$$

$$\exists k \in V / (i,k), (k,j) \in \Pi \quad (2.3)$$

$$e_i + T_{ij} > d_j \quad (2.4)$$

$$\exists k \in V \text{ tal que o bien } x_{ik} = 1 \text{ o bien } x_{kj} = 1. \quad (2.5)$$

- b. Fijar $x_{ij} = 1$ (es decir, eliminar el arco (i,j) y convertir los vértices i y j en un sólo nudo de peso $p_i + c_{ij} + p_j$) si se satisface

alguna de las condiciones siguientes:

$$x_{kj} = 0 \quad \forall (k,j) \in A \text{ tal que } k \neq i \text{ y } \exists l / (l,j) \in \Pi \quad (2.6)$$

$$x_{ik} = 0 \quad \forall (i,k) \in A \text{ tal que } k \neq j \text{ y } \exists l / (i,l) \in \Pi \quad (2.7)$$

Fijar $x_{ij} = 0$ si se verifica (2.2) ó (2.3) equivale a pasar del grafo $G = (V,A)$ al subgrafo factible $D = (V,\bar{A})$ (ver modelo 2.2, sección 2.2.2). La condición (2.4), donde $e_i = r_i - l + p_i$, implica que $j < i$ en cualquier camino factible \mathcal{H} . (2.5) se deduce de las restricciones (2) y (3) del modelo. La condición (2.6) (resp. (2.7)) indica que si el nudo j (resp. i) no puede ser el nudo inicial (resp. final) de ningún camino factible y $a(j) \neq k \quad \forall k \neq i$ (resp. $z(i) \neq k \quad \forall k \neq j$) entonces el arco (i,j) debe formar parte de cualquier camino factible \mathcal{H} .

2.3.2.- Reducción del intervalo (r_i, d_i) , $\forall i \in V$.

Cuanto menor sea este intervalo más fuertes son las restricciones (7) y (9) del modelo (2.5). Este procedimiento sólo es efectivo si existen relaciones de precedencia, es decir, si $\Pi \neq \emptyset$.

De forma recursiva se actualizan los vectores r y d , tal que se incrementa r_j y reduce $d_j \quad \forall j \in V$ de la forma siguiente:

$$* r_j = \max \{ r_j, \max \{ r_i + p_i + S_{ij}, \quad \forall i / (i,j) \in \Pi \} \} \quad \forall j \in V$$

donde S_{ij} es la distancia mínima del nudo i al nudo j en el digrafo

$G = (V,\bar{A})$, siendo T_{ab} la distancia entre los nudos a y b .

Observación 2.1.: Esta nueva cota es la más ajustada que se puede conseguir, pero es una cota costosa de calcular cuando el cardinal del conjunto Π es grande, ya que supone resolver $|\Pi|$ problemas de camino mínimo. En esos casos se utiliza la siguiente relajación alternativa:

$$* r_j = \max \{ r_j, R1, R2 \} \quad \forall j \in V \quad (2.8)$$

donde

$$R1 \equiv \max \{ r_i + p_i + \underline{C}_i \quad \forall i / (i,j) \in \Pi \} \quad (2.9)$$

$$R2 \equiv \underline{R}_i^j + \sum_{i / (i,j) \in \Pi} (p_i + \underline{C}_i) \quad (2.10)$$

y

$$\underline{C}_i \equiv \min \{ c_{ik}, \quad \forall k / (i,k) \in \bar{A} \} \quad (2.11)$$

$$\underline{R}_i^j \equiv \min \{ r_i, \quad \forall i / (i,j) \in \Pi \} \quad (2.12)$$

$$* d_i = \min \{ d_i, D1, D2 \} \quad \forall i \in V \quad (2.13)$$

donde

$$D1 \equiv \min \{ d_j - (p_j + \underline{C}_j) \quad \forall j / (i,j) \in \Pi \} \quad (2.14)$$

$$D2 \equiv \underline{D}_i - \sum_{j / (i,j) \in \Pi} (p_j + \underline{C}_j) \quad (2.15)$$

y

$$\underline{C}_i^j \equiv \min \{ c_{kj}, \quad \forall k / (k,j) \in A \} \quad (2.16)$$

$$\underline{D}_i \equiv \max \{ d_j, \quad \forall j / (i,j) \in \Pi \} \quad (2.17)$$

Observación 2.2: Siempre ha de verificarse que $r_i \geq 1$, $d_j \leq \bar{z}$, donde \bar{z} es la mejor cota superior disponible del peso acumulado del último nudo que se secuencie.

2.3.3.- Eliminación de arcos del conjunto Π

Se debe eliminar del conjunto de arcos Π todo arco (i,j) que

verifique

$$d_i - (C_i^1 + p_i) < r_j + p_j \quad (2.18)$$

ya que si esta condición se verifica entonces la restricción (8) del modelo 2.5 es redundante incluso en la relajación lineal y, por tanto, $(i,j) \in \Pi$ también es redundante.

2.3.4.- Adición de arcos al conjunto Π

Sean los nudos $i, j \in V$ tal que se cumplen las siguientes condiciones:

- (a) $(i,j), (j,i) \notin \Pi$
- (b) $c_{ij} = c_{ji} = 0$
- (c) $c_{ik} = c_{jk}$ y $c_{ki} = c_{kj} \quad \forall k \in V \setminus \{i,j\}$
- (d) $c_{uv} \leq c_{uj} + c_{jv} \quad \forall u, v \in V \setminus \{i,j\}, u \neq v$
- (e) j no tiene ningún predecesor ni sucesor en P .

Entonces se puede añadir el arco (i,j) o el arco (j,i) al conjunto Π , de forma que al menos una solución óptima del SOP original es óptima para el nuevo problema. Esta observación puede ayudar en algunos casos a reducir sustancialmente el tamaño del problema.

Observación 2.3: Se deben ejecutar los procedimientos 2.3.1, 2.3.2, 2.3.3 y 2.3.4 secuencialmente, y repetir esta secuencia hasta que no se obtenga ningún nuevo cambio.

2.3.5.- Transformación de desigualdades en igualdades

Dados $G = (V,A)$ y $P = (N,\Pi)$ definimos:

- (a) $V_1 = \{ v \in V / \exists (v,j) \in \Pi, j \neq v \}$
- (b) $V_2 = \{ v \in V / \exists (i,v) \in \Pi, i \neq v \}$

es decir, V_1 es el conjunto de nudos que tienen sucesores en Π y V_2 es el conjunto de nudos que tienen predecesores en Π .

Entonces las desigualdades (2) y (3) del modelo se pueden transformar en:

$$(2a) \quad x(\delta^-(j)) = 1 \quad \forall j \in V_2$$

$$(2b) \quad x(\delta^-(j)) \leq 1 \quad \forall j \in V \setminus V_2$$

$$(3a) \quad x(\delta^+(j)) = 1 \quad \forall j \in V_1$$

$$(3b) \quad x(\delta^+(j)) \leq 1 \quad \forall j \in V \setminus V_1$$

Este tipo de análisis se efectúa no sólo en el preproceso, sino en cualquier fase del algoritmo general cuando algunas variables se fijan a cero o a uno.

2.3.6.- Test de infactibilidad

Un problema dado será infactible si se verifica alguna de las 10 condiciones siguientes:

- 1) $B = \emptyset$, donde $B = \{ (0,b) \}$
con $b \in V / T_{ob} \leq d_b$ y $\exists a \in V / (a,b) \in \Pi$
- 2) $\exists i \in V / e_i > d_i$
- 3) $\exists (a,b) \in \Pi$ y $\bar{\exists} (a,j) \in A$
- 4) $|L^h| > 1$, con $L^h = \{ a / \exists (a,b) \in \Pi \text{ y } \exists (i,a) \in A \}$
- 5) $\exists (a,b) \in \Pi$ y $\bar{\exists} (i,b) \in A$
- 6) $|L^t| > 1$, con $L^t = \{ b / \exists (a,b) \in \Pi \text{ y } \exists (b,j) \in A \}$
- 7) $a \in L^h$ y $a \notin B$
- 8) $\exists a, b \in V / (a,b), (b,a) \in \Pi$
- 9) $\exists a, b, c \in V / (a,b), (b,c), (c,a) \in \Pi$
- 10) $G = (V, A)$ no es conexo

Si se verifica la condición 1 ningún nudo podría ser el nudo inicial del camino \mathcal{H} . La condición 2 indica que una cota superior no puede satisfacerse. La condición 3 identifica un nudo que debe ser predecesor de otro y al mismo tiempo último nudo del camino, y la condición 5 identifica un nudo que debe ser sucesor de otro y al mismo tiempo nudo inicial del camino. La condición 4 (resp. 6) implica que más de un nudo debe ser nudo inicial (resp. final) del camino. La condición 7 identifica un nudo que debe ser inicial del camino pero al que no se puede llegar desde el nudo 0. Finalmente las condiciones 8 y 9 detectan ciclos en Π .

2.4.- OBTENCION DE COTAS INFERIORES. SOP CON RELACIONES DE PRECEDENCIA

Se plantea el problema de encontrar un camino hamiltoniano \mathcal{H} con peso total mínimo en un grafo de forma que se satisfagan las relaciones de precedencia dadas por el digrafo $P=(N,\Pi)$. (Se supone, por tanto, que no hay restricciones de cota superior en el peso acumulado de ningún nudo.)

Para ello se desarrollan algoritmos de planos de corte, que siguen el esquema general para este tipo de algoritmos, aplicandolos al modelo 2.4 de la sección 2.2.2. (Ver algoritmo 2.2)

2.4.1. Algoritmos de separación. Metodología.

En el desarrollo de un algoritmo de planos de corte juega un papel principal la utilización de procedimientos que verifiquen si un punto dado (generalmente la solución óptima de la última relajación lineal del problema) satisface todas las desigualdades correspondientes a un tipo de restricciones determinado, o, en caso contrario, identifiquen al menos una desigualdad de este tipo que sea violada por dicho punto.

Ese tipo de procedimientos reciben el nombre de *algoritmos de separación*. Nuestra investigación se centra en algoritmos de separación que se ejecuten en tiempo polinomial.

En las dos secciones siguientes se describen algoritmos de separación polinomiales para las restricciones de eliminación de subciclos (SEC'S) y para las restricciones que fuerzan la precedencia (PFC'S). En ambos casos tenemos un número de desigualdades exponencial en n .

2.4.2. Identificación de condiciones de eliminación de subciclos

La entrada del algoritmo es un punto $x \in \mathbb{Q}^{\bar{A}}$, con $x_{ij} \geq 0 \forall (i,j) \in \bar{A}$. El algoritmo es válido para situaciones más generales que las consideradas en el modelo 2.4. Así, no se exige a x satisfacer las restricciones (1)-(3),(6) del modelo.

El algoritmo garantiza que x satisface todas las desigualdades de la forma

$$x(W) \leq |W| - 1 \quad \forall W \subseteq V, \quad 2 \leq |W| \leq n \quad (2.19)$$

o bien proporciona un conjunto $W \subseteq V$, $2 \leq |W| \leq n$ tal que $x(W) > |W| - 1$. De hecho proporciona un conjunto W tal que la diferencia $x(W) - (|W| - 1)$ sea lo mayor posible, es decir, identifica la SEC más violada.

Para ello, se construye el digrafo auxiliar $D_0 = (V_0, A_0)$, de la forma siguiente (ver figura 2.2):

$$\begin{aligned} V_0 &= V \cup \{0\}, \quad \text{donde } 0 \text{ es un nudo nuevo} \\ A^x &= \{(i,j) \in \bar{A} / x_{ij} > 0\} \\ A_0 &= A^x \cup \{(0,v) / v \in V\} \cup \{(j,i) / (i,j) \in A^x \text{ y } (j,i) \notin A^x\} \end{aligned}$$

Se resuelve el problema de separación para las SEC's reduciendolo a una secuencia de problemas de corte mínimo. Para convertir el problema en uno de flujo sobre un digrafo se introducen capacidades c_{ij}^0 para los arcos de D_0 de la forma siguiente:

$$\text{Sea } \zeta_j = x(\delta^-(j)) + x(\delta^+(j)) \quad \forall j \in V \quad (2.20)$$

$$\text{Se define } c_{0j}^0 = 1 - (1/2)\zeta_j + M \quad \forall j \in V \quad (2.21)$$

con M entero positivo elegido de forma que $c_{0j}^0 \geq 0 \quad \forall j \in V$

$$\text{Se define } c_{ij}^0 = c_{ji}^0 = 1/2 (x_{ij} + x_{ji}) \quad \forall j \in V \quad (2.22)$$

Observación 2.3: si $(j,i) \notin A^x$ para algún $(i,j) \in A^x$, se supone que x_{ji} vale cero.

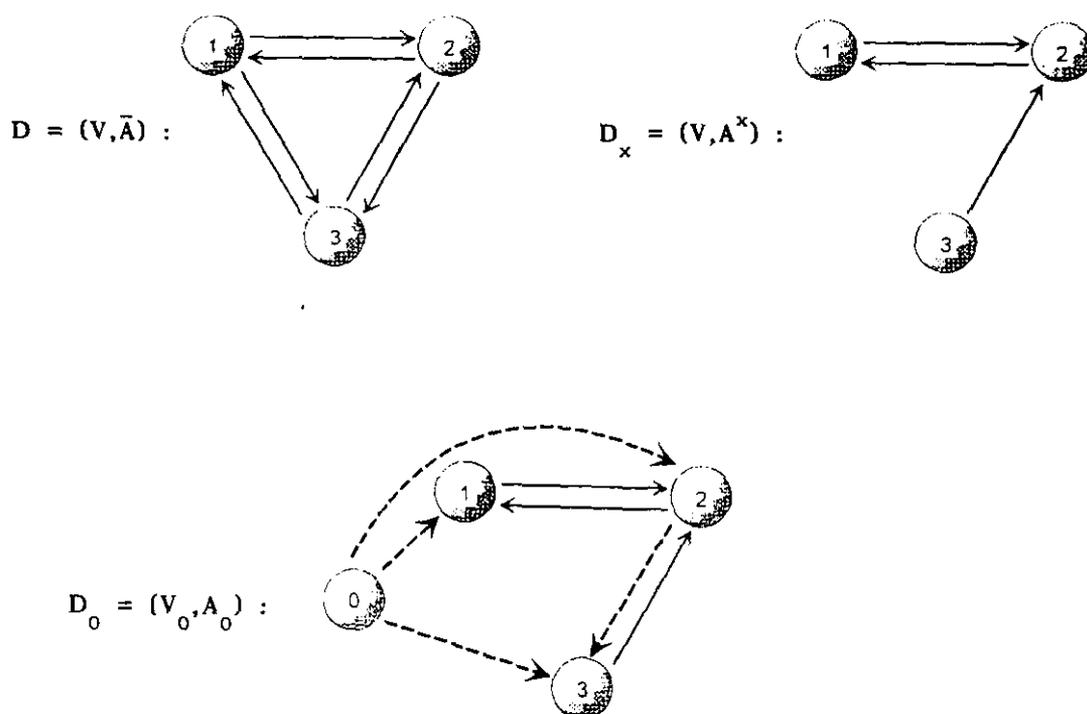


Figura 2.2

A continuación se introducen n digrafos auxiliares, uno para cada nudo de V , que son pequeñas modificaciones de D_0 . Para cada $k \in V$ se define $D_k = (V_k, A_k)$ con capacidades c_{ij}^k , de la forma siguiente:

(a) $V_k = V_0$

- (b) $A_k = A_0 \dot{\cup} B_k$ donde $B_k = \{(v,k) / v \in V \setminus \{k\}\}$
 (c) $c_{ij}^k = c_{ij}^0$ $\forall (i,j) \in A_0$
 (d) $c_{vk}^k = M$ $\forall (v,k) \in B_k$

Observación 2.4: En (b) $\dot{\cup}$ significa unión disjunta. Por tanto, si A_0 contiene un arco de B_k se añade un arco paralelo. (Ver figura 2.3).

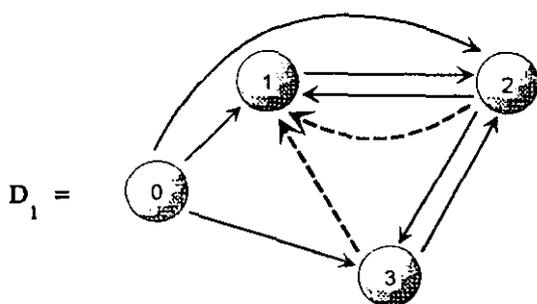


Figura 2.3

Algoritmo 2.1: Algoritmo de separación para las restricciones de eliminación de subciclos.

Entrada: Un punto $x \in \tilde{Q}^A$ tal que $x_{ij} \geq 0$

Salida: Al menos un conjunto de nudos, sea W , de cardinal entre 2 y n tal que x viola su SEC correspondiente, o la información de que no existe ningún conjunto de este tipo.

Algoritmo: Para cada $k \in V$:

Paso 1.- Construir el digrafo $D_k = (V_k, A_k)$ con capacidades c_{ij}^k tal como se ha indicado anteriormente.

Paso 2.- Utilizar un algoritmo de flujo máximo para determinar un

(0,k)-corte $\delta^-(W_k)$, es decir, un corte que separe 0 y k, y tal que $k \in W_k$, $0 \notin W_k$, de forma que la capacidad del corte $c^k(\delta^-(W_k))$ sea la menor posible.

Paso 3.- Si $c^k(\delta^-(W_k)) < nM + 1$ entonces x viola la SEC $x(W_k) \leq |W_k| - 1$. Si $c^k(\delta^-(W_k)) \geq nM + 1 \forall k \in V$ entonces x satisface todas las SEC's.

Teorema 2.1: Si la capacidad mínima de un (0,k)-corte en $D_k \forall k \in V$ no es menor que $nM + 1$, entonces x satisface todas las desigualdades del tipo (2.19). Si hay un (0,k)-corte $\delta^-(W_k)$ para algún $k \in V$, tal que $W_k \subseteq V$, $2 \leq |W_k| \leq n$, con $c^k(\delta^-(W_k)) \leq nM + 1$, entonces $x(W_k) > |W_k| - 1$.

Demostración: (Ascheuer et al., 1993)

En primer lugar se prueba que la capacidad de cualquier corte $\delta^-(W_k)$ en el digrafo D_k , con $0 \notin W_k$, $k \in W_k$ es la diferencia $|W_k| - (x(W_k) + nM)$. En efecto,

$$\begin{aligned}
 c^k(\delta^-(W_k)) &= \sum_{w \in W_k} c_{0w}^k + \sum_{v \in V \setminus W_k} \sum_{w \in W_k} c_{vw}^k + \sum_{v \in V \setminus W_k} c_{vk}^k = \\
 &= \sum_{w \in W_k} c_{0w}^0 + \sum_{v \in V \setminus W_k} \sum_{w \in W_k} c_{vw}^0 + \sum_{v \in V \setminus W_k} c_{vk}^k = \\
 &= \left(|W_k| - 1/2 \sum_{w \in W_k} \zeta_w + M|W_k| \right) + 1/2 \sum_{(v,w) \in \delta^-(W_k)} (x_{vw} + x_{wv}) + |V \setminus W_k| M = \\
 &= nM + |W_k| - 1/2 \left(\sum_{(v,w) \in \delta^-(W_k)} (x_{vw} + x_{wv}) + 2 \sum_{v,w \in A(W_k)} (x_{vw} + x_{wv}) \right) +
 \end{aligned}$$

$$\begin{aligned}
 & + 1/2 \sum_{(v,w) \in \delta^-(W_k)} (x_{vw} + x_{wv}) = \\
 & = |W_k| - x(W_k) + nM.
 \end{aligned}$$

Por tanto, se verifica la desigualdad $x(W_k) \leq |W_k| - 1$ si y sólo si $c^k(\delta^-(W_k)) \geq 1 + nM$.

Falta demostrar que si existe un corte $\delta^-(W_k)$ con capacidad menor que $1 + nM$, entonces $|W_k| \geq 2$. En efecto, como $k \in W_k$, $|W_k| \geq 1$. Si $W_k = \{k\}$ entonces la capacidad del corte sería $1 + nM$. Por tanto $c^k(\delta^-(W_k)) < nM + 1$ implica $|W_k| \geq 2$.

Además, por construcción, $W_k = V$ es una solución posible. ■

Conclusión:

El algoritmo de separación arriba indicado para las SEC's más las condiciones de no negatividad se puede ejecutar llamando n veces a un algoritmo de flujo máximo. Por tanto es un algoritmo que se ejecuta en tiempo polinomial. Recomendamos la utilización del algoritmo de flujo máximo debido a Goldberg y Tarjan (1988). Para una panorámica sobre algoritmos de flujo máximo ver Ahuja, Magnanti y Orlin (1989).

Aplicación del algoritmo

El algoritmo anterior contempla situaciones más generales que las necesarias en el SOP.

Si se parte de que el vector $x \in \mathbb{Q}^A$ satisface las restricciones (1) y (4) del modelo 2.4, entonces el conjunto de nudos cuya SEC es violada

verifica $|W_k| \leq n-1$. Esto es así ya que $c^k(\delta^-(W_k)) = |V| - x(V) + nM = nM + 1$, y por tanto $W_k \subset V$ como se deduce del teorema 2.1.

Si además suponemos que el punto x satisface las restricciones (2) y (3) del modelo, entonces el problema de separación se convierte en un problema de corte mínimo en un grafo no dirigido, y se pueden utilizar algoritmos como el de Gomory-Hu para obtener un corte de capacidad mínima. (Ver Padberg y Rinaldi, 1990).

En efecto, si el punto x satisface las restricciones (2) y (3), entonces $\zeta_j \leq 2 \quad \forall j \in V$, y por tanto según la expresión (2.21) basta tomar $M = 0$ para que $c_{0j}^0 \geq 0$. Esto implica que los conjuntos de arcos B_k y, por tanto, los digrafos auxiliares D_k para $k \in V$ no son necesarios. Además se puede transformar el digrafo D_0 en el digrafo simétrico $D' = (V_0, A')$ con capacidades c'_{ij} haciendo:

- (a) $A' = A_0 \cup \{(v,0) / v \in V\}$
- (b) $c'_{ij} = c'_{ji} = c_{ij}^0 = c_{ji}^0 = 1/2 (x_{ij} + x_{ji}) \quad \forall (i,j) \in A_0$
- (c) $c'_{v0} = c'_{0v} \quad \forall v \in V$

Sea $G'' = (V_0, E)$ el grafo no dirigido inducido por $D' = (V_0, A')$ con capacidades c''_{ij} definido de la forma siguiente:

- (a) $E = \{ \{ij\} / (i,j) \in A' \}$
- (b) $c''_{ij} = c'_{ij} = c'_{ji} \quad \forall (ij) \in E$

G'' tiene la propiedad de que $c''(\delta(W)) = c'(\delta^+(W)) = c'(\delta^-(W))$ para cualquier $W \subseteq V$, donde $\delta^+(W) = \delta^-(V \setminus W)$. Por tanto, un corte $\delta(W)$ en G'' con capacidad mínima $c''(\delta(W))$ corresponde a un corte de capacidad mínima $\delta^-(W)$ en D' y viceversa.

Por tanto el problema de separación para las SEC's, suponiendo que el punto inicial verifica las condiciones (2),(3),(4) del modelo, puede

resolverse utilizando cualquier algoritmo que determine un corte de capacidad mínima en un grafo no dirigido.

2.4.3.- Identificación de condiciones que preservan relaciones de precedencia

Se plantea el problema de separación para las restricciones que fuerzan la precedencia (PFC's),

$$x(j:W) + x(W) + x(W:i) \leq |W| \quad \forall (i,j) \in \Pi \text{ y } \emptyset \neq W \subseteq V \setminus \{i,j\} \quad (2.23)$$

Como en el apartado anterior, se reducirá el problema de separación a una serie de problemas de corte mínimo.

La entrada del algoritmo es un punto $x \in \mathbb{Q}^{\bar{A}}$, con $x_{ij} \geq 0 \quad \forall (i,j) \in \bar{A}$. El algoritmo se pregunta si x viola alguna restricción del tipo (2.23), y en caso afirmativo da como salida alguna restricción violada. Para ello, se construye para cada arco $(i,j) \in \Pi$ un problema de corte mínimo del que se obtiene la información de si la PFC (2.23) es violada para algún conjunto $\emptyset \neq W \subseteq V \setminus \{i,j\}$.

Dado el digrafo de precedencias $P = (N, \Pi)$, para cada arco $(i,j) \in \Pi$ se introduce un nuevo digrafo $D_{ij} = (V_{ij}, A_{ij})$ con capacidades d_{kl}^{ij} . Este digrafo viene definido por los siguientes parámetros (ver figura 2.4):

$$(a) V_{ij} = (V \setminus \{i,j\}) \cup \{v_{ij}\} \quad \text{donde } v_{ij} \text{ es un nudo nuevo}$$

$$(b) A^x = \{ (i,j) \in \bar{A} \mid x_{ij} > 0 \}$$

$$(c) A_{ij} = \{ (k,l) \mid (k,l) \in A^x, k,l \notin \{i,j\} \} \cup \\ \cup \{ (v_{ij},l) \mid (j,l) \in A^x, l \notin \{i,j\} \} \cup \\ \cup \{ (k,v_{ij}) \mid (k,i) \in A^x, k \notin \{i,j\} \}$$

$$(d) d_{kl}^{ij} = x_{kl} \quad \forall (k,l) \in A_{ij} \cap A^x$$

$$(e) \quad d_{v_{ij}}^{ij} = x_{jl} \quad \forall (j,l) \in A^x$$

$$(f) \quad d_{k v_{ij}}^{ij} = x_{kl} \quad \forall (k,i) \in A^x$$

Por tanto el digrafo D_{ij} se obtiene a partir de $D^x = (V, A^x)$ eliminando todos los arcos que terminan en el nudo j , todos los arcos que salen del nudo i y todos los arcos entre los nudos i y j y, finalmente, reemplazando los nudos i y j por el nuevo nudo v_{ij} . Las capacidades de los arcos son los valores de las componentes positivas del vector x .

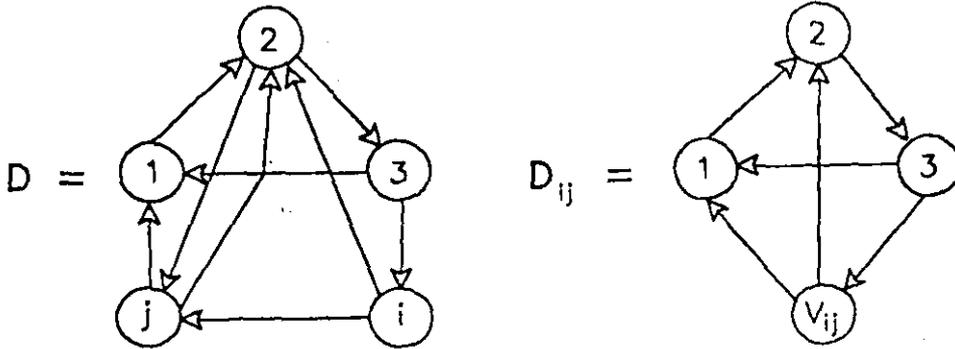


Figura 2.4

Las restricciones que fuerzan las relaciones de precedencia relativas al arco $(i,j) \in \Pi$ y al digrafo $D = (V, \bar{A})$,

$$x(j:W) + x(W) + x(W:i) \leq |W| \quad (2.24)$$

pueden escribirse, en términos de la transformación de D en D_{ij} , como:

$$x(W \cup \{v_{ij}\}) \leq |W| = |W \cup \{v_{ij}\}| - 1 \quad (2.25)$$

Por tanto, para contrastar si se verifica la PFC relativa al arco $(i,j) \in \Pi$ es preciso contrastar si el vector x verifica la SEC

$$x(\tilde{W}) \leq |\tilde{W}| - 1 \quad \forall \tilde{W} \subseteq V_{ij}, v_{ij} \in \tilde{W} \quad \text{y} \quad 2 \leq |\tilde{W}| \leq n-1$$

para el digrafo D_{ij} .

Si existe un conjunto de nudos $\tilde{W} \subseteq V_{ij}$ con $v_{ij} \in \tilde{W}$, $2 \leq |\tilde{W}| \leq n-1$ tal que $x(\tilde{W}) > |\tilde{W}| - 1$ entonces para el conjunto $W = \tilde{W} \setminus \{v_{ij}\}$ se satisface $x(j;W) + x(W) + x(W:i) > |W|$. Si no existe ningún \tilde{W} con esa propiedad entonces se satisfacen todas las restricciones de la forma (2.24) relativas al arco $(i,j) \in \Pi$.

Repitiendo este procedimiento $\forall (i,j) \in \Pi$ se resuelve el problema de separación para las restricciones (2.23).

Teorema 2.2: El problema de separación para las restricciones (2.25) se puede transformar en un problema de identificación de condiciones de eliminación de subciclos (resuelto en el apartado 2.4.2). Por tanto este problema se resuelve en tiempo polinomial.

Demostración:

Sea $\sigma \equiv v_{ij}$. Utilizando una versión simplificada del algoritmo 2.1, en el paso 1 del algoritmo normalmente sólo se tiene que construir el digrafo auxiliar $D_\sigma = (V_\sigma, A_\sigma)$ con capacidades c^σ (asociado al nudo comprimido σ) a partir del digrafo $D_{ij} = (V, A_{ij})$ con capacidades d^{ij} , y ejecutar los pasos 2 y 3 del algoritmo para este caso.

Si en el paso 3 se identifica un conjunto de nudos $\tilde{W} \subseteq V_\sigma$ con $\sigma \in \tilde{W}$, $0 \notin \tilde{W}$, tal que $c^\sigma(\delta^-(\tilde{W})) < 1 + (n-1)M$ entonces se verifica

$$x(\tilde{W}) > |\tilde{W}| - 1$$

y por tanto se viola la PFC asociada. En caso contrario el vector x satisface todas las PFC's.

Falta por comprobar que se verifica $|\tilde{W}| \geq 2$. Pero esto es así ya que

$$\sigma \in \tilde{W} \quad \text{y} \quad c^\sigma(\delta^-(\sigma)) = 1 + (n-1)M.$$

Por tanto queda demostrado que el problema de separación para las restricciones que fuerzan la precedencia se puede resolver en tiempo polinomial $\forall x \in Q^A (x \geq 0)$ ■

Si además el punto x dado satisface las restricciones (2) y (3) del modelo, entonces se puede hacer $M = 0$, lo que simplifica el algoritmo tal como se ha indicado en la sección 2.4.2.

2.4.4.- Algoritmo de planos de corte. Lineas generales

Las ideas básicas del algoritmo de planos de corte utilizado para resolver el problema son las siguientes:

Algoritmo 2.2

Entrada: Matriz $n \times n$ de coeficientes c_{ij} y digrafo acíclico de precedencias $P = (N, \Pi)$.

Paso 1: Calcular la envoltura transitiva cerrada del digrafo acíclico P para obtener el digrafo de precedencias inicial.

Paso 2: Preproceso (Sección 2.3.4). Se intentan añadir relaciones de precedencia. Si se añade alguna volver al paso 1. Repetir hasta que no pueda añadirse ninguna precedencia. El digrafo final será $P=(V, \Pi)$.

Paso 3: Preproceso. (Secciones 2.3.1, 2.3.3, 2.3.5). Se intentan eliminar arcos del conjunto A y del conjunto Π , fijar algunas variables, y se convierten algunas desigualdades en igualdades.

Paso 4: Cota superior inicial. Obtener una cota superior inicial de la solución óptima. La aplicación de un algoritmo heurístico proporciona la primera cota superior, con valor \bar{z} en la función objetivo.

Paso 5: Primera relajación del problema. Obtener la relajación inicial del problema, considerando el problema de asignación asociado (LPAP), integrado por las restricciones (1)-(4). Se obtiene así un problema de programación lineal al que se denomina relajación LP inicial.

Paso 6: Cota inferior inicial. Resolver el LP actual, con solución óptima x y valor en la función objetivo \underline{z} . Si x es factible para el problema de programación entera IP, **FIN** (x es la solución óptima del problema). Si $\bar{z} - \underline{z} < 1$, **FIN** (la solución encontrada por el algoritmo heurístico es óptima).

En otro caso ir al paso 7.

Paso 7: Selección del problema cuasi-global. Seleccionar las variables que intervienen en el problema cuasi-global. Se considera el siguiente conjunto de variables:

- las variables con valor 1 en la solución proporcionada por varias ejecuciones del algoritmo heurístico.
- las variables básicas en la solución óptima del problema de asignación.
- las variables no básicas en la solución óptima del problema de asignación con coste reducido igual a cero.
- las variables no básicas en la solución óptima del problema de asignación con valor 1.

Este conjunto de variables es la mejor estimación hasta el momento del conjunto de variables con valor 1 en la solución óptima. Estas

variables, junto con las restricciones (1)-(4) y la función objetivo del modelo, definen la primera relajación lineal del llamado problema cuasi-global. (LP cuasi-global). La solución inicial para la resolución de este problema es la mejor solución obtenida hasta el momento.

Si no se ha obtenido ninguna solución factible utilizando el algoritmo aproximado, entonces la primera versión del problema cuasi-global es el problema global.

Paso 8: Optimización del problema cuasi-global. Obtener la solución óptima de la relajación lineal del problema cuasi-global.

Paso 9: Obtención de costes reducidos. Para probar que la solución óptima del problema LP cuasi-global es también óptima para el problema LP global es necesario obtener los costes reducidos de las variables que no han sido incluidas en el problema cuasi-global.

Paso 10: Introducción de variables en el problema cuasi-global. Si el coste reducido de alguna de las variables no tiene el signo apropiado, se revisa el problema cuasi-global añadiendo las variables que no lo tengan. Formalmente, se añaden las variables que satisfacen la condición $\bar{c}_{ij} \leq 0$, donde \bar{c}_{ij} es su coste reducido. (Nota: Se supone que las variables no incluidas en el problema cuasi-global tienen valor $x_{ij} = 0$.)

En este caso se revisa la solución actual del LP y se vuelve al paso 8.

Paso 11: Fijación permanente de valores de las variables. Si todos los costes reducidos \bar{c}_{ij} tienen el signo adecuado, entonces el valor óptimo actual del LP, z , es una nueva cota inferior (más fuerte que la mejor obtenida hasta el momento) de la solución óptima del problema global. Además, las variables que satisfagan la

condición

$$|\bar{c}_{ij}| \geq \bar{z} - \underline{z}$$

se fijan permanentemente a su valor actual (que puede ser cero o uno).

Si al llegar aquí todas las variables no básicas tienen un valor fijado, **FIN**. Se ha obtenido la solución óptima del problema global.

Paso 12: Si en el paso anterior se han fijado un número considerable de variables, se ejecutan de nuevo los pasos 2 y 3 (Preproceso). A continuación se trata de reducir el tamaño del problema.

Paso 13: Reducción del problema cuasi-global. (Este paso es correcto debido a que se han fijado algunas variables como permanentes.)

La reducción se obtiene:

- a. Detectando nuevas variables cuyo valor deba fijarse por implicación, y eliminando todas las variables fijadas.
- b. Actualizando el lado derecho de la restricción (1) del modelo.
- c. Actualizando los vectores r y d tal como está indicado en preproceso (sección 2.3.2.).
- d. Eliminando la restricción (2) (resp. (3)) correspondiente al nudo i (resp. al nudo j) si se ha fijado la variable $x_{ij} = 1$.

Si esta reducción de variables afecta al valor de \underline{z} , se calcula de nuevo la solución del LP y se vuelve al paso 8.

Observación 2.5: Si la solución de dos optimizaciones del LP consecutivas es tal que su diferencia es menor que σ , donde σ es un nivel de tolerancia, normalmente $\sigma = 10^{-4}$, entonces no se llevan a cabo

los pasos correspondientes a la fijación de costes reducidos ni el paso 14 de generación de restricciones que se detalla a continuación.

Paso 14: Generación de restricciones. Una vez comprobado que no se pueden lograr más reducciones del problema utilizando los procedimientos anteriores, se ejecuta la fase de identificación de restricciones, añadiendo al problema cuasi-global la SEC y la PFC más violadas (utilizando los algoritmos de separación descritos en las secciones 2.4.2. y 2.4.3.). Cuando se añaden las nuevas restricciones, se elimina de la actual relajación LP cualquier restricción no activa que hubiera sido añadida con anterioridad.

Si no se ha obtenido ninguna restricción violada, **FIN**. El algoritmo proporciona como salida la cota inferior \underline{z} .

Paso 15: Si se ha podido añadir alguna restricción en el paso anterior se actualiza la solución actual del LP. Ir al paso 7.

Otros cortes válidos para el SOP

En lugar de introducir como corte una restricción tipo SEC, o PFC es posible introducir en el problema relajado cualquier restricción válida inducida por ellas. Muchos de los cortes válidos para el polítopo del ATSP, P^n , se pueden escribir de forma que sean también válidos para el SOP(n,P).

Las clases de desigualdades válidas y de desigualdades que definen facetas para el P^n se han estudiado en Grötschel (1977), Padberg y Rao (1982), Padberg y Rinaldi (1990), Grötschel y Holland (1991) y otros muchos. Sin embargo no es objetivo de este trabajo hacer un estudio exhaustivo sobre estas condiciones y su aplicabilidad al SOP, ya que no serán necesarias para la metodología que propondremos en el capítulo 3.

2.5.- OBTENCION DE COTAS INFERIORES. SOP CON RELACIONES DE PRECEDENCIA Y ACOTACIONES

Se plantea el problema de encontrar un camino hamiltoniano \mathcal{H} con peso total mínimo en un grafo de forma que se satisfagan las relaciones de precedencia dadas por el digrafo $P=(N,\Pi)$, cuando existen acotaciones superiores al peso de cada subcamino hamiltoniano, proporcionadas por el vector n -dimensional d .

Para ello se desarrolla un algoritmo de planos de corte basado en el modelo 2.5 de la sección 2.2.3. La metodología básica de este algoritmo sigue el esquema descrito en el algoritmo 2.2. Ahora, sin embargo, las condiciones que fuerzan la precedencia y las acotaciones son introducidas en la relajación lineal del modelo (debido a que el número de ellas es del orden de n^2 como máximo), y sólo se utiliza el algoritmo de separación descrito en la sección 2.4.2 para las SEC.

La experiencia computacional para este modelo indica que las cotas que se obtienen están muy alejadas de la solución óptima del problema combinatorio. Parece necesario, por tanto, utilizar un nuevo enfoque que permita la obtención de buenas cotas inferiores. Este será el problema que abordaremos con la metodología propuesta en el capítulo siguiente.

2.6.- ALGORITMO EXACTO

En cualquiera de los algoritmos de planos de corte descritos anteriormente, el algoritmo termina cuando se encuentra la solución óptima, o bien cuando la única restricción violada es la condición de integralidad (restricción (6) del modelo 2.5).

En problemas de no muy grandes dimensiones es posible llegar a la solución óptima utilizando las técnicas habituales de programación entera. En concreto, si la única restricción violada es la condición de

integralidad, se puede utilizar una técnica branch-and-cut, de forma similar a la descrita en Padberg y Rinaldi (1991) para el problema del viajante.

El objetivo es fijar variables a un valor determinado (0 ó 1). En cada nudo n habrá un subconjunto N_0^n de variables fijadas con valor 0, y un subconjunto N_1^n de variables fijadas con valor 1, de forma que se verifique $N_0 \cup N_1 \subset \bar{A}$.

$$N_0^n = \{ x_{ij} / x_{ij} = 0 \text{ en el nudo } n \}$$

$$N_1^n = \{ x_{ij} / x_{ij} = 1 \text{ en el nudo } n \}$$

En cada nudo se efectúa la fase de preproceso, con lo que probablemente nuevas variables fijaran sus valores a cero ó a uno. A continuación se resuelve para ese nudo el problema de programación lineal asociado, y después se ejecuta el procedimiento de generación de restricciones tratando de identificar condiciones violadas. Estas restricciones se añaden al subproblema lineal asociado.

Se resuelve por tanto en cada nudo un subproblema similar al problema original. Cuando en un nudo la única condición violada es la condición de integralidad se continúa el proceso de ramificación eligiendo una nueva variable que será fijada a cero ó a uno.

Así en cada nudo estarán fijadas no sólo las variables correspondientes a las ramificaciones hasta llegar a él, sino todas las variables que hayan sido fijadas en el fase de preproceso efectuada en el mismo nudo o en cualquiera de sus nudos predecesores en la ramificación.

De la misma forma en cada nudo el subproblema lineal asociado incluirá las restricciones generadas en el mismo nudo y en cualquiera de sus nudos predecesores. Por tanto en nudos provenientes de distintas ramificaciones tendremos distintos problemas a resolver, y la información requerida en cada nudo debe ser almacenada. Utilizando este procedimiento

se resuelven problemas más complejos, pero el número de ramificaciones es menor que si se efectúa el tradicional algoritmo branch-and-bound.

El proceso de ramificación continúa hasta que en una rama se obtenga una solución entera, que se almacenará si es mejor que la mejor solución obtenida hasta el momento, o hasta que la cota proporcionada por un nudo sea peor que la mejor solución obtenida. Cuando no se puede seguir adelante el proceso de ramificación, la mejor solución obtenida hasta el momento será la solución óptima del problema.

Capítulo 3

Metodología basada en el Dual

3.1.- INTRODUCCION

3.1.1.- Motivación

Una vez analizada la metodología basada en el primal, podemos observar dos inconvenientes fundamentales para su aplicación al SOP:

- 1) Dados los tipos de condiciones del SOP que se relajan (restricciones de eliminación de subciclos (SEC's), restricciones que fuerzan la precedencia (PFC's) y restricciones que fuerzan las acotaciones (DFC's)), el número de condiciones es finitamente grande (exponencial en $|V|+|A|$) en los dos primeros casos. Por tanto, no hay ninguna garantía de que en el peor de los casos no fuera necesario generar y, por tanto, añadir al problema LP un número finitamente grande de condiciones de este tipo. Por consiguiente, en ese punto extremo se llegaría a invalidar la razón fundamental por la cual se adopta este tipo de metodología.

- 2) Una de las condiciones más importantes desde el punto de vista algorítmico, como es la condición de integralidad, se relaja en este tipo de metodología. Por tanto, se corre el grave peligro de que la convergencia sea muy pequeña, es decir, que la diferencia en el valor de la función objetivo entre dos problemas sucesivos de PL (en el que en el segundo se ha incluido alguna condición violada por la solución óptima del primero) sea muy pequeña y, por tanto, la cota inferior esté alejada del valor óptimo.

Por tanto, proponemos utilizar una metodología distinta, que denominaremos *Metodología Dual*, para la obtención de cotas inferiores para el SOP. Esta metodología se basa en técnicas de relajación Lagrangiana, adecuadamente adaptadas al problema que nos ocupa.

Una de las ideas más útiles a nivel computacional en las últimas décadas ha sido la observación de que muchos problemas difíciles de programación entera se pueden considerar como problemas fáciles (en términos de complejidad), complicados por un conjunto relativamente pequeño de restricciones adicionales. La dualización de estas restricciones da lugar a un problema Lagrangiano fácil de resolver, cuya solución óptima proporciona una cota inferior (en problemas de minimización) al valor óptimo del problema original. Así, es posible utilizar el problema Lagrangiano en lugar de la relajación lineal para la obtención de cotas. Además, el enfoque Lagrangiano ofrece un número importante de ventajas respecto a la relajación lineal.

Este enfoque fue utilizado por Held y Karp (1970) para, a partir de un problema Lagrangiano basado en árboles generadores de peso mínimo, proporcionar un algoritmo para el Problema del Viajante mucho más eficiente que todos los obtenidos hasta entonces utilizando relajaciones lineales. A partir de ese momento la técnica se ha utilizado en muchos problemas combinatorios "clásicos", proporcionando, en la mayoría de los casos, el mejor algoritmo existente para el problema.

Sin embargo, tradicionalmente se suponía que el conjunto de restricciones adicionales a dualizar era relativamente pequeño en relación con las dimensiones del problema. Con nuestra *Metodología Dual* proponemos una extensión de las técnicas Lagrangianas a problemas donde el número de restricciones complicadas es finitamente grande, como es el problema que nos ocupa.

Las características de esta metodología son las siguientes:

- 1) Al igual que la metodología primal, esta metodología consiste en obtener cotas inferiores resolviendo iterativamente problemas relajados para los que se tiene la garantía de que el valor en la función objetivo (a minimizar) de cada nuevo problema no es superior al valor de la solución óptima, pero tampoco inferior a la cota obtenida anteriormente. En nuestro problema concreto relajaremos, al menos, las así llamadas restricciones complicadas, es decir, las PFC's y DFC's.
- 2) La primera diferencia con la metodología primal es que no se relaja la condición de integralidad de las variables binarias. Por tanto el problema resultante relajado ya no es un problema de Programación Lineal, sino que sigue siendo un Problema Combinatorio.

Esta característica es muy importante, porque en el caso de que ese problema combinatorio pueda resolverse en tiempo polinomial hay cierta esperanza de que la convergencia al óptimo del problema original sea más rápida que en el caso de la relajación lineal.

- 3) La segunda gran diferencia (en nuestro problema concreto) es que, en principio, no se relaja un tipo de condiciones muy difícil de tratar, como son las SEC's (número exponencial de condiciones) y, por tanto, se incrementa la expectativa de una deteriorización fuerte en la función objetivo.

El problema resultante de relajar en el SOP las condiciones PFC's y

DFC's es un problema combinatorio para el que no hay algoritmos conocidos con complejidad polinomial (ver sección 1.6.5).

El problema relajado queda planteado de la forma siguiente:

$$z^* = \text{MIN } c^t x$$

s.a.

$$(1) x(\bar{A}) = n-1$$

$$(2) x(\delta^-(j)) = 1 \quad \forall j \in V \setminus \{0\}$$

$$(3) x(\delta^+(j)) \leq 1 \quad \forall j \in V$$

$$(4) x(W) \leq |W| - 1 \quad \forall W \subset V, 2 \leq |W| \leq n-1$$

$$(5) x_{ij} \in \{0,1\} \quad \forall (i,j) \in \bar{A}$$

Se puede observar que este modelo es el programa 0-1 para obtener el camino Hamiltoniano en el grafo $D=(V,\bar{A})$ de mínimo peso, o *problema del viajante asimétrico*. Este problema es *NP*-duro y, por tanto, la relajación obtenida no es un "buen" problema combinatorio.

La condición (3) exige que no salga más de un arco de cada nudo. Podemos observar que el problema resultante al relajar esta condición es el problema de obtener una arborescencia generadora de mínimo peso, problema que tiene complejidad polinomial. Ver sección 1.6.2.

4) Así, y ésta es una característica muy importante de la metodología que proponemos, no se relajan las condiciones SEC's, pero en su lugar se relaja (en forma dualizada, como ya se verá) el tipo de condiciones (3), que llamaremos *de asignación*. Es preciso notar que una aproximación al problema del viajante no necesitaría de ningún otro tipo de relajación. (Ver Held y Karp (1970)).

Una vez identificadas las condiciones (y cortes reforzados) de asignación, PFC's y DFC's que hayan sido violadas por la solución óptima del problema del árbol generador de mínimo peso, si se incluyesen estas

condiciones en el conjunto de restricciones del problema combinatorio relajado, obviamente, se destruiría la estructura de arborescencia generadora de dicho conjunto de condiciones. Vemos aquí, pues, dos graves inconvenientes:

El primer inconveniente sería del mismo tipo que en el caso de la metodología primal puesto que, actuando de esta forma, se puede llegar a incrementar el problema relajado en un número exponencial de condiciones. El segundo inconveniente es que no existen hoy día algoritmos con complejidad polinomial para resolver el nuevo problema combinatorio.

Por tanto, necesitamos utilizar la información obtenida por la violación de las condiciones de asignación, PFC's y DFC's sin que por ello destruyamos la estructura del problema combinatorio y, por tanto, sin incrementar el número de condiciones.

- 5) Así, proponemos **dualizar** las condiciones y cortes implicados cuya violación hayamos identificado. Para ello hemos de expresar estos cortes de forma que su dualización signifique únicamente una modificación (obviamente, lineal) en los coeficientes de la función objetivo de las mismas variables del problema. Precisaremos, por tanto, una metodología apropiada para estimar los correspondientes multiplicadores de Lagrange. Ver sección 3.3. Este esquema no es totalmente nuevo en el tratamiento de problemas combinatorios (ver por ejemplo Aboundi, Hallefjord y Jornsten (1991) para el problema de asignación con ciertas condiciones adicionales), pero si es novedoso para el SOP.

Un problema combinatorio alternativo al anteriormente propuesto como problema auxiliar a resolver en cada iteración consistiría en relajar las SEC's en lugar de las condiciones de asignación. Tendríamos así el programa 0-1 para resolver el problema de asignación, que tiene también complejidad polinomial.

Identificar condiciones de asignación no satisfechas por una solución

binaria es tan trivial como identificar SEC's para la misma solución binaria. Ahora bien, en nuestro problema sólo tenemos n condiciones de asignación y, por tanto, en el peor de los casos sólo es necesario estimar n multiplicadores de Lagrange. En cambio hay un número exponencial de condiciones SEC's, lo que implica que el número de multiplicadores de Lagrange a estimar y actualizar podría ser excesivamente grande. Por otro lado, conjeturamos que este tipo de relajación sería más débil.

Parece pues, en principio, que hay razones para justificar la utilización del problema de la arborescencia generadora.

En definitiva nuestra metodología dual propone:

- (1) Resolver problemas combinatorios difíciles via la resolución de problemas combinatorios con complejidad polinomial.
- (2) Identificar condiciones violadas, y obtener con complejidad polinomial cortes equivalentes y más fuertes para la solución óptima del problema combinatorio auxiliar.
- (3) La utilización apropiada de la teoría Lagrangiana al dualizar la función objetivo para su convergencia hacia la solución óptima, via la utilización de las condiciones y cortes identificados.

3.1.2.- Desarrollo de la metodología

Una vez motivada la conveniencia de enfrentarse al problema desde un nuevo enfoque, en la sección 3.2 se desarrolla el modelo matemático adecuado para esta metodología, y se describe el tipo de preproceso necesario para ese modelo. En la sección 3.3 se describe el método para la actualización y utilización de los multiplicadores de Lagrange de las condiciones y cortes válidos a dualizar. En el capítulo 4 se presenta nuestra metodología para identificar cortes válidos para las condiciones "naturales" PFC's y DFC's. El capítulo 5 presenta la motivación teórica para reforzar los cortes válidos tal que se obtenga una dualización más fuerte de las condiciones violadas.

3.2.- MODELO MATEMATICO

3.2.1.- Modelización

Entre las modelizaciones 0-1 equivalentes posibles para el SOP, el siguiente modelo es el que más se adapta a nuestros requerimientos:

Modelo 3.1

$$\text{Min } \sum_{(i,j) \in A} c_{ij} x_{ij}$$

s.a.

$$(1) x(\bar{A}) = n-1$$

$$(2) x(\delta^-(j)) = 1 \quad \forall j \in V \setminus \{0\}$$

$$(3) x(W) \leq |W| - 1 \quad \forall W \subset V, |W| \geq 1$$

$$(4a) x(\delta^+(j)) = 1 \quad \forall j \in V_1, V_1 = \{ j \in V / \exists (j,i) \in \Pi, i \neq j \}$$

$$(4b) x(\delta^+(j)) \leq 1 \quad \forall j \in V_2, V_2 = V \setminus V_1$$

(5) $i < j$ en \mathcal{H} si el arco $(i,j) \in \Pi$; es decir, x satisface las relaciones de precedencia dadas por el conjunto de arcos en Π

(6) $y_j \leq d_j \quad \forall j \in V$; es decir, x satisface la acotación al peso acumulado de cada nudo, donde y_j tiene la expresión (1.1).

$$(7) x \in \{0,1\}^{|A|}$$

El problema relajado, problema de la arborescencia generadora de peso mínimo, consiste en minimizar la función objetivo anterior sujeta a las condiciones (1), (2), (3) y (7).

En nuestra metodología dual no es necesario contar con una modelización matemática precisa de las condiciones (5) y (6), ya que, dada una solución al problema relajado, es inmediato comprobar si alguna PFC o DFC es violada, en cuyo caso habría que utilizar esta información para modificar los coeficientes de la función objetivo en el problema. En el capítulo 4 se estudia la forma de obtener un corte válido para el SOP a

partir de una condición (5) ó (6) violada.

3.2.2.- Preproceso

Tal como se ha formulado en la sección 2.3, el objetivo fundamental del preproceso en la metodología primal es hacer más fuerte la formulación del problema, transformando el sistema en otro sistema 0-1 equivalente y más fuerte (i.e., se han eliminado soluciones continuas sin eliminar ninguna solución entera). De esta forma la solución del problema lineal relajado puede proporcionar una cota inferior más alta, ya que se ha eliminado un haz de soluciones factibles lineales.

No todos los procedimientos de preproceso presentados en la sección 2.3 son necesarios en el planteamiento de la metodología dual, ya que el problema en todo momento es un programa 0-1. No obstante, es conveniente efectuar un preproceso para tratar de fijar variables a 0 ó a 1, ya que se disminuyen las dimensiones del problema.

Por tanto, el preproceso descrito en la sección 2.3 tiene la siguiente especialización para la metodología dual:

- 1.- Se utiliza el procedimiento 2.3.1 para fijar variables a un valor determinado.
- 2.- Los procedimientos 2.3.2, 2.3.3 y 2.3.4 no son necesarios.
- 3.- El procedimiento 2.3.5 está implícito en el modelo.
- 4.- Se utiliza el procedimiento 2.3.6 para detectar cuando un problema dado es infactible.

3.3.- RELAJACION LAGRANGIANA

3.3.1.- Resultados fundamentales

Sea un problema combinatorio formulado como el problema de programación entera IP

$$\begin{aligned}
 z_{IP} &= \text{Min } cx \\
 \text{s.a. } & Ax \leq b \\
 & Bx \leq d \\
 & x \geq 0 \text{ y entero}
 \end{aligned}$$

Sea LP la relajación lineal del problema IP, y sea z_{LP} su valor óptimo.

Sea la relajación Lagrangiana LR_u

$$\begin{aligned}
 z_D(u) &= \text{Min } cx + u(Bx-d) \\
 \text{s.a. } & Ax \leq b \\
 & x \geq 0 \text{ y entero}
 \end{aligned}$$

donde u es el vector de variables duales asociadas con el conjunto de restricciones a dualizar. Se puede observar que las restricciones del problema IP se han separado en dos conjuntos de forma que LR_u es fácil de resolver en relación con IP para una realización de u .

Proposición 3.1: El problema LR_u es una relajación del problema IP, con $u \geq 0$.

Demostración:

Veamos que se verifican las condiciones de la definición 1.21:

(i) Obvio.

(ii) Sea x^* óptimo para IP. Entonces x^* es factible para LR_u y

$$\begin{aligned}
 z_D(u) &= \min \{ cx + u(Bx-d) \} \leq cx^* + u(Bx^*-d) \leq cx^* = z_{IP} \\
 \text{dado que } & u \geq 0 \text{ y } Bx^* - d \leq 0
 \end{aligned}$$

En general, no es posible garantizar que se pueda encontrar un vector u tal que $z_D(u) = z_{IP}$ pero, puesto que el problema LR_u proporciona cotas inferiores para la solución óptima de IP, la mejor elección para u será la solución óptima del problema D

$$z_D = \max_u z_D(u)$$

Este problema se denomina el *dual lagrangiano* del problema IP. Ver Geoffrion (1974), Fisher (1981) y Guignard y Kim (1987), entre otros.

Teorema 3.1: Dado el problema IP, su relajación lineal LP y su dual lagrangiano D, se verifica:

(a) $z_{LP} \leq z_D \leq z_{IP}$

(b) Si para un $u \exists x^*$ que satisface las condiciones

(i) x^* es óptimo para (LR_u)

(ii) $Bx \leq d$

(iii) $u(Bx-d) = 0$

entonces x^* es óptimo para IP.

(c) Si x^* satisface las condiciones (i) y (ii) del apartado (b) pero no (iii), entonces x^* es una solución ϵ -óptima de IP, con $\epsilon = |u(Bx-d)|$.

Demostración:

(a) Por la proposición 3.1, $z_D(u) \leq z_{IP} \forall u \Rightarrow z_D = \max_u z_D(u) \leq z_{IP}$.

Por otra parte,

$$\begin{aligned} z_D &= \max_u z_D(u) = \max_u \{ \min_x cx + u(Bx-d) / Ax \leq b, x \geq 0 \text{ y entero} \} \geq \\ &\geq \max_u \{ \min_x cx + u(Bx-d) / Ax \leq b, x \geq 0 \} = \\ &= \max_u \{ \max_v vb - ud / vA \leq c + uB, v \geq 0, u \geq 0 \} = \\ &= \min_x \{ cx / Bx \leq d, Ax \leq b, x \geq 0 \} = z_{LP} \end{aligned}$$

donde las dos igualdades se deben al argumento de dualidad lineal.

(b) Si x^* satisface (i) y (ii) entonces x^* es factible para IP.

Además, $z_{IP} \geq z_D(u)(x^*) = cx^*$

ya que, por la proposición 3.1, $z_{IP} \geq z_D(u)$, y

$z_D(u)(x^*) = cx^* + u(Bx-d) = cx^*$ (dado que x^* satisface (iii)).

Por tanto x^* es óptimo para IP.

(c) Obvio, ya que x^* es factible y satisface la condición de la definición 1.22. ■

Definición 3.1: El problema LR_u tiene la *propiedad de integralidad* si su valor óptimo no varía si se elimina la condición de integralidad en las variables, i.e.,

$$z_D(u) = z_{LD}(u) \quad \forall u \geq 0,$$

donde LD es el problema D, donde se ha eliminado la condición de integralidad.

Teorema 3.2: Si LP es factible y LR_u tiene la propiedad de integralidad, entonces

$$z_{LP} = z_D$$

Demostración:

$$\begin{aligned} z_{LP} &= \max_{u \geq 0} z_{LD}(u) && \text{(por dualidad lineal)} \\ &= \max_{u \geq 0} z_D(u) && \text{(por la propiedad de integralidad)} \\ &= z_D \end{aligned}$$

Por tanto la relajación Lagrangiana proporciona una cota igual o mejor que la relajación lineal. En modelizaciones para las que LR_u tiene la propiedad de integralidad sólo es operativo utilizar la relajación Lagrangiana cuando la obtención (o aproximación) de z_D sea más fácil que la obtención de z_{LP} .

3.3.2.- Estimación de los multiplicadores de Lagrange.

Puesto que la mejor cota posible para el problema IP en la metodología lagrangiana es $z_D = \max_u z_D(u)$, se trata de encontrar la solución óptima del problema dual Lagrangiano. Esto no siempre es posible, pero la mayoría de los métodos de estimación de multiplicadores se basan en encontrar soluciones óptimas o casi óptimas para este problema.

El problema D cuenta con un gran número de propiedades estructurales que facilitan su resolución. Suponiendo que el conjunto $X = \{ x / Ax \leq b, x \geq 0 \text{ y entero } \}$ de soluciones factibles para LR_u es finito, entonces X se puede representar como $X = \{ x^t, t=1, \dots, T \}$, y el problema D se puede expresar como el siguiente problema lineal con muchas restricciones

$$\begin{aligned} z_D &= \max z \\ \text{s.a.} \quad z &\leq cx^t + u(Bx^t - d) \end{aligned}$$

Se observa así que para cada u la función $z_D(u)$ es la envoltura inferior de una familia finita de funciones lineales. $z_D(u)$ es una función continua, cóncava, diferenciable c.s., pero no diferenciable en general en los puntos \bar{u} , tal que el problema $LR_{\bar{u}}$ tiene óptimos múltiples.

Definición 3.2: Un vector y es un *subgradiente* de $z_D(u)$ en \bar{u} si se satisface:

$$z_D(u) \leq z_D(\bar{u}) + y(u - \bar{u}) \quad \forall u \geq 0$$

Observación 3.1: La función $z_D(u)$ es subdiferenciable en todo punto.

Proposición 3.2: Un vector u^* es óptimo para D \Leftrightarrow 0 es un subgradiente de $z_D(u)$ en u^* .

Existen en la literatura gran número de algoritmos para el problema dual Lagrangiano. En general, estos algoritmos siguen alguno de los siguientes enfoques:

- (1) El método del subgradiente.
- (2) Versiones del método simplex utilizando técnicas de generación de columnas.
- (3) Métodos específicos de ajuste de multiplicadores.

Dado que el método (1) es el que mejores resultados está proporcionando, omitimos el desarrollo de los otros dos métodos. (Ver

Fisher (1981) para una extensión de técnicas de estimación de multiplicadores).

Método del subgradiente para resolver el problema D

Dado un valor inicial u^* se genera una secuencia $\{u_k\}$, tal que

$$u_{k+1} = u_k + \alpha_k (Bx_k - d)$$

donde x_k es una solución óptima de LRu_k y α_k es un escalar positivo, denominado *longitud de paso*.

Teorema 3.5: $z_D(u_k) \rightarrow z_D$ si $\alpha_k \rightarrow 0$ y $\sum_{i=0}^k \alpha_i \rightarrow \infty$.

Demostración:

Ver Bertsekas (1982), entre otros.

Generalmente se utiliza la siguiente expresión para α_k

$$\alpha_k = \frac{\lambda_k (z^* - z_D(u_k))}{\| Bx_k - d \|^2}$$

donde λ_k es un escalar tal que $0 < \lambda_k \leq 2$ y z^* es una cota superior para z_D (obtenida, frecuentemente, aplicando una heurística al problema IP), ver Held, Karp y Wolfe (1974) y Held, Wolfe y Crowder (1974).

A menudo la sucesión λ_k se obtiene fijando $\lambda_0 = 2$ y dividiendo por 2 λ_k cuando el valor $z_D(u)$ no ha aumentado en un número fijo de iteraciones. Esta regla funciona bien empíricamente, aunque no tiene porqué verificar la condición suficiente establecida en el teorema 3.5, ya que por otra parte la diferencia $(z^* - z_D(u_k))$ no tiende en general a cero si z^* se obtiene a partir de una heurística para IP, pues $z \geq z_{IP}^* \geq z_D$.

El método normalmente termina cuando se llega a un número determinado de iteraciones.

3.3.3.- Relajación Lagrangiana del SOP

Tal como se indicó en la sección 3.1 las condiciones del SOP a relajar son las condiciones de tipo (4), (5) y (6) del modelo 3.1.

Por tanto, la relajación Lagrangiana obtenida es

$$\min cx + u(\gamma x - 1) + w(\alpha x - \beta)$$

s.a.

$$(1) x(\bar{A}) = n-1$$

$$(2) x(\delta^-(j)) = 1 \quad \forall j \in V \setminus \{0\}$$

$$(3) x(W) \leq |W|-1 \quad \forall W \subset V, |W| \geq 1$$

$$(7) x \in (0,1)^{|A|}$$

donde u es el vector de variables duales para las condiciones de asignación $x(\delta^+(j)) \leq 1$, representadas en la forma $\gamma x \leq 1$, y w es el vector de variables duales para los cortes válidos de la forma $\alpha x \leq \beta$ correspondientes a las condiciones (5) y (6) del modelo, donde las matrices γ y α pueden contener elementos superiores a uno debido al reforzamiento válido de las condiciones respectivas. (Ver capítulos 4 y 5).

Puesto que las restricciones de tipo (4a) son restricciones de igualdad, el multiplicador correspondiente no está restringido en signo. El resto de los multiplicadores han de ser mayores o iguales que cero. (Se supone que al multiplicador de una condición violada todavía no detectada se le asigna el valor estimado nulo).

Cada iteración Lagrangiana se distingue por la actualización de los vectores de multiplicadores u y w . Así, en cada iteración se resuelve un problema combinatorio de la arborescencia generadora de peso mínimo, y se pasa a la siguiente iteración identificando un corte válido correspondiente a una restricción violada y realizando una modificación en

los coeficientes de las variables en la función objetivo.

Se utilizarán a continuación los conceptos y notación de las secciones anteriores adaptándolas y desarrollándolas para el problema que nos ocupa.

Sea el siguiente problema SOP, donde en el espacio de soluciones X se recogen las condiciones que definen una arborescencia:

$$(P) \quad \min_x \{ cx \mid Ax \leq b, x \in X \}$$

donde b y c son vectores y A es la matriz de dimensiones apropiadas tal que el sistema $Ax \leq b$ recoge el conjunto de condiciones de asignación, condiciones que fuerzan las precedencias y condiciones que fuerzan las acotaciones. La relajación Lagrangiana de (P) relativa a $Ax \leq b$ con el vector de multiplicadores u (dimensionado como b y con las restricciones en signo indicadas anteriormente) es

$$(LR_u) \quad \min_x \{ cx + u(Ax-b) \mid x \in X \}$$

Claramente, para $u = 0$, las condiciones llamadas "complicadas" desaparecen de la función objetivo. En general, (LR_u) depende de u , tal que el valor óptimo de (LR_u) , sea $z_D(u)$, es un límite inferior del valor óptimo de (P), sea z_P . El dual Lagrangiano (D) determina el mejor límite inferior de z_P .

$$(D) \quad z_D = \max_u z_D(u)$$

Puesto que $z_D \leq z_P$, el valor óptimo de (D) es un buen límite inferior del valor óptimo del problema original. En muchos casos, es fácil crear un algoritmo heurístico para, manipulando la solución de (D), obtener una solución factible (y probablemente muy buena) para el problema original (P).

Para resolver (D) se podría utilizar en principio el método del subgradiente tal como se ha indicado en la sección anterior. Esta

metodología resuelve el problema (D) calculando, iterativamente, el subgradiente $g = Ax_k - b$, donde x_k es la solución óptima al problema de arborescencia (LR_{u_k}). Para la iteración k se actualizan los multiplicadores de Lagrange u , según la fórmula

$$u_{k+1} = \max \{ 0, u_k + \alpha_k g \}$$

donde

$$\alpha_k = \lambda_k \frac{\bar{z}_k - z_D(u_k)}{\|g_k\|} \quad (3.1)$$

De nuevo se resuelve el problema para la iteración $k+1$, etc.

En la expresión (3.1) \bar{z}_k es un estimador de z_D , tal que cuanto menor sea la diferencia $|\bar{z}_k - z_D|$, mayor precisión tendrá la actualización de u (demostrado por Held y Karp, (1970)) y, por tanto, menor número de iteraciones serán necesarias para obtener z_D . Se suele tomar un valor suficientemente grande (lo que perjudica la convergencia del algoritmo), salvo que haya una solución heurística, en cuyo caso se toma el valor correspondiente en la función objetivo. Salvo que este valor \bar{z}_k sea tal que $\bar{z}_k - z_P$ (siempre ≥ 0) es muy pequeño, y además $z_P - z_D$ (siempre ≥ 0) sea también muy pequeño, incluso tomar para \bar{z}_k el valor de la solución heurística no es un buen método.

Si se pudiera obtener \bar{z}_k como una mejor aproximación a z_D , el algoritmo tendría mayor eficiencia e, incluso, se podría dejar constante el escalar λ_k (e.g., $\lambda_k = 1$) si se consigue que la diferencia $(\bar{z}_k - z_D(u_k)) \rightarrow 0$ cuando $k \rightarrow \infty$. A continuación se presenta una metodología que, en el marco de la generación de cortes, puede ayudar a obtener \bar{z}_k para cada iteración (en la esperanza de ir reduciendo la distancia $|\bar{z}_k - z_D|$).

Método de generación de filas y columnas para la estimación de multiplicadores.

Sea T el conjunto de puntos extremos del conjunto X , que engloba las restricciones del problema con condiciones fáciles (en nuestro caso, el

problema de arborescencia). Se supone que $T \neq \emptyset$ (de lo contrario, (P) sería infactible). Se puede reescribir el problema (LR_u) , tal que

$$z_D(u) = \min_{x \in X} \{ cx + u (Ax-b) \} = \min_{t \in T} \{ cx^t + u (Ax^t-b) \}$$

donde el vector x^t representa las coordenadas del punto extremo t para $t \in T$.

Por tanto, la resolución del problema (D) es equivalente a la resolución de problema

$$\max_u z_D(u) = \max_u \{ \min_{t \in T} \{ cx^t + u (Ax^t-b) \} \}$$

con $u_i \geq 0 \quad \forall i$ excepto aquellos correspondientes a restricciones de asignación con igualdad, cuyo multiplicador asociado no está restringido en signo; problema que se puede reescribir de nuevo como un problema de programación lineal, sea (DM) (Dual Maestro) según la descomposición de Benders (1962)

$$(DM) \max_u \{ \gamma / \gamma \leq cx^t + u (Ax^t-b) \quad \forall t \in T \}.$$

Definición 3.3: Una *desigualdad tipo Benders* es una desigualdad del tipo $\gamma \leq cx^t + u (Ax^t-b)$ para algún $t \in T$.

Cada punto extremo $t \in T$ proporciona una desigualdad tipo Benders, i.e., una de las restricciones del problema. Por tanto el problema (DM) es un problema con tantas filas como puntos extremos haya en X , y tantas columnas como restricciones "complicadas" haya en el problema (P).

Si el conjunto T fuera conocido explícitamente bastaría con resolver el problema lineal (DM) para obtener el vector de multiplicadores u óptimo. Pero $|T|$ puede ser un número enormemente grande que haría impráctico trabajar con (DM), dado el número de condiciones que sería preciso considerar. En cambio es muy posible que sólo un número pequeño de

condiciones en (DM) sean necesarias para resolver el problema, i.e., pocos puntos extremos en T proporcionan condiciones activas en (DM). Supongamos por un momento que se conoce el conjunto de puntos extremos, sea S para $S \subseteq T$, cuyas correspondientes condiciones en (DM) son suficientes para resolver el problema. Por tanto, se puede reescribir (DM), tal que

$$(DM) \max_u \{ \gamma / \gamma \leq cx^t + u(Ax^t - b) \forall t \in S \}.$$

La metodología de Benders (ver apéndice 1) permite en general obtener el conjunto S de una forma eficiente. Sin embargo en nuestro caso, no es sólo el conjunto T quien no es conocido explícitamente. Tampoco están explícitamente modelizadas todas las restricciones "complicadas" del problema y, por tanto, se desconocen los elementos del vector $Ax^t - b$ correspondientes a condiciones no identificadas.

Observación 3.2: El vector de multiplicadores u para el problema (P) forma, junto con la variable γ , el vector de variables (u, γ) para el problema (DM).

Definición 3.4: El multiplicador u_i está *activado* si su valor está prefijado a cero y está *deactivado* si es una variable libre del problema (DM).

Llamaremos (DMR) (Problema Dual Maestro Relajado) a una relajación del problema (DM) donde no aparecen las condiciones tipo Benders relativas a puntos extremos todavía no identificados y donde algunas componentes del vector u (aquellos multiplicadores correspondientes a restricciones complicadas todavía no identificadas) están activadas.

Podemos utilizar el método que se verá en el capítulo siguiente para generar cortes para, a su vez, construir progresivamente un conjunto cada vez más restrictivo de condiciones en (DMR). Es preciso notar que en la iteración k se resuelve el problema de la arborescencia (LR_{u_k}) y, por tanto, se obtiene una nueva solución, sea x^k . La fase de identificación de cortes violados (y su correspondiente reforzamiento) proporciona un

conjunto de restricciones, sean $(Ax^k \leq b)_i$ para $i \in I^k$, (donde I^k es el conjunto de cortes) que son violadas en el problema original por el punto x^k . La aplicación del punto x^k en la función objetivo de (LR_{u_k}) proporciona la condición tipo Benders a incluir en el problema (DMR_{k+1}) . Las condiciones en I^k implican que los correspondientes multiplicadores de Lagrange (y, por tanto, variables en (DMR)) deben desactivarse y, así, utilizarse en la optimización de (DMR_{k+1}) . Esto implica la adición de las correspondientes $|I^k|$ columnas en la matriz de condiciones del problema (DMR_{k+1}) . Ver figura 3.1.

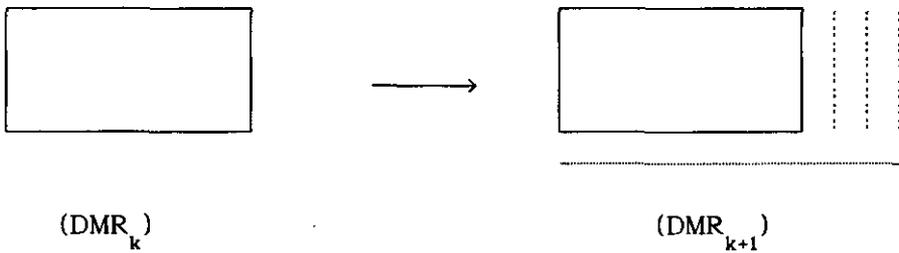


figura 3.1

A continuación se resolvería este nuevo problema para obtener el nuevo vector de multiplicadores de Lagrange, sea u^{k+1} , que se utilizaría en el problema de arborescencia $(LR_{u_{k+1}})$ correspondiente a la siguiente iteración.

Supóngase que antes de añadir la nueva condición basada en el punto x^k al problema dual maestro relajado, el conjunto de condiciones en (DMR_k) es S^k , y el correspondiente valor en la función objetivo es $v(DMR_k)$. La nueva condición derivada del punto x^k será un corte en el problema dual relajado, esto es,

$$v(DMR_k) \geq cx^k + \sum_{\substack{i \in U \cup I^j \\ j=1,2,\dots,k-1}} u_i (Ax^k - b)_i \quad (3.2)$$

Observación 3.3: Incluso si no se identifica ningún nuevo corte válido para el problema original que esté violado por el punto x^k , el corte inducido por la condición (3.2) debe introducirse en (DMR_{k+1}) .

Observación 3.4: En caso de que x^k no viole ninguna condición del problema original y la situación anterior se cumpla con igualdad estricta, x^k es la solución óptima del problema general (P).

Es preciso notar que en la metodología de Benders, dado que todas las restricciones no incluidas en el conjunto que definen los puntos extremos en X están dualizadas con valores nulos en sus respectivos multiplicadores, la sucesión de valores de $v(DMR_k)$ es monótona decreciente. Por tanto, se puede utilizar el valor $v(DMR_k)$ como el valor \bar{z}_k en la obtención de la amplitud de paso α_k y, finalmente, converge a $v(DM)$. En la práctica, se podría terminar la ejecución del algoritmo cuando la diferencia entre \bar{z}_k y $v(LR_u)$ (límites superior e inferior de $v(DM)$, respectivamente) sea menor que una tolerancia dada.

Ahora bien, dado que la dualización (LR_u) no incluye en el SOP todas las condiciones de asignación, precedencias y acotaciones (i.e., los multiplicadores duales respectivos están activados), resulta que cuando se añaden nuevos cortes, el problema (DMR) se convierte en un problema de mayores dimensiones (o lo que es lo mismo, se permite desactivar los correspondientes multiplicadores de Lagrange). En esta situación podemos visualizar el problema como una secuencia de relajaciones del problema dual maestro (DMR_1) , (DMR_2) , ..., (DMR_k) , tal que la serie $v(DMR_1)$, $v(DMR_2)$, ..., $v(DMR_k)$ converge al óptimo $v(DM)$, aunque no necesariamente de forma monótona.

En principio, se podría comenzar utilizando la metodología del subgradiente para obtener los multiplicadores duales u . A medida que el óptimo del problema de la arborescencia no se deteriore adecuadamente, se podría utilizar el problema relajado (DMR) para incrementar la convergencia. El problema (DMR) podría incluir las condiciones correspondientes a las arborescencias x^1 , x^2 , ... obtenidas en las iteraciones anteriores.

Algoritmo 3.1

La metodología del subgradiente tiene normalmente un ritmo de convergencia muy rápido en las primeras iteraciones, pero a medida que se incrementa el número de iteraciones, los multiplicadores de Lagrange tienen escasa variación y, por tanto, el ritmo de deteriorización del óptimo de la relajación Lagrangeana se va paulatinamente reduciendo.

La utilización de cortes violados en el problema (DMR) puede reemplazar a la metodología del subgradiente e incrementar el ritmo de convergencia a $v(DM)$.

Paso 0: Resolver LR_0 con $u = 0$, primera relajación Lagrangiana del problema.

Sea x^0 su solución óptima y $v(LR_0)$ su valor óptimo. Hacer $\underline{z} = v(LR_0)$.

Hacer $S = \{ 0 \}$, $\epsilon = 10^{-4}$, $k = 0$, $\sigma = 0$, $\bar{\sigma} = K$ (cte.), $\bar{z} = \infty$.

Estimar el vector u según el método del subgradiente.

Paso 1: $k = k+1$

Resolver (LR_k) y hacer $\underline{z} = \max \{ \underline{z}, v(LR_k) \}$

Si $\frac{|v(LR_k) - v(LR_{k-1})|}{1 + |v(LR_{k-1})|} < \epsilon$ ó x^k no proporciona nuevos cortes,

hacer $\sigma = \sigma+1$. En caso contrario hacer $S = S \cup \{ x^k \}$ y añadir los nuevos cortes.

Paso 2: Heurística: Si se encuentra una nueva solución, actualizar

$\bar{z} = \min \{ \bar{z}, \text{nueva solución} \}$

Paso 3: Resolver (DMR_k) para el conjunto S actual.

Hacer $\bar{z}_B = v(DMR_k)$

Paso 4: Actualizar u :

Si $\sigma > \bar{\sigma}$ entonces obtener u a partir de (DMR_k) . En caso contrario obtener u utilizando la metodología del subgradiente con \bar{z}_B como cota superior.

Paso 5: Criterio de parada:

$$\text{Sea INT} = \frac{\bar{z} - z}{z}$$

Si $\text{INT} < \epsilon$, o el número de iteraciones es alto, o x^k factible
FIN. En caso contrario ir al paso 1.

3.4.- ALGORITMO GENERAL DE PLANOS DE CORTE UTILIZANDO METODOLOGÍA DUAL. Líneas generales:

Entrada: Matriz de coeficientes c_{ij} y digrafo acíclico de precedencias $P=(N,\Pi)$.

Paso 1: Calcular la envoltura transitiva cerrada del digrafo acíclico P para obtener el digrafo de precedencias inicial.

Paso 2: Preproceso. (Sección 3.2.2)

Paso 3: Solución inicial factible. Obtener una cota superior inicial de la solución óptima. La aplicación de un algoritmo heurístico proporciona la primera cota superior, con valor \bar{z} en la función objetivo.

Paso 4: Obtención de la primera relajación del problema. Considerar el siguiente problema relajado:

- Incluir las restricciones (1), (2), (3) y (7) del modelo 3.1 (que junto con la función objetivo representan el problema de encontrar la arborescencia de peso mínimo).
- Ignorar las PFC y las DFC (restricciones (5) y (6) del modelo, respectivamente).
- Relajar las restricciones tipo (4) utilizando la metodología Lagrangiana con multiplicadores u .

Se plantea así el siguiente problema relajado, al que denominaremos *Relajación Lagrangiana Débil*:

$$\text{Min } L_k(u) = \sum_{(i,j) \in A} (c_{ij} + u_i) x_{ij} - \sum_{v \in V} u_v \quad \text{con } u_v \geq 0 \quad \forall v \in V_2$$

s.a.

$$(1), (2), (3), (7)$$

Para la primera iteración, en principio, asignar $u_v = 0 \quad \forall v \in V$, $k = 1$.

Paso 5: Resolver la relajación Lagrangiana L_k para la etapa k -ésima.
(Problema de la arborescencia de mínimo peso, con solución T).

Paso 6: Identificar condiciones de asignación violadas por T .

Paso 7: Identificar, por inspección, PFC y DFC violadas por la arborescencia T .

Paso 8: Reforzar las restricciones PFC y DFC identificadas por adición de variables (Secciones 4.2 y 4.3 respectivamente)

Paso 9: Reforzar las restricciones anteriores por incremento de coeficientes utilizando información proporcionada por otras restricciones violadas ya identificadas. (Capítulo 5)

Paso 10: Reforzar restricciones violadas ya identificadas utilizando la información proporcionada por las nuevas restricciones identificadas. (Capítulo 5).

En general, después de los pasos 7, 8, 9 y 10 se obtienen cortes violados del tipo

$$\sum_{(i,j) \in F_1 \subset A} \alpha_{ij} x_{ij} \leq k_1 < |F_1|$$

a los que se asocian multiplicadores w_1 , obteniendo así la *Relajación Lagrangiana fuerte*

$$\text{Min } L_k(u,w) = cx + u(x-1) + w(\alpha x - k) \quad \text{con } u_v \geq 0 \quad \forall v \in V_2, \quad w_1 \geq 0 \quad \forall 1$$

s.a.

$$(1),(2),(3),(7)$$

Para la primera iteración después de introducido el corte l-ésimo, en principio, asignar $w_1 = 0$.

Paso 11: Actualización de los multiplicadores de Lagrange. Actualizar los multiplicadores de las restricciones violadas identificadas, según el algoritmo 3.1.

Si no se ha identificado ninguna restricción violada **FIN**.

Paso 12: $k = k+1$. Ir al paso 5.

Capítulo 4

Cortes válidos para el SOP

4.1.- INTRODUCCION.

Sea el modelo (3.1) para el SOP. La solución óptima al problema de la arborescencia generadora de mínimo peso dado por las restricciones (1), (2), (3) y (7) del modelo proporcionará una cota inferior para el SOP. La dualización (i.e., penalización en la función objetivo) de las restricciones de tipo (4) puede incrementar dicha cota.

Si la solución obtenida satisface todas las restricciones del problema, entonces es la solución óptima. En caso contrario, se trata de encontrar aquellas restricciones que sean "fuertemente" violadas por la solución actual y, a partir de ellas, obtener cortes válidos para el SOP. Estos cortes se introducirán, dualizados, en la función objetivo. La dualización de los mismos requiere una estimación de sus correspondientes multiplicadores de Lagrange. Ver sección 3.3.

Las restricciones SOP no consideradas en la arborescencia generadora (aparte de las restricciones de tipo (4) que se dualizan de entrada) son

las restricciones de tipos (5) y (6). Las primeras evitan que forme parte de la solución un subcamino de la forma $(0, \dots, j)$ si $i \notin (0, \dots, j)$ para $(i, j) \in \Pi$, y las segundas obligan a que el subcamino $(0, \dots, j) \forall j \in V$ sea tal que se satisfaga

$$\sum_{(i,k) \in (0, \dots, j)} T_{ik} \leq d_j, \quad (4.1)$$

es decir, eliminan la posibilidad de que forme parte de la solución un subcamino $(0, \dots, j)$ tal que $y_j > d_j$, para y_j según (1.1).

Las restricciones que fuerzan la precedencia para cada arco $(i, j) \in \Pi$ se han modelizado en la literatura con la siguiente expresión,

$$\begin{aligned} x(j:W) + x(W) + x(W:i) &\leq |W|, \quad \forall (i, j) \in \Pi, \\ \forall W \subseteq V \setminus \{0, i, j\}, \quad |W| &\geq 1 \end{aligned} \quad (4.2a)$$

o, equivalentemente, con la expresión

$$\begin{aligned} x(0:W) + x(W) + x(W:j) &\leq |W|, \quad \forall (i, j) \in \Pi, \\ \forall W \subseteq V \setminus \{0, i, j\}, \quad |W| &\geq 1 \end{aligned} \quad (4.2b)$$

El corte (4.2a) fue descubierto independientemente por Ascheuer, Escudero, Grötschel y Stoer (1993) y Balas, Fischetti y Pulleyblank (1994).

Una violación de las restricciones de tipo (6) corresponde a un vértice j cuya acotación d_j es menor que el peso total acumulado del subcamino $(0, \dots, j)$. Por tanto, ese subcamino debe prohibirse. Para ello proponemos imponer la condición

$$x(0, \dots, j) \leq |W_j| - 1 \quad (4.3)$$

con $W_j = A(0, \dots, j)$.

En las dos secciones siguientes se obtienen modelizaciones para las relaciones de precedencia y las restricciones que fuerzan las acotaciones

que son más fuertes que las expresiones (4.2) y (4.3), respectivamente, y válidas para el problema SOP original. Utilizamos la definición 2.4 para el concepto de "fortaleza" de un sistema.

4.2.- CORTES VALIDOS INDUCIDOS POR LAS RELACIONES DE PRECEDENCIA.

4.2.1. Obtención de cortes válidos.

Teorema 4.1: Sean S_1 y S_2 subconjuntos de $V \setminus W$, tal que $S_1 \times S_2 \subseteq \Pi$.

Entonces

$$x(S_2:W) + x(W) + x(W:S_1) \leq |W| \quad (4.4)$$

es una desigualdad válida para el SOP, $\forall W \subset V \setminus \{0\}$, $|W| \geq 1$.

Demostración:

Sea \mathcal{H} un camino hamiltoniano que satisface las relaciones de precedencia. Para una solución dada, x , sean

$$\begin{aligned} \hat{S}_1 &= \{ i \in S_1 / x(W:i) = 1 \} \\ \hat{S}_2 &= \{ j \in S_2 / x(j:W) = 1 \} \end{aligned} \quad (4.5)$$

* Para cada $j \in \hat{S}_2$ se considera el subcamino de \mathcal{H} que empieza en el nudo j , recorre un subconjunto de nudos de W y termina en el primer nudo que no pertenece a W , sea l . Entonces $l \notin S_1$, ya que $S_1 \times S_2 \subseteq \Pi$ y, por tanto, l no puede ser sucesor de j .

* De la misma forma, dado $i \in \hat{S}_1$, si se considera el subcamino de \mathcal{H} que empieza en un vértice $l \in W$, a continuación recorre un subconjunto de nudos de W y termina en i , resulta que $l \notin S_2$, ya que $S_1 \times S_2 \subseteq \Pi$ y, por tanto, l no puede ser predecesor de i .

* Por tanto, el subgrafo de \mathcal{H} generado por los nudos de W tiene como mínimo $|\hat{S}_1| + |\hat{S}_2|$ componentes. Como cada componente es un camino, entonces el subgrafo es un bosque, y así

$$x(W) \leq |W| - (|\hat{S}_1| + |\hat{S}_2|). \quad (4.6)$$

Como $|\hat{S}_1| = x(W; S_1)$ y $|\hat{S}_2| = x(S_2; W)$ entonces se verifica (4.4) c.q.d. ■

Corolario 4.1a: Si además de las hipótesis del teorema 4.1, existe $k \in W$ tal que $\{k\} \times S_2 \subset \Pi$ y $S_1 \times \{k\} \subset \Pi$, entonces la desigualdad (4.4) se puede reforzar a:

$$x(S_2; W) + x(W) + x(W; S_1) \leq |W| - 1 \quad (4.7)$$

Demostración:

Sean las hipótesis del teorema 4.1 pero, ahora, el subgrafo de \mathcal{H} generado por los nudos de W tiene como mínimo $|\hat{S}_1| + |\hat{S}_2| + 1$ componentes, ya que el nudo k no puede pertenecer a ningún subcamino que empiece en un nudo $j \in \hat{S}_2$ (pues $k < j$) ni a ningún subcamino que termine en un nudo $i \in \hat{S}_1$ (pues $i < k$). ■

Teorema 4.2: Sea $S_1 = \{ j \in V \setminus W \text{ tal que } \exists (i, j) \in \Pi; i \neq j, i \neq 0, i \notin W \}$ y sea S_2 un subconjunto de $V \setminus W$ tal que $S_1 \times S_2 \subseteq \Pi$. Entonces

$$x(0; W) + x(S_2; W) + x(W) + x(W; S_1) \leq |W| \quad (4.8)$$

es una desigualdad válida para el SOP, $\forall W \subset V \setminus \{0\}$, $|W| \geq 1$.

Observación 4.1: El conjunto S_2 puede ser vacío. El conjunto S_1 debe contener algún elemento, pero no necesariamente todos los que satisfagan las condiciones dadas en su definición.

Demostración:

* Si $x(0; W) = 0$ se cumplen las hipótesis del teorema 4.1.

* Si $x(0; W) = 1$ el subgrafo tiene $|\hat{S}_1| + |\hat{S}_2| + 1$ componentes como mínimo, ya que el primer nudo no perteneciente a W del camino que empieza en el nudo 0 y atraviesa W , sea l , no puede pertenecer a S_1 y, por tanto, esta

componente del subgrafo generado por W no está contabilizada. Es preciso notar que el nudo l no puede pertenecer a S_1 ya que, por definición, no puede tener ningún predecesor i tal que $i \in W$, $i \neq 0$, $i \neq l$. ■

Corolario 4.2a: Si además de la hipótesis del teorema 4.2 existe $k \in W$ tal que $\{k\} \times S_2 \subset \Pi$ y $S_1 \times \{k\} \subset \Pi$, entonces la desigualdad (4.8) se puede reforzar como sigue,

$$x(0:W) + x(S_2:W) + x(W) + x(W:S_1) \leq |W| - 1 \quad (4.9)$$

Demostración:

El nudo k no puede pertenecer al subcamino, si existe, que empieza en el nudo 0 , atraviesa W , y termina en el primer nudo l no perteneciente a W , a no ser que $S_1 = \emptyset$. Además k no puede pertenecer a ningún subcamino que empiece en un nudo $j \in \hat{S}_2$ ($k < j$) ni a ningún subcamino que termine en un nudo $i \in \hat{S}_1$ ($i < k$). ■

Teorema 4.3: Sean S_1 y S_2 dos subconjuntos de $V \setminus W$, y sean los nudos i y j tales que $i \in S_1$, $j \in S_2$, $S_1 \times S_2 \subset \Pi$, $(S_1 - \{i\}) \times \{i\} \subset \Pi$ y $\{j\} \times (S_2 - \{j\}) \subset \Pi$. Sea k un nudo de $V \setminus (W \cup S_1 \cup S_2)$ tal que $(i,k) \in \Pi$ y $(k,j) \in \Pi$. Entonces

$$x(S_2:W) + x(i:W) + x(W) + x(W:j) + x(W:S_1) \leq |W| \quad (4.10)$$

es una desigualdad válida para el SOP, $\forall W \subset V \setminus \{0\}$, $|W| \geq 1$.

Observación 4.2: En el caso en que los conjuntos S_1 y S_2 del teorema 4.1 verifiquen las hipótesis del teorema 4.3, la desigualdad (4.10) es un reforzamiento del corte (4.4).

Demostración:

* Dado el teorema 4.1, y definidos los conjuntos \hat{S}_1 y \hat{S}_2 según (4.5), el subgrafo generado por W tiene al menos $|\hat{S}_1| + |\hat{S}_2|$ componentes, dado que $S_1 \times S_2 \in \Pi$.

* Si $x(i:W) = 1$, el subcamino que empieza en i , atraviesa el conjunto W y termina en el primer nudo l no perteneciente a W , proporciona una componente del subgrafo generado por W que no ha sido contabilizada, ya que $l \notin S_1$, pues $(S_1 - \{i\}) \cup \{l\} \in \Pi$. (Análogo para $x(W:j) = 1$).

Además $l \neq j$, ya que $\exists k$ perteneciente a $V \setminus (W \cup S_1 \cup S_2)$ tal que $(i,k) \in \Pi$ y $(k,j) \in \Pi$.

* Y puesto que $|\hat{S}_1| + |\hat{S}_2| = x(S_2:W) + x(W:S_1)$ (4.11)

entonces $x(W) \leq |W| - (|\hat{S}_1| + |\hat{S}_2| + x(i:W) + x(W:j))$ c.q.d. ■

El siguiente teorema define otro tipo de desigualdades válidas para el SOP :

Teorema 4.4: Sea $(i_1, i_2, i_3, \dots, i_k)$ un camino en (V, Π) . Se define

$$S = \{ i_1, i_2, i_3, \dots, i_k \}$$

$$S_1 = \{ i_j \in S \mid j \text{ impar} \} \quad (4.12)$$

$$S_2 = \{ i_j \in S \mid j \text{ par} \} \quad (4.13)$$

$$W \subseteq V \setminus S.$$

entonces

$$x(S_1:W) + x(W) + x(W:S_1) \leq |W| \quad (4.14)$$

$$y \quad x(S_2:W) + x(W) + x(W:S_2) \leq |W| \quad (4.15)$$

son desigualdades válidas para el SOP.

Demostración:

* Sean los siguientes conjuntos:

$$\hat{S}_{11} = \{ i_j \in S_1 \mid x(W:i_j) = 1 \} \quad (4.16)$$

$$y \quad \hat{S}_{12} = \{ i_j \in S_1 \mid x(i_j:W) = 1 \}. \quad (4.17)$$

* Para todo $i_j \in \hat{S}_{11}$, el subcamino de \mathcal{H} que empieza en un nudo $l \notin W$, a continuación recorre un subconjunto de nudos de W y termina en i_j , es tal que $l \notin S_1$, ya que o bien $j = 1$, o bien $\exists i_{j-1} \in S_2$ tal que $i_{j-2} < i_{j-1} < i_j$, y, por tanto, $l = i_{j-1} \in S_2$ ó $l \notin S$.

* Para todo $i_j \in \hat{S}_{12}$, el subcamino en \mathcal{H} que empieza en el nudo i_j , recorre un subconjunto de nudos de W y termina en el primer nudo que no pertenece a W , sea l , es tal que $l \notin S_2$, ya que, o bien $j = k$, o bien $\exists i_{j+1} \in S_1$ tal que $i_j < i_{j+1} < i_{j+2}$ y, por tanto, $l = i_{j+1} \in S_1$ ó $l \notin S$.

* Por tanto, el subgrafo generado por W tiene $|\hat{S}_{11}| + |\hat{S}_{12}|$ componentes como mínimo. Como $|\hat{S}_{11}| = x(W:S_1)$ y $|\hat{S}_{12}| = x(S_1:W)$, se verifica la expresión (4.14) c.q.d.

* La expresión (4.15) se demuestra utilizando un razonamiento análogo. ■

4.2.2.- Identificación de restricciones violadas. Algoritmos de separación.

Sean $\Omega_1, \Omega_2, \Omega_3, \Omega_4$ los conjuntos de índices de las familias de cortes especificados por los teoremas 4.1, 4.2, 4.3 y 4.4, respectivamente. (Cada conjunto contiene, en principio, un número exponencial de índices).

Sea T la arborescencia cuyo nudo raíz es el nudo 0 que optimiza la relajación lagrangiana actual del problema combinatorio auxiliar. Se trata de identificar alguna relación de precedencia que sea violada por esta solución y, a partir de ella, generar un corte con índice en algún Ω_i , para $i=1,2,3,4$.

La figura 4.1 ilustra las dos posibles situaciones en las que una relación de precedencia $(i,j) \in \Pi$ puede ser violada por la arborescencia T .

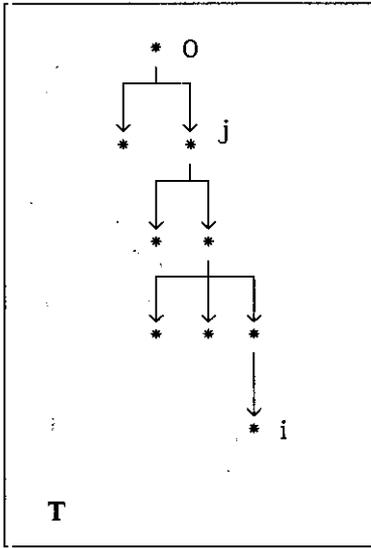


Figura 4.1a

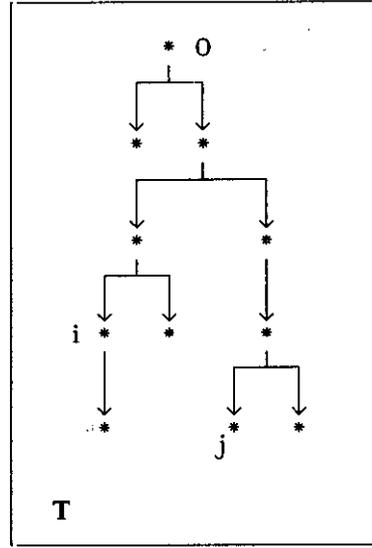


Figura 4.1b

Figura 4.1

Caso 1: Dado un par de nudos (i,j) tales que $i \in P(j)$ y T contiene un subcamino de la forma (j, \dots, i) , (ver figura 4.1a), se puede identificar una desigualdad violada con índice en Ω_1 o en Ω_3 , más fuerte que la desigualdad (4.2a), definiendo los siguientes conjuntos:

* $W = \{ l \in V / l \in (j, \dots, i), l \neq i, l \neq j \}$ en T .

* Para obtener una desigualdad con índice en Ω_1 (i.e., el corte (4.4) dado por el teorema 4.1), definimos

$$S_1 = P(j) \cap (V \setminus W) \quad (i \in S_1)$$

$$S_2 \equiv (\text{mayor subconjunto de } V \setminus W / j \in S_2 \text{ y } S_1 \times S_2 \subseteq \Pi).$$

Ejemplo 4.1: Ver figura 4.2

A continuación se comprueba la condición impuesta en el corolario 4.1a para introducir, si es posible, el corte más fuerte (4.7).

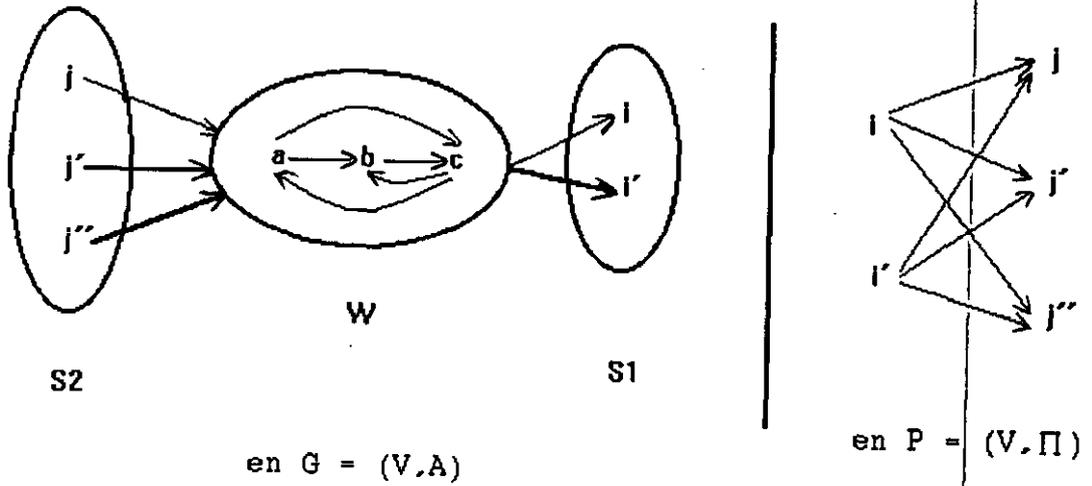


Figura 4.2

* Para obtener una desigualdad con índice en Ω_3 (i.e., el corte (4.10) dado por el teorema 4.3), definimos

$$S_1 \equiv (\text{mayor subconjunto de } P(j) \cap (V \setminus W) / i \in S_1, (S_1 - i) \times i \subset \Pi)$$

$$S_2 \equiv (\text{mayor subconjunto de } V \setminus W / j \in S_2, S_1 \times S_2 \in \Pi \text{ y } (j) \times (S_2 - (j)) \subset \Pi)$$

siempre que $\exists k \in V \setminus (W \cup S_1 \cup S_2)$ tal que $(i, k) \in \Pi$ y $(k, j) \in \Pi$.

Ejemplo 4.2: Ver figura 4.3

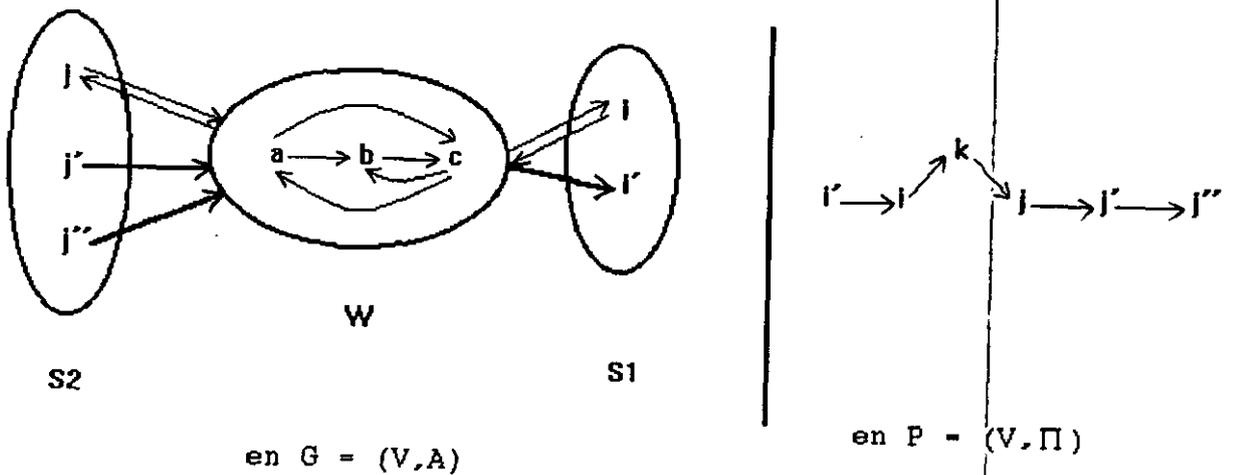


Figura 4.3

Caso 2: Dado un nudo j , si existe algún nudo $i \in P(j)$ tal que $i \in (0, \dots, j)$ en T , (ver figura 4.1b) identificamos en general una familia de desigualdades violadas con índice en Ω_2 . Podemos entonces introducir en el problema cortes del tipo (4.8) proporcionados por el teorema 4.2, definiendo el siguiente conjunto W , tal que

$$* W = \{ l \in V / l \in (0, \dots, j), l \neq 0, l \neq j \} \text{ en } T.$$

De acuerdo con la elección de los conjuntos S_1 y S_2 tales que verifiquen las hipótesis del teorema 4.2 se obtendrán distintos cortes, que estarán violados siempre que $j \in S_1$, y serán más fuertes que el corte correspondiente (4.2b).

Sean Q y Q_1 dos subconjuntos de V tales que,

$$Q \equiv \{ \forall k \in V \setminus W / \exists i \in P(k), i \neq k, i \neq 0, i \in W \}$$

$$Q_1 \equiv \{ k \in Q / \exists (l, k) \in T \text{ con } l \in W \}$$

Observación 4.3: El nudo j siempre pertenece a Q_1 .

Proposición 4.1: Si se construyen los conjuntos S_1 y S_2 de forma que

$$\begin{aligned} S_1 &= Q \\ S_2 &= \emptyset \end{aligned} \tag{4.18}$$

el corte (4.8) obtenido es el corte más violado por la arborescencia T de entre todos los cortes posibles proporcionados por el teorema 4.2.

Demostración:

Sea el corte (4.8) $x(0;W) + x(S_2;W) + x(W) + x(W;S_1) \leq |W|$.

La solución actual x es la correspondiente a la arborescencia T , y el conjunto W es tal que $W = \{ l \in V / l \in (0, \dots, j) \text{ en } T, l \neq 0, l \neq j \}$. Entonces:

- * $x(W)$ y $|W|$ son constantes para toda elección de S_1 y S_2 .
- * El único arco con origen en algún nudo de $V \setminus W$ y destino en algún nudo de W es el arco con origen en el nudo 0. Así, para la solución actual,

$$x(0:W) = 1$$

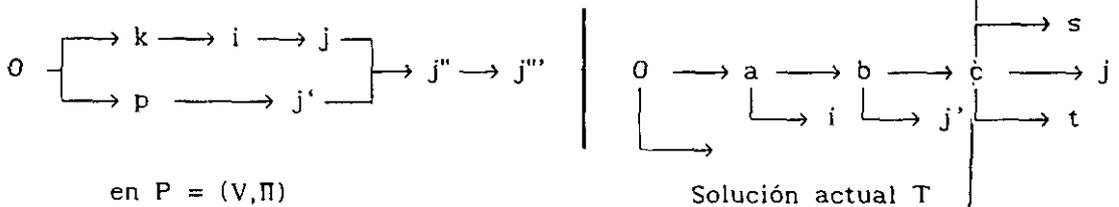
$$x(S'_2:W) = 0 \quad \forall S'_2 \subset V / 0 \notin S'_2 \text{ y, por tanto, } x(S_2:W) = 0$$

* Por consiguiente, el corte (4.8) estará más violado cuanto mayor sea $x(W:S_1)$. Cualquier $S'_1 \neq Q$ es tal que $S'_1 \subseteq Q$ (para que se verifiquen las hipótesis del teorema 4.2), y entonces $x(W:S_1) \geq x(W:S'_1)$.

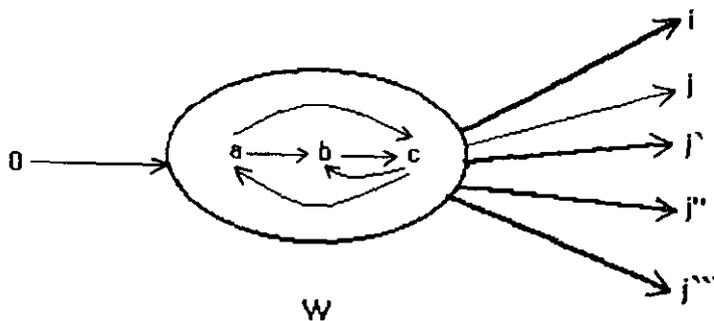
Así, el corte (4.8) obtenido a partir de los conjuntos S_1 y S_2 definidos según (4.18) es el más violado. ■

Observación 4.4: Los únicos arcos que influyen en la violación del corte (4.8) son los arcos de la forma $(i,k) \in T$ con $i \in W$, $k \in Q_1$. Por tanto todos los cortes obtenidos a partir de conjuntos S_1 y S_2 tales que $Q_1 \subseteq S_1$ son tan violados como el corte dado por la proposición 4.1.

Ejemplo 4.3: Ver figura 4.4.



$$Q = \{ i, j, j', j'', j''' \}$$



$$S_1 = \{ i, j, j', j'', j''' \}$$

$$S_2 = \emptyset$$

$$\text{en } G = (V, A)$$

Figura 4.4. Corte más violado

Proposición 4.2: Sea $S_1 \subseteq Q$ tal que $Q_1 \subseteq S_1$. Sea $S_2 \subseteq Q \setminus S_1$ tal que $S_1 \times S_2 \subseteq \Pi$. El corte (4.8) proporcionado por S_1 y S_2 es tan violado y más fuerte que el corte proporcionado por la proposición 4.1. si se verifica la siguiente condición:

$$|S_2:W| > |W:S_2| + |W:Q \setminus S_1 \cup S_2| \quad (4.19)$$

Demostración:

- * Si $Q_1 \subseteq S_1$ el corte es tan violado como el proporcionado por la proposición 4.1.
- * $|0:W|$ y $|W|$ son constantes para toda elección de S_1 y S_2 .
- * El corte (4.8) tiene además $|W:Q|$ variables en las hipótesis de la proposición 4.1, frente a $|S_2:W| + |W:S_1|$ variables en las hipótesis consideradas en la proposición 4.2. Utilizando la condición (4.19), se verifica:

$$|S_2:W| + |W:S_1| > |W:S_2| + |W:Q \setminus S_1 \cup S_2| + |W:S_1| = |W:Q|,$$

pues $S_1, S_2 \subseteq Q$.

■

Sea el siguiente algoritmo para identificar el corte más fuerte posible entre los más violados:

Algoritmo 4.1:

Paso 1: Hacer $S_1 = Q_1$.

Hacer $Q_2 \equiv$ máximo subconjunto de $Q \setminus S_1$ tal que $S_1 \times Q_2 \subseteq \Pi$.

Paso 2: Si $Q_2 = \emptyset$ ir al paso 7.

Paso 3: Si $|Q_2:W| \leq |W:Q_2| + |W:Q \setminus S_1 \cup Q_2|$ ir al paso 6.

Paso 4: Hacer $S_2 = Q_2$.

Paso 5: $\forall j^* \in Q \setminus S_1 \cup S_2$, si $\{j^*\} \times S_2 \subseteq \Pi$ hacer $S_1 = S_1 \cup \{j^*\}$.

Ir al paso 8.

Paso 6: $\forall Q'_2 \subset Q_2$, si $Q'_2 \times Q_2 \setminus Q'_2 \subseteq \Pi$ y $|Q'_2:W| < |W:Q'_2|$, hacer

$$\begin{aligned} S_1 &= S_1 \cup Q'_2 \\ Q_2 &= Q_2 \setminus Q'_2 \end{aligned}$$

Si S_1 ha sido modificado, ir al paso 2.

Paso 7: Hacer $S_1 = Q$
 $S_2 = \emptyset$

Paso 8: Fin. Se obtiene el corte $x(0:W) + x(S_2:W) + x(W) + x(W:S_1) \leq |W|$

Observación 4.5: Un conjunto de nudos Q'_2 en las hipótesis del paso 6 es candidato a pertenecer a S_1 o a S_2 . Introducirlo en S_1 en lugar de en S_2 supone una variación en el número de variables del corte (4.8) de $|W:Q'_2| - |Q'_2:W|$. Esta diferencia es positiva y, por tanto, la introducción de Q'_2 en S_1 hace más fuerte el corte.

Observación 4.6: Como el paso 6 se ejecuta $\forall Q'_2 \subset Q_2$, la complejidad de este algoritmo es no polinomial.

Observación 4.7: Si se sustituye el paso 6 del algoritmo por

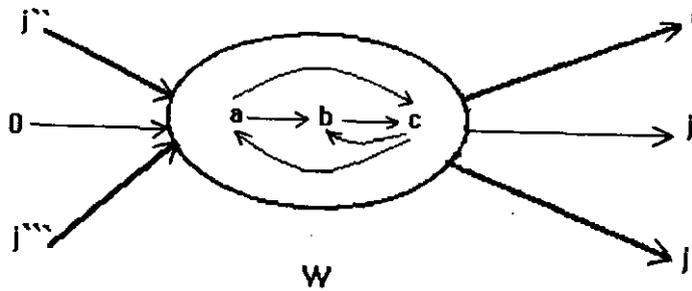
Paso 6: $\forall j^* \in Q_2$, si $\{j^*\} \times Q_2 \setminus \{j^*\} \subseteq \Pi$ y $|j^*:W| < |W:j^*|$ hacer

$$\begin{aligned} S_1 &= S_1 \cup \{j^*\} \\ Q_2 &= Q_2 \setminus \{j^*\} \end{aligned}$$

Si S_1 ha sido modificado, ir al paso 2.

se obtiene un algoritmo con complejidad cuadrática (y, por tanto, polinomial). Pero en este caso no hay garantía de que el corte obtenido sea el más fuerte entre los más violados, aunque, sí es uno de estos cortes y tan fuerte o más que el obtenido a partir de la proposición 4.1.

Ejemplo 4.4: Para la situación descrita en el ejemplo 4.3, el corte más fuerte obtenido por el algoritmo es el dado por la figura 4.5.



en $G = (V, A)$

si $|j^*:W| > |W:j^*|$ en A para $j^* \in \{j'', j'''\}$

Figura 4.5

Una vez obtenido el corte, a continuación se comprueba la condición impuesta en el corolario 4.2a para introducir, si es posible, el reforzamiento (4.9).

Caso 3: Para identificar desigualdades violadas con índices en Ω_4 :

1.- Dado un par de nudos (i, j) tales que $i \in P(j)$ y T contiene un subcamino de la forma (j, \dots, i) , se define el conjunto W de forma que,

$$* W = \{ l \in V / l \in (j, \dots, i), l \neq i, l \neq j \} \text{ en } T.$$

Dado el digrafo $P=(V, \Pi)$ de precedencias, sea P' el subgrafo generado por $V \setminus W$. Sea C un camino desde el nodo 0 hasta el nodo j en P' tal que $i \in C$, y sea C' el conjunto de nudos del camino C , tal que

$$C' = \{ i_1=0, i_2, \dots, i_k=i, i_{k+1}, i_{k+2}, \dots, i_{k+m}, i_{k+m+1}=j \}$$

Formamos el conjunto S del teorema 4.4 de la siguiente forma:

$$* \text{ Si } k \text{ es par y } m \text{ es impar} \Rightarrow S = C' \quad (4.20)$$

$$* \text{ Si } k \text{ es impar y } m \text{ es impar} \Rightarrow S = C' \quad (4.21)$$

$$* \text{ Si } k \text{ es par y } m \text{ es par} \Rightarrow S = C' \setminus \{ i_h \} \quad (4.22)$$

$$* \text{ Si } k \text{ es impar y } m \text{ es par} \Rightarrow S = C' \setminus \{ i_h \} \quad (4.23)$$

$$\text{donde} \quad i_h \in \{ i_{k+1}, \dots, i_{k+m} \} \quad (4.24)$$

Sean S_1^h y S_2^h los conjuntos formados a partir del conjunto S^h , donde

$$S^h = C' \setminus \{ i_h \} \quad \text{con} \quad h = \emptyset \quad \text{ó} \quad h = k+1, \dots, k+m \quad (4.25)$$

tal que $h = \emptyset$ significa que no se ha eliminado ningún nudo del camino C, de forma que:

Si k es par (casos (4.20) y (4.22)),

$$\begin{aligned} S_1^h &= \{ i_n \in S^h / n \text{ es par} \}, \\ S_2^h &= \{ i_n \in S^h / n \text{ es impar} \} \end{aligned} \quad (4.26)$$

Si k es impar (casos (4.21) y (4.23)),

$$\begin{aligned} S_1^h &= \{ i_n \in S^h / n \text{ es impar} \}, \\ S_2^h &= \{ i_n \in S^h / n \text{ es par} \} \end{aligned} \quad (4.27)$$

La desigualdad (4.14) para $S_1 = S_1^h$ es violada por la arborecencia T, dado que $i, j \in S_1^h$, y es más fuerte que la desigualdad (4.2a).

Dependiendo de la elección del nudo i_h a eliminar (en caso de que sea necesario), se obtiene una familia ξ de cortes no dominados e

igual de fuertes (e.d., con el mismo número de variables).

Sea $D_1^h = \{ (a,b) \in T \mid a \in W, b \in S_1^h \}$ para cada h dado. El corte (4.14) más violado de la familia ξ por la solución actual T será aquel tal que $|D_1^h|$ sea máximo.

Sea $D_2^h = \{ (a,b) \in T \mid a \in W, b \in S_2^h \}$ para cada h dado. El corte (4.15) para $S_2 = S_2^h$ estará violado por la arborescencia T si $|D_2^h| > |W|$. Si se obtiene una familia de cortes violados del tipo (4.15), el corte más violado por la solución actual T será aquel tal que $|D_2^h|$ sea máximo.

Proposición 4.3: Si no existe ninguna elección de h tal que $i, j \in S_1^h$ (i.e., $\exists l \in V \setminus W$ tal que $(i,l), (l,j) \in T$) entonces no existe garantía de que la arborescencia T viole ningún corte en Ω_4 .

Demostración:

* Construidos los conjuntos S_1^h y S_2^h según las expresiones (4.26) o (4.27), se verifica que $i \in S_1^h$ siempre. En las hipótesis de la proposición, $j \in S_2^h$.

* Para la solución actual x correspondiente a la arborescencia T se verifica:

$$x(S_1^h:W) = 0 \quad (j \notin S_1^h).$$

$$x(S_2^h:W) = 1$$

$$x(W) = |W| - 1$$

Si $x(W:S_1^h) = 1$ (i.e., no existen arcos de W a S_1^h excepto al nudo i), entonces la desigualdad (4.14) no es violada por la solución x .

Si $x(W:S_2^h) = 0$ (i.e., no existen arcos de W a S_2^h), entonces la desigualdad (4.15) no es violada por la solución x . ■

Observación 4.8: Es preciso notar que los arcos de j a W y de W a i no intervienen en el corte (4.15) salvo en las hipótesis de la proposición 4.3.

Observación 4.9: Si $h = \emptyset$ entonces $S = C'$ y se obtiene un único corte del tipo (4.14) y un único corte del tipo (4.15) como máximo.

Obtención del camino C

Cualquier camino C en el digrafo P' (excepto aquellos que satisfagan la condición de la proposición 4.3) proporciona una o más desigualdades violadas. El término del lado derecho es el mismo para cualquier corte.

Si se toma C como el camino máximo del nudo O al nudo j pasando por i (donde se considera que todos los arcos de Π tienen igual longitud), entonces se obtiene la familia de cortes más fuertes (i.e., con mayor número de variables), pero no necesariamente el corte más violado por la solución actual T .

Por el contrario, si el camino máximo se obtiene asignando pesos superiores a aquellos arcos incidentes en el conjunto de vértices $\{i_n \in V / \exists \text{ algún arco de } W \text{ a } i_n \text{ en la arborescencia } T\}$, entonces se obtienen cortes en general con menos número de variables, pero más violados por la solución actual.

En concreto, a la vista de la experiencia computacional desarrollada, proponemos la utilización del primer procedimiento para la obtención de cortes con el mayor número de variables, que facilitarán la utilización de las técnicas de reforzamiento descritas en el capítulo 5. En aquellos casos en los que, para un número determinado de iteraciones, no se consiga una deteriorización fuerte de la función objetivo, se introducirá el corte obtenido a partir del segundo procedimiento.

2.- Dado un nudo j , si existe algún nudo $i \in P(j)$, y T contiene un subcamino $(0, \dots, j)$ tal que $i \notin (0, \dots, j)$, se define el conjunto W tal que,

$$* W = \{ l \in V / l \in (0, \dots, j), l \neq 0, l \neq j \} \text{ en } T.$$

Dado el digrafo $P=(V, \Pi)$ de precedencias, sea P' el subgrafo generado por $V \setminus W$. Sea C un camino desde el nudo 0 hasta el nudo j en P' , y sea C' el conjunto de nudos del camino C , tal que

$$C' = \{ i_1=0, i_2, \dots, i_k, i_{k+1}=j \}$$

Formamos el conjunto S del teorema 4.4 de la siguiente forma:

$$* \text{ Si } k \text{ es par } \Rightarrow S = C'$$

$$* \text{ Si } k \text{ es impar } \Rightarrow S = C' \setminus \{ i_h \} \quad (4.28)$$

$$\text{donde } i_h \in \{ i_2, \dots, i_k \} \quad (4.29)$$

Sean S_1^h y S_2^h los conjuntos formados según (4.12) y (4.13) respectivamente, a partir del conjunto S^h , donde

$$S^h = C' \setminus \{ i_h \} \quad \text{con } h = \emptyset \text{ ó } h = 2, \dots, k \quad (4.30)$$

tal que $h = \emptyset$ significa que $S^h = C'$.

La desigualdad (4.14) para $S_1 = S_1^h$ es violada por la arborescencia T , dado que $0, j \in S_1^h$, y es más fuerte que la desigualdad (4.2b).

Dado un camino C , y dependiendo de la elección de i_h , se obtiene una familia ξ de cortes no dominados e igual de fuertes (e.d., con el mismo número de variables).

Sea $D_1^h = \{ (a, b) \in T / a \in W, b \in S_1^h \}$ para cada h dado. El corte (4.14) más violado de la familia ξ por la solución actual T será aquel tal que $|D_1^h|$ sea máximo.

Sea $D_2^h = \{ (a,b) \in T \mid a \in W, b \in S_2^h \}$ para cada h dado. El corte (4.15) para $S_2 = S_2^h$ estará violado si $|D_2^h| > |W|$. Si se obtiene una familia de cortes violados del tipo (4.15), el corte más violado por la solución actual T será aquel tal que $|D_2^h|$ sea máximo.

4.2.3.- Ejemplo de identificación de restricciones violadas.

Sea el grafo dirigido $G = (V,A)$ de la figura 4.6, y sea el digrafo de precedencias $P = (V,\Pi)$ tal que $\{(2,3),(3,4),(4,5),(5,8),(8,9)\} \subseteq \Pi$ y Π es cerrado transitivamente. El nudo 1 hace las veces de nudo auxiliar, y, por tanto, $(1,v) \in \Pi \forall v \in V$.

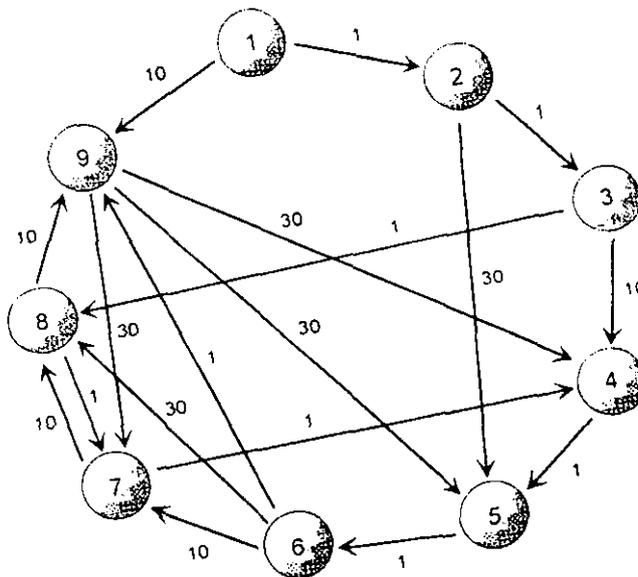


Figura 4.6

La arborescencia T de mínimo peso en el grafo G es:

$$1 \rightarrow 2 \rightarrow 3 \rightarrow 8 \rightarrow 7 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 9$$

es decir, se obtiene una primera solución relajada haciendo

$$x_{12} = x_{23} = x_{38} = x_{87} = x_{74} = x_{45} = x_{56} = x_{69} = 1$$

$$x_{ij} = 0 \quad \forall (i,j) \notin T$$

Esta solución no viola ninguna condición de asignación (i.e., es un camino hamiltoniano en G), pero sí viola dos relaciones de precedencia: (4,8) y (5,8). Por tanto necesitamos identificar cortes válidos inducidos por estas relaciones que sean violados por la solución x actual.

Cortes inducidos por la precedencia (4,8):

* El corte de tipo (4.2a) que se obtiene es $x_{87} + x_{74} \leq 1$

* Se identifica un corte de tipo 1 haciendo

$$W = \{ 7 \}$$

$$S_1 = \{ 2,3,4,5 \}$$

$$S_2 = \{ 8,9 \}$$

obteniéndose el corte $x_{87} + x_{97} + x_{74} \leq 1$, que es más fuerte que el corte (4.2a).

No se verifican las hipótesis del corolario 4.1a, por lo tanto no se puede reforzar disminuyendo el término independiente.

* Se identifica un corte de tipo 3 haciendo

$$W = \{ 7 \} \quad i = 4$$

$$S_1 = \{ 2,3,4 \} \quad j = 8$$

$$S_2 = \{ 8,9 \} \quad k = 5$$

obteniéndose el corte $x_{87} + x_{97} + x_{78} + x_{74} \leq 1$

Este corte es un reforzamiento del corte de tipo 1 obtenido anteriormente.

* Para identificar posibles cortes de tipo 4, sea

$$W = \{ 7 \}$$

El camino C de longitud máxima en P' del nudo 1 al nudo 8 es

$$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 8$$

Por tanto, sea $C' = \{ 1,2,3,4,5,8 \}$

$$S = C'$$

$$S_1 = \{ 2,4,8 \}$$

$$S_2 = \{ 1,3,5 \}$$

El conjunto S_1 proporciona el corte $x_{87} + x_{78} + x_{74} \leq 1$

Este corte está dominado por el corte de tipo 3 anterior.

El conjunto S_2 no proporciona ningún corte violado, ya que no hay ningún arco en T de un nudo en W a un nudo en S_2 , y por tanto, $|D_2| = 0 < |W|$.

Cortes inducidos por la precedencia (5,8):

* El corte de tipo (4.2a) que se obtiene es $x_{87} + x_{74} + x_{45} \leq 2$

* Se identifica un corte de tipo 1 haciendo

$$W = \{ 7,4 \}$$

$$S_1 = \{ 2,3,5 \}$$

$$S_2 = \{ 8,9 \}$$

obteniéndose el corte $x_{87} + x_{97} + x_{74} + x_{45} + x_{94} \leq 2$

No se verifican las hipótesis del corolario 4.1a, por lo tanto no se puede reforzar disminuyendo el término independiente.

* Para cualquier elección de S_1 y S_2 no existe ningún k verificando las hipótesis del teorema 4.3. Por tanto no es posible obtener ningún corte de tipo 3 para esta precedencia violada.

* Para identificar posibles cortes de tipo 4, sea

$$W = \{ 4,7 \}$$

El camino C de longitud máxima en P' del nudo 1 al nudo 8 es

$$1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 8$$

Por tanto, sea $C' = \{ 1,2,3,5,8 \}$

Estamos en las hipótesis de la proposición 4.3, ya que el nudo 5 es inmediatamente anterior al nudo 8 en C . Así, no tiene porqué haber ningún corte de tipo 4 violado. En este caso, sea

$$S_1 = \{ 2,5 \}$$

$$S_2 = \{ 1,3,8 \}$$

El conjunto S_1 no proporciona ningún corte válido para el problema.

El conjunto S_2 proporciona el corte $x_{87} + x_{74} + x_{78} \leq 2$, válido pero no violado por la solución actual x .

Introducidos estos cortes en el problema y estimados sus multiplicadores de Lagrange se obtiene un nuevo vector de coeficientes en la función objetivo del problema relajado. La solución a este nuevo problema es la arborescencia

$$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \begin{matrix} \nearrow 8 \\ \searrow 9 \end{matrix} \rightarrow 7$$

En este caso, la condición de asignación correspondiente al nudo 6 está violada, y habría que activarla. Además, la precedencia (8,9) está violada, y dará lugar a cortes de tipo 2.

Cortes inducidos por la precedencia (8,9):

* Para esta precedencia violada, el corte 4.2b que se obtiene es

$$x_{12} + x_{23} + x_{34} + x_{45} + x_{56} + x_{69} \leq 5$$

* Para identificar cortes de tipo 2, sea

$$W = \{ 2,3,4,5,6 \}$$

$$Q = \{ 8,9 \}$$

$$Q_1 = \{ 8,9 \}$$

Utilizando la proposición 4.1, se obtiene el corte más violado de entre los posibles cortes de tipo 2 haciendo

$$S_1 = \{ 8,9 \}$$

$$S_2 = \emptyset$$

obteniendose el corte (más fuerte que el corte 4.2b)

$$x_{12} + x_{23} + x_{34} + x_{45} + x_{56} + x_{25} + x_{38} + x_{68} + x_{69} \leq 5$$

Se obtiene otro posible corte de tipo 2 haciendo

$$S_1 = \{ 8 \}$$

$$S_2 = \{ 9 \}$$

Como $|S_2:W| = 2$, $|W:S_2| = 1$ y $|W:Q \setminus S_1 \cup S_2| = 0$, se verifican las hipótesis de la proposición 4.2 ($2 > 1 + 0$). Por tanto el corte

$$x_{12} + x_{23} + x_{34} + x_{45} + x_{56} + x_{25} + x_{38} + x_{68} + x_{94} + x_{95} \leq 5$$

obtenido es tan violado como el anterior y más fuerte, ya que se ha incrementado el número de variables.

4.3.- CORTES VALIDOS INDUCIDOS POR LAS ACOTACIONES SUPERIORES AL PESO ACUMULADO DE CADA NUDO

4.3.1.- Obtención de cortes válidos.

Se trata de obtener cortes válidos para el SOP tales que eliminen aquellas soluciones que no satisfagan la condición (4.1).

El primer corte propuesto es el corte (4.3), $x(0, \dots, j) \leq |W_j| - 1$, con $W_j = A(0, \dots, j)$; por tanto se prohíbe el camino $(0, \dots, j)$ si $y_j > d_j$. Veamos como reforzar ese corte.

Ejemplo 4.5: (Ver figura 4.7). Sea el camino $(0, i, j, g)$ tal que viola la condición (4.3). Es fácil ver que el corte $x_{0i} + x_{ij} + x_{jg} + x_{oj} + x_{ji} + x_{ig} \leq 2$ proporciona una desigualdad válida para el problema original, siempre que el peso acumulado del camino permutado $(0, j, i, g)$ también viole la cota del nudo g , d_g .

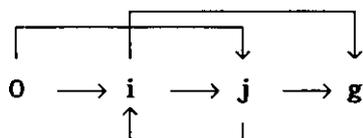


Figura 4.7

Dado una camino $p = (0, i(1), \dots, i(k)=g)$, con $P = N(p)$ donde $N(p)$ es el conjunto de nudos en el camino p , sea $\mathcal{G} = (P, E)$ un subgrafo parcial de G . Inicialmente $E = A(p)$, donde $A(p)$ es el conjunto de arcos del camino p . Se trata de ir aumentando el conjunto de arcos E añadiendo nuevos arcos provenientes de permutaciones de la forma $p' = (0, i'(1), \dots, i'(k)=g)$ del camino p cuyos arcos no estén todos incluidos en E , a las que se denominará *permutaciones explícitas*.

Definición 4.1: Dado un camino $p = (0, i(1), \dots, i(k)=g)$ desde el nudo 0 al nudo g , llamaremos *longitud de p* , sea L_p , al peso acumulado del nudo g , tal que T_{ab} es el peso del arco $(a,b) \in p$. Por tanto,

$$L_p = \sum_{(a,b) \in p} T_{ab} \quad (4.31)$$

El objetivo es identificar secuencialmente permutaciones explícitas p' de un camino p dado que verifique $L_p > d_g$, de forma que, añadiendo los arcos no redundantes de $A(p')$ a E , el corte $x(E) \leq |W_g| - 1$ sea todavía un corte válido. Es preciso notar que no todas las permutaciones de p son permutaciones que permiten añadir arcos al conjunto E . Sean los siguientes conceptos:

Definición 4.2: Dado un camino $p^0 = (0, i(1), \dots, i(k)=g)$, y el subgrafo parcial $\mathcal{G} = (P, E)$, donde $P = N(p^0)$ y $E = \bigcup_{i=0}^j A(p^i)$ con p^i permutación explícita de $p^0 \forall i=1, \dots, j$, llamaremos *permutación implícita p^** de p^0 en \mathcal{G} a todo camino del nudo 0 al nudo g en el grafo \mathcal{G} que pase por todos los nudos y tal que $p^* \neq p^i \forall i=0, \dots, j$.

Definición 4.3: Llamaremos *permutación permisible* a una permutación explícita p' de un camino $p = (0, \dots, g)$ con $L_p > d_g$, tal que satisfaga la siguiente condición:

$$L_p > d_g, \text{ y no hay ninguna permutación implícita } p^* \text{ en } \mathcal{G}' = (P, E') \text{ con } L_{p^*} \leq d_g, \text{ donde } E' = E \cup \{(k, l) \in p'\}. \quad (4.32)$$

Así, el corte (4.3) se puede reforzar utilizando el corte definido en el siguiente teorema:

Teorema 4.5: Sea $\mathcal{G} = (P, E)$ un subgrafo parcial de G tal que inicialmente $E = A(p)$, donde p es un camino dado $p = (0, \dots, g)$, $P = N(p)$ y $L_p > d_g$.

Entonces, el corte

$$x(E) \leq |W_g| - 1 \tag{4.33}$$

es un corte válido para el SOP con $E \leftarrow E \cup \{(k,l) \in p'\}$ si y sólo si p' es una permutación permisible de p .

Demostración

⇐) Cualquier camino en el grafo \mathcal{G} del nudo $\dot{0}$ al nudo g que pase por todos los nudos tiene $|W_g|$ arcos, y su longitud es superior a la máxima permitida por definición de permutación permisible. Por tanto, ninguno de estos caminos es factible para el problema original, de donde el corte (4.33) es válido al no eliminar ninguna solución factible.

⇒) Si al introducir los arcos de una permutación p' en E el corte (4.33) sigue siendo válido es debido a que no se ha eliminado ninguna solución factible. Por tanto al introducir p' no se ha formado ninguna permutación implícita p^* con $L_p^* \leq d_g$ y, entonces, p' es permisible. ■

Ejemplo 4.6: El siguiente ejemplo ilustra la necesidad de que se verifique la condición (4.32) para que el nuevo corte siga siendo válido. Ver figura 4.8.

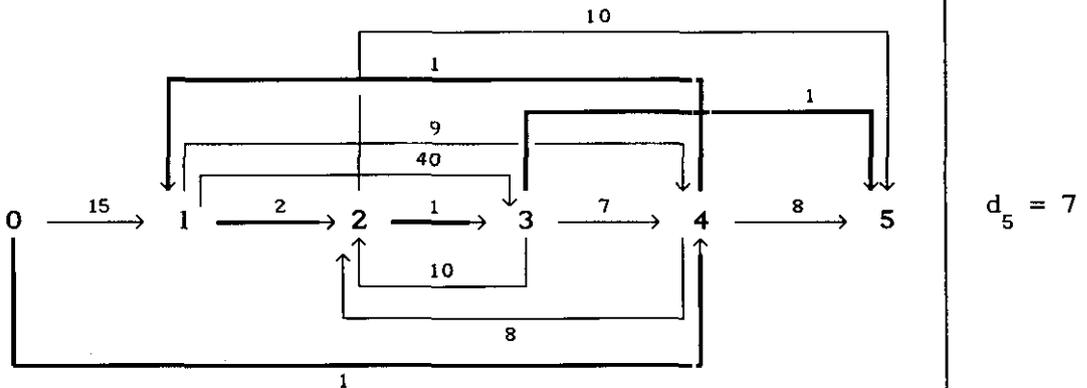


Figura 4.8

Sea el camino inicial $p = (0,1,2,3,4,5) / L_p = 33 > d_5 = 7$. Supongamos que identificamos el camino $q = (0,1,4,2,3,5) / L_q = 34 > d_5$. Así, inicialmente, $\mathcal{G}=(P,E)$, donde $P = (0,1,2,3,4,5)$, $E = \{(0,1),(1,2), (2,3),(3,4),(4,5),(1,4),(4,2),(3,5)\}$, y c_{ik} es la longitud del arco $(i,k) \in E$. Supongamos ahora que queremos añadir el camino $p' = (0,4,1,3,2,5) / L_{p'} = 62 > d_5$. El conjunto de arcos del grafo \mathcal{G}' sería $E \cup \{(0,4),(4,1),(1,3),(3,2),(2,5)\}$. Finalmente, sea $p^* = (0,4,1,2,3,5)$ una permutación implícita en el grafo \mathcal{G}' .

Adviertase que $x(p^*) = 5$. $L_{p^*} = 6 \leq d_5 = 7$, pero el corte (4.33) no permitiría este camino, a pesar de que no viola la acotación superior al peso acumulado del nudo 5. Por tanto la condición (4.32) debe ser satisfecha por cualquier camino p' para poder introducirlo en el corte y que este siga siendo un corte válido para el SOP.

Observación 4.10: Cuantas más permutaciones permisibles se identifiquen, más fuerte será el corte (4.33).

Observación 4.11: La verificación de la condición (4.32) puede consumir mucho tiempo. Por ello se introducirá un concepto de más fácil verificación que el concepto de permutación permisible.

Sea $\mathcal{G} = (P,E)$ un subgrafo parcial de $G = (V,A)$ tal que inicialmente $E = A(p)$, donde p es un camino dado $p = (0, \dots, g)$. $P = N(p)$, $L_p > d_g$ y T_{ab} es la longitud asociada a un arco $(a,b) \in A$. Sea s_{0k} la distancia mínima desde el nudo 0 al nudo k en el digrafo \mathcal{G} . Sea $\hat{\mathcal{G}} = (P, \hat{E})$ el digrafo inducido por \mathcal{G} tal que $(b,a) \in \hat{E}$ si y sólo si $(a,b) \in E$ y $T_{ba} = T_{ab}$. Sea \hat{s}_{gm} la distancia mínima desde el nudo g hasta el nudo m en el digrafo $\hat{\mathcal{G}}$. Dada una permutación p' , sea $\mathcal{G}' = (P, E')$, con $E' = E \cup A(p')$.

Observación 4.12: La distancia mínima desde el nudo 0 al nudo g en

$p=(0,\dots,g)$ no corresponde necesariamente a la longitud de una permutación de nudos en $P=N(p)$.

Definición 4.4: Llamaremos *permutación aceptable* a una permutación p' de un camino $p = (0,\dots,g)$ con $L_p > d_g$ tal que satisfaga la siguiente condición:

$$s'_{0g} > d_g, \text{ donde } s'_{0g} \text{ es la distancia mínima del nudo } 0 \text{ al nudo } g \text{ en el digrafo } \mathcal{G}'. \quad (4.34)$$

Lema 4.1: Toda permutación aceptable es una permutación permisible, pero no viceversa.

Demostración

a) Sea p' una permutación aceptable. Entonces, por definición, $s'_{0g} > d_g$ en el digrafo \mathcal{G} . Por tanto, todo camino de 0 a g tiene longitud mayor que d_g y, en particular, $L_{p'} > d_g$ y toda permutación implícita p^* en \mathcal{G}' es tal que $L_{p^*} > d_g$. Así, p' es permisible. ■

b) No toda permutación permisible es aceptable. Ver Ejemplo 4.7.

Ejemplo 4.7:

Sea el camino $p = (0,1,2,3,4,g)$, con $L_p = 20 > d_g = 5$, y la permutación permisible $p' = (0,4,1,2,3,g)$, con $L_{p'} = 17 > 5$. Por tanto, el grafo $\mathcal{G} = (P,E)$ es tal que $P = N(p)$ y $E = A(p) \cup A(p')$.

La permutación $p'' = (0,2,4,3,1,g)$ es una permutación permisible, ya que $L_{p''} = 11 > 5$ y toda permutación implícita que se forme con su introducción tiene, obviamente, 5 arcos y, por tanto, longitud mayor o igual a 8, suma de los pesos de los cinco arcos de menor peso en \mathcal{G}' .

Sin embargo en el nuevo grafo $\mathcal{G}' = (P,E')$ con $E' = E \cup A(p'')$, se

cumple que $s'_{0g} = 4 < 5$ y, por tanto, p'' no es aceptable.

Nota: La distancia mínima $s'_{0g} = 4$ corresponde al camino $(0,2,4,1,g)$.

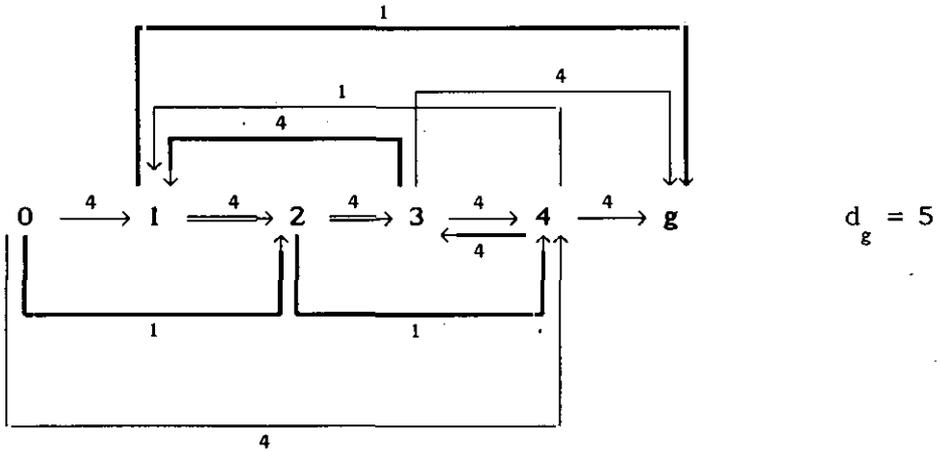


Figura 4.9

Corolario 4.3a: El corte $x(E) \leq |W_g| - 1$ es válido para $E \leftarrow E \cup A(p')$, donde p' es una permutación aceptable de p .

Definición 4.5: Una permutación p' de p es *descartable* si satisface la condición

$$s_{0k} + T_{kl} + \hat{s}_{gl} \leq d_g \quad (4.35)$$

para algún arco $(k,l) \in p'$.

Lema 4.2: Toda permutación descartable es no aceptable.

Demostración

Supongamos que la condición (4.35) se satisface para un arco $(k,l) \in p'$. Sea B el siguiente conjunto de arcos:

$$B = \{(r,s) \in CM(0,\dots,k)\} \cup \{(k,l)\} \cup \{(a,b) / (b,a) \in CM(g,\dots,l)\} \quad (4.36)$$

donde

$CM(0,\dots,k)$ es el conjunto de arcos que forman parte del camino mínimo del nudo 0 al nudo k en \mathcal{G} .

$CM(g,\dots,l)$ es el conjunto de arcos que forman parte del camino mínimo del nudo g al nudo l en $\hat{\mathcal{G}}$.

Los arcos en B forman un camino desde el nudo 0 hasta el nudo g en $\mathcal{G}' = (P, E')$ donde $E' = E \cup \{(k,l) \in p'\}$, tal que su longitud es menor o igual a d_g . Por tanto, la longitud del camino mínimo del nudo 0 al nudo g en \mathcal{G}' es menor o igual a d_g y la permutación p' no es aceptable. ■

Observación 4.13: El recíproco del lema 4.2 no es cierto, como se puede ver en el siguiente ejemplo:

Ejemplo 4.8: Sea el grafo $\mathcal{G} = (P, E)$ del ejemplo 4.7. Supongamos que queremos introducir la permutación p'' . Se calcula en \mathcal{G} y $\hat{\mathcal{G}}$ s_{0k} y \hat{s}_{gk} $\forall k \in \{0, \dots, g\}$.

$s_{01} = 4$	$s_{02} = 8$	$s_{03} = 12$	$s_{04} = 4$
$\hat{s}_{g1} = 12$	$\hat{s}_{g2} = 8$	$\hat{s}_{g3} = 4$	$\hat{s}_{g4} = 4$

A continuación se verifica la condición (4.35) $\forall (k,l) \in p''$:

$(0,2) \rightarrow 0 + 1 + 8 > 5$	$(3,1) \rightarrow 12 + 4 + 12 > 5$
$(2,4) \rightarrow 8 + 1 + 4 > 5$	$(1,g) \rightarrow 4 + 1 + 0 = 5$
$(4,3) \rightarrow 4 + 4 + 4 > 5$	

Así la permutación p'' es descartable y, por tanto, no aceptable, como se observaba en el ejemplo 4.7, pero no ha sido necesario aplicar un algoritmo de camino mínimo al grafo \mathcal{G}' . Por tanto, no se introducen sus arcos en \mathcal{G} .

Consideremos el mismo ejemplo, donde ahora se asigna al arco (0,1) longitud 5 en lugar de 4, e intentemos introducir la permutación p'' .

$$s_{01} = 5$$

$$\hat{s}_{g1} = 12$$

$$s_{02} = 9$$

$$\hat{s}_{g2} = 8$$

$$s_{03} = 13$$

$$\hat{s}_{g3} = 4$$

$$s_{04} = 4$$

$$\hat{s}_{g4} = 4$$

A continuación se comprueba la condición (4.35) $\forall(k,l) \in p''$:

$$(0,2) \rightarrow 0 + 1 + 8 > 5$$

$$(3,1) \rightarrow 13 + 4 + 12 > 5$$

$$(2,4) \rightarrow 9 + 1 + 4 > 5$$

$$(1,g) \rightarrow 5 + 1 + 0 > 5$$

$$(4,3) \rightarrow 4 + 4 + 4 > 5$$

La permutación p'' , por tanto, no es descartable. Sin embargo la permutación es no aceptable, pues en el grafo \mathcal{G}' $s'_{og} = 4 < 5$, donde s'_{og} sigue correspondiendo al camino (0,2,4,1,g).

Así, permutación no descartable \neq permutación aceptable.

Observación 4.14: permutación no descartable \neq permutación permisible.

Sea el ejemplo 4.6 correspondiente a la figura 4.8. En él la permutación p' no es permisible, pues $\exists p^*$ implícita / $L_p^* = 6 \leq d_5 = 7$.

Sin embargo ningún arco $(k,l) \in p'$ verifica la condición 4.35, ya que

$$(0,4) \quad 9 > 7$$

$$(1,3) \quad 56 > 7$$

$$(2,5) \quad 27 > 7$$

$$(4,1) \quad 28 > 7$$

$$(3,2) \quad 30 > 7$$

y, por tanto, p'' es no descartable.

Comprobando primero la condición (4.35) podemos descartar automáticamente permutaciones que no van a ser aceptables. Así, sólo será necesario aplicar un algoritmo de camino mínimo a aquellas permutaciones que no verifiquen (4.35) para ningún $(k,l) \in p'$. (Obsérvese que la complejidad del análisis de la condición (4.35) es sensiblemente inferior a la complejidad de un algoritmo de camino mínimo, ver sección 1.6.3).

Para reforzar el corte (4.3), $x(0, \dots, j) \leq |W_j| - 1$, se siguen en líneas generales los siguientes pasos:

Algoritmo 4.2:

Entrada: Un camino $p = (0, \dots, g)$ tal que $L_p > d_g$.

Inicialmente sea $\mathcal{G} = (P, E)$ con $P = N(p)$ y $E = A(p)$.

Paso 0: Determinar AM , donde AM es el conjunto de los $|W_g|$ arcos con peso mínimo en \mathcal{G} , conjunto que en este caso coincide con E . Hacer $I = 0$.

Paso 1: Dado el grafo $\mathcal{G} = (P, E)$, encontrar una permutación explícita p' de p . (Ver sección 4.3.2). Si no se encuentra ninguna nueva permutación o no se quiere reforzar más el corte, FIN.

Paso 2: Si $L_{p'} \leq d_g$ ir al paso 1.

Paso 3: Construir el conjunto AM' para el nuevo grafo $\mathcal{G}' = (P, E')$ con $E' = E \cup A(p')$. Si el peso total de los arcos en AM' es superior a d_g p' es permisible. Hacer $I = 0$ e ir al paso 7.

Paso 4: Determinar s_{01} y $\hat{s}_{g1} \forall l \in P$ en el grafo \mathcal{G} . Si $I=1$, $s_{01} \leftarrow s'_{01} \forall l \in P$, $I=0$.

Paso 5: Comprobar la condición (3.22) $\forall (l, k) \in p'$. Si p' es descartable ir al paso 1.

Paso 6: En \mathcal{G}' calcular $s'_{01} \forall l \in P$. Si $s'_{0g} \leq d_g$ p' no es aceptable. Ir al paso 1. En caso contrario hacer $I=1$ e ir al paso 7.

Paso 7: Hacer $E \leftarrow E \cup A(p')$ y $AM \leftarrow AM'$ en \mathcal{G} . El nuevo corte reforzado es $x(E) \leq |W_g| - 1$. Ir al paso 1.

4.3.2.- Identificación de permutaciones. Búsquedas locales.

Para generar permutaciones aceptables p' de un subcamino dado $p = (0, i(1), \dots, g) / L_p > d_g$ proponemos un procedimiento heurístico basado en búsquedas locales. Dado el camino p , el procedimiento básicamente consiste en encontrar aquellos otros caminos que pueden obtenerse por medio de un número limitado de intercambios de arcos del camino p por arcos no pertenecientes a dicho camino.

El procedimiento realiza dos tipos de búsquedas locales. El primer tipo, denominado TRIA, se basa en tres intercambios de arcos y tiene dos variantes. El segundo tipo, denominado B1, se basa en dos intercambios de arcos y una orientación inversa.

i) El tipo de búsqueda denominado TRIA-1 parte de un par de nudos $i, j \in p$. Supongamos que p es de la forma $(0, \dots, a(i), i, z(i), \dots, a(j), j, z(j), \dots, g)$. El nuevo camino que se considera es $(0, \dots, a(i), i, j, z(i), \dots, a(j), z(j), \dots, g)$ supuesto que estos arcos pertenecen a \bar{A} . Ver figura 4.10.

Observación 4.15: $i=0$ y $j=g$ es una elección válida para la búsqueda.

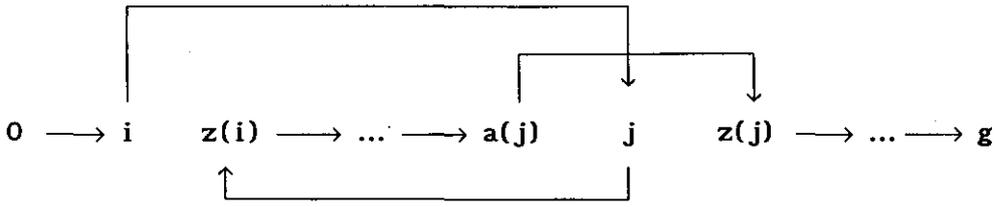


Figura 4.10

ii) El tipo de búsqueda TRIA-2 parte de un par de nudos $i, j \in p$, y considera el nuevo camino $(0, \dots, a(i), z(i), \dots, j, i, z(j), \dots, g)$, supuesto que estos arcos pertenecen a \bar{A} . Ver figura 4.11.

Observación 4.16: $a(i)=0$ y $j=g$ es una elección válida para la búsqueda.

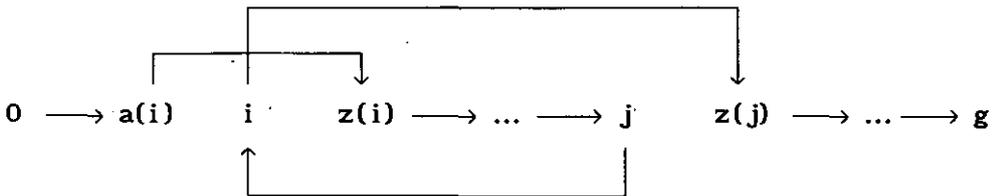


Figura 4.11

iii) El tipo de búsqueda B1 parte de un par de nudos $i, j \in p$, y considera el nuevo camino $(0, \dots, i, R(z(i), \dots, j), z(j), \dots, g)$, supuesto que estos arcos pertenecen a \bar{A} . Ver figura 4.12.

Observación 4.17: $i=0$ y $j=g$ es una elección válida para la búsqueda.

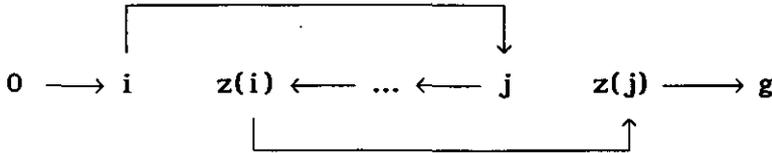


Figura 4.12

4.3.3.- Obtención del peso L_p de un camino permutación.

Dado un camino $p = (0, i(1), \dots, g)$ con peso L_p , sea una permutación $p' = (0, i'(1), \dots, g)$ obtenida con los tipos de búsqueda TRIA-1, TRIA-2 y B1. El peso $L_{p'}$ se obtiene de la forma siguiente:

Sea α_{lk} la ganancia que se obtiene insertando el nudo l entre el nudo k y el nudo $z(k)$ en el camino p . Esta ganancia se puede expresar como

$$\alpha_{lk} = c_{k,z(k)} - (c_{kl} + c_{l,z(k)}) \quad (4.37)$$

Sea β_k la ganancia que se obtiene eliminando el nudo k del camino p . Esta ganancia se puede expresar como

$$\beta_k = c_{a(k),k} + c_{k,z(k)} - c_{a(k),z(k)} \quad (4.38)$$

Tipo de búsqueda TRIA-1:

$$L_{p'} = L_p - (\alpha_{ij} + \beta_j) \quad (4.39)$$

Tipo de búsqueda TRIA-2:

$$L_{p'} = L_p - (\beta_i + \alpha_{ji}) \quad (4.40)$$

Tipo de búsqueda B1:

$$\begin{aligned}
 L_{(0,i'(1),\dots,j)} &= L_{(0,\dots,i)} + T_{ij} \\
 L_{(0,i'(1),\dots,j,\dots,v)} &= L_{(0,i'(1),\dots,j,\dots,z(v))} + T_{z(v),v} \quad \forall v \in R(a(j),\dots,z(i)) \\
 L_{(0,i'(1),\dots,j,\dots,z(i),z(j))} &= L_{(0,i'(1),\dots,j,\dots,z(i))} + T_{z(i),z} \\
 L_{p'} &= L_p - \Phi
 \end{aligned} \tag{4.41}$$

donde $\Phi \equiv L_{(0,1,\dots,z(j))} - L_{(0,i'(1),\dots,z(j))}$

Capítulo 5

Reforzamiento de cortes válidos

5.1.- INTRODUCCION

Sea el SOP expresado según el modelo 3.1. La solución óptima del problema del árbol generador de peso mínimo dado por las restricciones (1), (2) y (3) del modelo 3.1, junto con la condición de integralidad, y la dualización de las restricciones tipo (4), proporcionará una cota inferior para el SOP.

Si la solución obtenida satisface todas las restricciones del SOP, entonces es la solución óptima. En caso contrario, se trata de encontrar aquellas restricciones de tipos (5) (PFC's) y (6) (DFC's) que sean "fuertemente" violadas por la solución actual, y a partir de ellas obtener cortes válidos para el SOP. Estos cortes se introducirán, dualizados, en la función objetivo y, para ello, se estiman los correspondientes multiplicadores de Lagrange del nuevo problema, según la metodología descrita en la sección 3.3.

Sea el politopo

$$P_{SOP}^n = \text{CONV} \left\{ \begin{array}{l} x \in \{0,1\}^{|A|}, \text{ t.q.} \\ x \text{ satisface} \end{array} \right. \left. \begin{array}{l} 1) \bar{x}(A) = n \\ 2) x(\delta^-(j)) = 1 \quad \forall j \in V \setminus \{0\} \\ 3) x(W) \leq |W| - 1 \quad \forall W \subset V, |W| \geq 1 \\ 4) x(\delta^+(j)) \leq 1 \quad \forall j \in V \\ 5) x \text{ satisface las relaciones} \\ \text{de precedencia.} \\ 6) x \text{ satisface la acotación al} \\ \text{peso acumulado de cada nudo.} \end{array} \right\}$$

$$= \text{CONV} \left\{ x \in \{0,1\}^{|A|}, \text{ t.q. } Ax \leq b \right\}$$

donde P_{SOP}^n es un politopo 0-1 positivo, ya que A es una matriz de ceros y unos y, por tanto, todos los elementos de A son no negativos, y $b > 0$.

Para cada restricción violada, el corte válido que se obtiene de acuerdo con la metodología descrita en el capítulo 4, es una restricción tipo $ax \leq b$, donde el vector a contiene sólo ceros y unos, y el escalar b es entero positivo. Se puede intentar reforzar este corte válido antes de su dualización, utilizando la metodología que se describe en este capítulo. Así, en la sección 5.2 se desarrollan los fundamentos teóricos que permiten reforzar un corte utilizando la información proporcionada por otros cortes anteriormente identificados, mientras que en las secciones 5.3, 5.4 y 5.5 se desarrollan los algoritmos específicos que aplican la metodología a los distintos tipos de desigualdades obtenidas. La sección 5.6 desarrolla un ejemplo de identificación y reforzamiento de cortes para un programa SOP dado.

5.2.- FUNDAMENTOS TEORICOS

Las siguientes definiciones son necesarias:

Definición 5.1: Una desigualdad $\alpha x \leq \alpha_0$ con $\alpha_0 > 0$ válida para P_{SOP}^n es *maximal* si $\exists l \in N$ y $\alpha'_1 > \alpha_1$ tales que

$$\alpha'_1 x_1 + \sum_{j \in N \setminus \{1\}} \alpha_j x_j \leq \alpha_0$$

sea válida para P_{SOP}^n , con $N = \{1, 2, 3, \dots, |A|\}$

Definición 5.2: Una desigualdad $\alpha x \leq \alpha_0$ con $\alpha_0 > 0$ válida para P_{SOP}^n es *maximal para el coeficiente 1* si $\exists \alpha'_1 > \alpha_1$ tal que

$$\alpha'_1 x_1 + \sum_{j \in N \setminus \{1\}} \alpha_j x_j \leq \alpha_0$$

sea válida para P_{SOP}^n , con $N = \{1, 2, 3, \dots, |A|\}$

Definición 5.3: Un *clique* C es un conjunto de índices de variables 0-1 tal que sólo una de las variables con índices en C puede tomar valor uno. En este caso se dice que la desigualdad $x(C) \leq 1$ es la *desigualdad inducida por el clique* C .

Definición 5.4: Un clique C es *maximal* si no está dominado por ningún otro clique C' ; es decir, C es maximal si no hay otro clique C' tal que $C \subset C'$.

Definición 5.5: Un clique C es *trivial* si $|C| = 1$. Es preciso notar que en todo programa con variables 0-1 el clique formado por el índice de cada variable es trivial.

Definición 5.6: Un *cubrimiento* C es un conjunto índices de variables 0-1 tal que sólo un número determinado de variables con índices en C , sea k_C , puede tomar valor uno, con $0 < k_C < |C|$. En este caso se dice que la desigualdad $x(C) \leq k_C$ es *inducida por el cubrimiento* C .

Definición 5.7: Una desigualdad $\alpha x \leq \alpha_0$ es de *tipo mochila* si $\alpha_j \geq 0 \forall j \in N$ y $\alpha_0 > 0$.

Observación 5.1: Se supone que $\exists j \in N$ tal que $\alpha_j > \alpha_0$, y que $\sum_{j \in N} \alpha_j > \alpha_0$.

Definición 5.8: Se dice que un cubrimiento C está implicado por una desigualdad tipo mochila si existe un número entero, sea k_C para $0 < k_C < |C|$, tal que toda solución 0-1 factible para la desigualdad tipo mochila también es factible para la desigualdad $x(C) \leq k_C$.

Observación 5.2: Si la relación inversa es también cierta, entonces ambas desigualdades son equivalentes.

Proposición 5.1: Sea $F = \{ x \in P_{SOP}^n : \alpha x = \alpha_0 \}$ la cara definida por la desigualdad $\alpha x \leq \alpha_0$, válida para P_{SOP}^n (con $\alpha_0 > 0$). La desigualdad es maximal si y sólo si $\forall j \in N \exists x \in F$ tal que $x_j = 1$.

Demostración:

\Rightarrow $\alpha x \leq \alpha_0$ es maximal. Supongamos que $\exists j \in N$ tal que $\forall x \in F x_j = 0$.

Entonces cualquier punto x factible tal que $x_j = 1$ verifica $\alpha x < \alpha_0$.

Por tanto el coeficiente α_j se podría incrementar en $\text{Min} \{ \alpha_0 - \alpha x \}$, $\forall x$ factible y tal que $x_j = 1$, cantidad estrictamente positiva, o en cualquier cantidad si no existe un x con estas características, y la condición no sería maximal.

\Leftarrow Si $\forall j \in N \exists x \in F / x_j = 1$, entonces si incrementáramos el coeficiente α_j y lo sustituyéramos por $\alpha'_j > \alpha_j$, para ese punto x tendríamos:

$$\sum \alpha_i x_i = \alpha_0 \Rightarrow \sum \alpha_i x_i + \alpha'_j x_j > \alpha_0$$

desigualdad no válida para P_{SOP}^n .

Por tanto la desigualdad $\alpha x \leq \alpha_0$ es maximal. ■

Estudiemos a continuación el problema de identificar si una desigualdad que modelice una PFC ó una DFC, o algún corte inducido por ellas, es maximal, para, en caso de que no lo sea, obtener una condición más fuerte incrementando para ello alguno de los coeficientes de la desigualdad dada.

Observación 5.3: Todas las PFC's, DFC's y los cortes inducidos por ellas son desigualdades del tipo $a_r x \leq b_r$. Dada una restricción de este tipo, para cada $k \in \{1, \dots, |A|\}$ estudiemos si se puede aumentar el valor del coeficiente a_r^k en a_r , para $a_r^k \geq 0$.

Proposición 5.2: Sea $R_r^1 = \max \{ a_r x / Ax \leq b, x_1 = 1, x \in (0,1)^{|A|} \}$

a) Si $R_r^1 = b_r \forall l$ entonces $a_r x \leq b_r$ es maximal.

b) Si $\exists l / R_r^1 < b_r$ entonces $a_r x \leq b_r$ no es maximal, y el coeficiente a_r^1 se puede sustituir por $a_r^{1'} = a_r^1 + b_r - R_r^1$ siendo entonces maximal para el coeficiente l' la nueva condición.

Demostración:

a) Si $R_r^1 = b_r$ entonces $\exists x^* / a_r x^* = b_r$, con $x_1 = 1$

Si esto se verifica $\forall l \in \{1, \dots, |A|\}$, entonces, por la proposición 5.1, $a_r x \leq b_r$ es maximal.

b) Supongamos que $\exists l / R_r^1 < b_r$ y que sustituimos el coeficiente a_r^1 por $a_r^{1'} = a_r^1 + b_r - R_r^1 > a_r^1$. Veamos que la restricción

$$a_r^{1'} x_1 + \sum_{j \in N \setminus \{1\}} a_r^j x_j \leq b_r \quad (5.1)$$

sigue siendo válida para P_{SOP}^n :

Sea un punto factible $x = (x_1, \dots, x_1, \dots, x_{|A|})$. Veamos que x satisface la restricción (5.1):

* Si $x_1 = 0$ entonces $a_r^{1'} x_1 + \sum_{j \in N \setminus \{1\}} a_r^j x_j = \sum_{j \in N} a_r^j x_j \leq b_r$ por

hipótesis, ya que $a_r x \leq b_r$ era una desigualdad válida para P_{SOP}^n .

$$\begin{aligned}
 * \text{ Si } x_1 = 1, \quad R_r^1 &= \max \{ a_r^1 x / Ax \leq b, x_1 = 1, x \in \{0,1\}^{|A|} \} = \\
 &= a_r^1 + \max \{ \sum_{j \in N \setminus \{1\}} a_r^j x_j / Ax \leq b, x \in \{0,1\}^{|A|} \}
 \end{aligned}$$

$$\begin{aligned}
 \text{Por tanto } \max \{ a_r^1 x_1 + \sum_{j \in N \setminus \{1\}} a_r^j x_j / Ax \leq b, x_1 = 1 \} &= \\
 = a_r^1 + R_r^1 - a_r^1 &= a_r^1 + b_r - R_r^1 + R_r^1 - a_r^1 = b_r
 \end{aligned}$$

y la desigualdad sigue siendo válida para todo x factible de P_{SOP}^n . ■

La proposición 5.2 es por tanto una condición necesaria y suficiente para que una restricción $a_r^1 x \leq b_r$ sea maximal. Sin embargo para contrastar esta condición es necesario resolver un problema con la misma complejidad que el problema original. Buscamos por eso condiciones más débiles que puedan contrastarse en tiempo polinomial:

Proposición 5.3: Sea $R_{r_i}^1 = \max \{ a_r^1 x / a_i^1 x \leq b_i, x_1 = 1, x \in \{0,1\}^{|A|} \}$, tal que se resuelven independientemente estos problemas para $i \in M \subset I$, donde I es el conjunto de cortes válidos (reforzados o no) que se hayan identificado más las desigualdades tipo (4) del problema original.

Sea $\bar{R}_r^1 = \min_{i \in M} \{ R_{r_i}^1 \}$. Entonces

* Si $\exists i / \bar{R}_r^1 < b_r$ resulta que la condición no es maximal y el coeficiente a_r^1 se puede sustituir por $a_r^1 = a_r^1 + b_r - \bar{R}_r^1$

Observación 5.4: El problema a resolver para cada i para obtener $R_{r_i}^1$ es del tipo denominado mochila 0-1. Para $i = r$ el problema se denomina SSP (Subset Sum Problem), para el que hay algoritmos especiales. Para una panorámica de algoritmos mochila y SSP, ver Martello y Toth (1990). En especial se pueden utilizar los algoritmos descritos en Martello y Toth (1984) y (1988).

Demostración:

\bar{R}_r^1 es una cota superior al valor de R_r^1 . Por tanto es válida en este caso la demostración del apartado b de la proposición 5.2, teniendo en cuenta que si $x_1 = 1$ en el punto factible x ,

$$\begin{aligned} \max \{ a_r^1 x_1 + \sum_{j \in N \setminus \{1\}} a_r^j x_j / Ax \leq b, x_1 = 1, x_j \in (0,1)^{|A|} \} = \\ = a_r^1 + R_r^1 - a_r^1 = a_r^1 + b_r - \bar{R}_r^1 + R_r^1 - a_r^1 \leq b_r \end{aligned}$$

y por tanto la desigualdad sigue siendo válida para todo x factible de P_{SOP}^n . ■

La proposición 5.3 es una condición suficiente para incrementar los coeficientes individualmente. Así, el incremento de cada coeficiente da lugar a una nueva restricción, que puede reemplazar a la condición original. Si además un conjunto de variables, cuyos coeficientes han sido incrementados, forman parte de una estructura tipo clique, entonces se puede sustituir la restricción original por una nueva en la que todos los coeficientes con índices en el clique han sido aumentados.

Proposición 5.4: Dada una restricción $\alpha x \leq \alpha_0$, si existe un conjunto de variables con índices en C , tal que en un subconjunto $S \subseteq C$ todo coeficiente α_i , con $i \in S$, puede ser individualmente incrementado, y además existe una restricción del tipo $x(C) \leq 1$, entonces la restricción original se puede sustituir por una nueva en la que todos los coeficientes con índices en S han sido incrementados.

Demostración:

Obviamente la nueva restricción con todos los coeficientes con índices en S incrementados es más fuerte que la restricción original. Veamos que no elimina ninguna solución factible:

Cualquier solución factible es tal que sólo una de las variables con

índices en C , sea x_c , puede tomar valor uno. Si $c \notin S$ o todas las variables con índices en C toman valor cero, entonces la nueva condición es equivalente a la condición original. Si $c \in S$ entonces la nueva condición es equivalente a la condición obtenida al incrementar únicamente el coeficiente de x_c , condición válida por hipótesis. ■

Ejemplo 5.1:

Dada una restricción de la forma

$$\alpha_1 x_1 + \alpha_{1'} x_{1'} + \alpha_{1''} x_{1''} + \dots \leq \alpha_0 \quad (5.2)$$

supongamos que se ha conseguido un incremento individual de coeficientes para los índices 1 , $1'$ y $1''$. Tenemos entonces tres nuevas restricciones, cada una de las cuales podría reemplazar a la restricción original:

$$\alpha'_1 x_1 + \alpha_1 x_{1'} + \alpha_{1''} x_{1''} + \dots \leq \alpha_0 \quad (5.3)$$

$$\alpha_1 x_1 + \alpha'_1 x_{1'} + \alpha_{1''} x_{1''} + \dots \leq \alpha_0 \quad (5.4)$$

$$\alpha_1 x_1 + \alpha_1 x_{1'} + \alpha'_1 x_{1''} + \dots \leq \alpha_0 \quad (5.5)$$

Pero si además se cuenta con la restricción inducida por un clique, por ejemplo $x_1 + x_{1'} + x_{1''} \leq 1$, entonces se puede reemplazar cada una de las restricciones anteriores por:

$$\alpha'_1 x_1 + \alpha'_1 x_{1'} + \alpha'_1 x_{1''} + \dots \leq \alpha_0 \quad (5.6)$$

ya que sólo una de las variables x_1 , $x_{1'}$, $x_{1''}$ puede tomar valor uno en cada solución factible. Por tanto, la condición (5.6) es equivalente al sistema dado por las condiciones (5.3), (5.4) ó (5.5) y, obviamente, más fuerte.

Una vez obtenida la solución óptima para el problema del árbol generador de peso mínimo, es preciso modificar los coeficientes de la función objetivo para obtener una cota inferior del valor óptimo de SOP que sea más fuerte que la anterior. Sea el siguiente algoritmo básico:

Algoritmo 5.1:

Paso 1: Identificar cortes válidos inducidos por PFC's y DFC's "fuertemente" violadas por la solución óptima del problema relajado actual. (Nota: Estos cortes son desigualdades inducidas por cubrimientos). Ver capítulo 4.

Paso 2: Tratar de reforzar cada uno de los cortes válidos identificados anteriormente, a base de incrementar sus coeficientes. Para ello se utiliza la información proporcionada por cortes anteriormente identificados (reforzados o no) y restricciones inducidas por cliques. (Nota: La desigualdad resultante es de tipo mochila).

Paso 3: Inducir del corte reforzado el correspondiente conjunto de cliques maximales. Ver sección 5.5.

Paso 4: Utilizar este nuevo corte y las restricciones inducidas por cliques para reforzar los cortes, a su vez reforzados o no, anteriormente introducidos.

Paso 5: Estimar los multiplicadores de Lagrange correspondientes a los nuevos cortes, y resolver el nuevo problema relajado.

Definición 5.9: Dado un corte $ax \leq b$, se llama *densidad* del corte al cardinal del conjunto de variables $\{ x_i / a_i \neq 0 \}$.

Definición 5.10: Dada la condición $ax \leq b$ a reforzar, y una condición soporte $\alpha x \leq \alpha_0$, se llama *potencial de la condición soporte* al cardinal del conjunto de variables $\{ x_i / a_i \neq 0 \text{ y } \alpha_i \neq 0 \}$.

Observación 5.5: Dada una condición a reforzar, se puede utilizar como condición soporte cada una de las condiciones anteriormente identificadas. Para ello se van eligiendo las condiciones soporte en orden decreciente en cuanto a su potencial.

En los pasos 2 y 4 del algoritmo se trata de reforzar un corte utilizando en principio una sola condición, cada vez como soporte. Una vez calculado cuanto se puede incrementar cada coeficiente según las técnicas descritas en este capítulo, se decide sustituir el corte actual por un único corte reforzado. Para ello se tendrán en cuenta las siguientes observaciones:

Observación 5.6: La proposición 5.4 puede aplicarse en el caso de que exista algún clique de potencial mayor que uno. En ese caso se utilizará el clique identificado de máximo potencial.

Observación 5.7: Si no se puede incrementar simultáneamente más de un coeficiente de la condición a reforzar, ante la posibilidad de incrementar el coeficiente de una variable presente en el corte o introducir una variable nueva, preferimos incrementar la densidad del corte, puesto que hay indicios de que ese corte pueda ser soporte más frecuente de otras condiciones tipo cubrimiento y mochila a reforzar.

Estudiemos a continuación el problema planteado en la proposición 5.3. Se trata de obtener

$$R_{ri}^1 = \max \{ a_r x / a_i x \leq b_i, x_i = 1, x \in \{0,1\}^N \}$$

donde $ax \leq b$ puede ser una condición tipo cubrimiento o tipo mochila según se indica en los pasos anteriores. En todos los casos se verifica la siguiente proposición:

Proposición 5.5: Sea el problema

$$(P) \quad \begin{aligned} & \max a_r x \\ & a_i x \leq b_i \\ & x \in \{0,1\}^N \end{aligned} \tag{5.7}$$

Sea J_1 el conjunto de índices j tales que $x_j = 1$ en el óptimo de (P), y J_2

el conjunto de índices j tales que $x_j = 0$ en el óptimo de (P), i.e., $J_2 = N \setminus J_1$.

$$\text{Sea } R_{r_1} = \max \{ a_r x / a_1 x \leq b_1, x \in \{0,1\}^N \}$$

Entonces se verifica que $R_{r_1}^I = R_{r_1} \forall I \in J_1$.

Demostración:

Si $I \in J_1$ entonces $x_1^* = 1$ en x^* solución óptima de (P). Por tanto x^* es factible para el problema

$$\max \{ a_r x / a_1 x \leq b_1, x_1 = 1, x \in \{0,1\}^N \}$$

y si es óptimo para uno de los dos problemas lo es también para el otro.

Por tanto $R_{r_1}^I = R_{r_1}$. ■

Por tanto, según la proposición 5.5, sólo será necesario obtener $R_{r_1}^I$ cuando $I \in J_2$. En las dos secciones siguientes se estudiará el problema de la obtención de $R_{r_1}^I$ y $R_{r_1}^I$ para cada tipo de restricciones soporte y condiciones a reforzar.

5.3.- REFORZAR RESTRICCIONES TIPO CUBRIMIENTO.

Estudiemos ahora el problema planteado en la proposición 5.3, para el caso en el que se pretende reforzar una restricción tipo cubrimiento, problema que puede ser resuelto en tiempo polinomial: Obtener

$$R_{r_1}^I = \max \{ c_r x / a_1 x \leq b_1, x_1 = 1, x \in \{0,1\}^N \} \tag{5.8}$$

donde c_r es el vector de ceros y unos en la función objetivo. La restricción $a_1 x \leq b_1$ es de tipo mochila. Puede ser una restricción original del problema, es decir, un corte tipo cubrimiento sin reforzar, que denotaremos por $x(J_1) \leq k_1$, donde J_1 es el conjunto de índices de las variables con coeficiente positivo en la restricción, o puede ser una

restricción reforzada (con algún coeficiente entero mayor que uno) que sea válida para el problema, y que denotaremos por $\bar{x}(J_1) \leq k_1$.

Estudiemos cada uno de estos problemas:

5.3.1.- Caso 1: Una restricción tipo cubrimiento como soporte.

Problema: Incrementar los coeficientes de la restricción tipo cubrimiento $x(J_r) \leq k_r$, utilizando como soporte la restricción tipo cubrimiento $x(J_1) \leq k_1$, para $J_r \cap J_1 \neq \emptyset$.

Sea el siguiente problema para cada $l \in J$, donde $J = J_r \cup J_1$:

$$(C_l) \quad R_{rl}^1 = \text{Max} \{ x(J_r) / x(J_1) \leq k_1, x_1 = 1, x \in \{0,1\}^{|J|} \} \quad (5.9)$$

y sea el problema general

$$(C) \quad R_{rl} = \text{Max} \{ x(J_r) / x(J_1) \leq k_1, x \in \{0,1\}^{|J|} \} \quad (5.10)$$

Proposición 5.6: Sean R_{rl}^1 y R_{rl} los valores óptimos de los problemas (5.9) y (5.10) respectivamente, y sean los siguientes conjuntos de índices:

$$B = J_r \cap J_1, \quad A = J \setminus J_1, \quad D = J \setminus J_r$$

$J1$ = conjunto de índices correspondientes a variables que toman valor 1 en alguna solución óptima de (C), y

$J2$ = conjunto de índices correspondientes a variables que toman valor 0 en alguna solución óptima de (C).

Entonces se verifica que:

a) $R_{rl} = |A| + \text{Min} \{ k_1, |B| \}$, $J1 = J_r$, $J2 = D$

b) Si $l \in J_r$ $R_{rl}^1 = R_{rl}$

c) Si $l \in D$ $R_{rl}^1 = |A| + \text{Min} \{ k_1 - 1, |B| \}$

Demostración:

- a) Las variables con índices en el conjunto A no intervienen en la restricción. Su valor no está restringido y, por tanto, pueden tomar el valor uno (i.e. $1 \in J_1 \forall i \in A$). Las variables con índices en D no intervienen en la función objetivo. Entonces en el mejor de los casos tomarán valor cero (i.e. $1 \in J_2 \forall i \in D$). Las variables de B pertenecen a la función objetivo y a la restricción. Tomarán el valor uno el máximo número permitido:

$$\text{Min} \{ k_1, |B| \}.$$

Además, como k_1 y $|B|$ son siempre mayor o igual que uno, cualquier variable cuyo índice esté en B tomará valor uno en alguna solución óptima (i.e., $1 \in J_1 \forall i \in B$). Por tanto:

$$R_{r1} = |A| + \text{Min} \{ k_1, |B| \},$$

$$J_1 = A \cup B = J_r, \text{ y}$$

$$J_2 = D.$$

- b) Si $1 \in J_r \Rightarrow 1 \in J_1$. Entonces $x_1 = 1$ en alguna solución óptima del problema (C). Esa solución es también óptima para el problema (C_1) , y por tanto $R_{r1}^1 = R_{r1}$.

- c) Las variables con índices en el conjunto A no intervienen en la restricción. Su valor no está restringido y, por tanto, pueden tomar el valor uno. Las variables con índices en D no intervienen en la función objetivo. Como $1 \in D$, todas las variables con índices en D excepto x_1 deben tomar el valor cero. Las variables con índices en B pertenecen a la función objetivo y a la restricción. Por tanto tomarán el valor uno el máximo número permitido: $\text{Min} \{ k_1 - 1, |B| \}$, ya que la restricción es de la forma

$$\sum_{\substack{j \in J_1 \\ j \neq 1}} x_j \leq k_1 - 1.$$

$$\text{Así, } R_{r1}^1 = |A| + \text{Min} \{ k_1 - 1, |B| \} \quad \blacksquare$$

Corolario 5.6a: Se verifica que

a) $R_{ri}^l = R_{ri}^{l'}$ si $l, l' \in D$ ó $l, l' \notin D$.

b) $R_{ri}^l \leq R_{ri}^{l'}$ si $l \in D$ y $l' \notin D$.

Demostración:

El número de variables que toman valor uno en el máximo (y por tanto el valor de R_{ri}^l) es

$$|A| + \text{Min} \{ k_1 - 1, |B| \} \text{ si } l \in D, \text{ y}$$

$$|A| + \text{Min} \{ k_1, |B| \} \text{ si } l \notin D,$$

cantidades ambas independientes de l .

Además $\text{Min} \{ k_1 - 1, |B| \} \leq \text{Min} \{ k_1, |B| \}$ ■

El algoritmo de incremento de coeficientes es el siguiente:

Algoritmo 5.2:

Datos: Índices r e i , los conjuntos J_r y J_i , y el coeficiente entero k_1 .

Paso 1: Obtener A, B, D .

Paso 2: Sea $R_{ri}^D = |A| + \text{Min} \{ k_1 - 1, |B| \}$.

Si $R_{ri}^D \geq k_r$ STOP. No se puede incrementar ningún coeficiente.

Paso 3: Si $R_{ri}^D < k_r$ se obtiene una nueva restricción para cada $l \in D$, donde el nuevo coeficiente para x_l , con $l \in D$, será

$$a_l' = k_r - R_{ri}^D$$

Paso 4: Sea $R_{ri}^{AB} = |A| + \text{Min} \{ k_1, |B| \}$.

Si $R_{ri}^{AB} \geq k_r$ STOP. No se puede incrementar ningún coeficiente con índice en J_r .

Paso 5: Si $R_{r1}^{AB} < k_r$ se obtiene una nueva restricción para cada $l \in J_r$, tal que el nuevo coeficiente para x_l , con $l \in J_r$, será

$$a'_l = 1 + k_r - R_{r1}^{AB}$$

Observación 5.8: Cada nueva restricción con un coeficiente incrementado puede sustituir a la restricción original.

El algoritmo 5.2 puede mejorarse en caso de contar con un clique de potencial mayor que 1, según la siguiente proposición:

Proposición 5.7: Si el clique C elegido es tal que $\exists S \subseteq C$ con $|S| > 1$ y tal que $S \subseteq A \cup B$, entonces

$$R_{r1}^D = |A| - |S \cap A| + \delta + \text{Min} \{ k_i - 1, |B| - |S \cap B| + (1 - \delta) \} \text{ y}$$

$$R_{r1}^{AB} = |A| - |S \cap A| + \delta + \text{Min} \{ k_i, |B| - |S \cap B| + (1 - \delta) \},$$

$$\text{con } \delta = \begin{cases} 1 & \text{si } |\bar{S} \cap A| > 0 \\ 0 & \text{en caso contrario} \end{cases}$$

Demostración:

* $x(C) \leq 1 \Rightarrow x(S) \leq 1 \forall S \subseteq C$. Por tanto si $S \subseteq A \cup B$ se verifica $x(S \cap A) + x(S \cap B) \leq 1$.

* Si $|S \cap A| > 0$

Sólo $|A| - |\bar{S} \cap A| + 1$ variables con índices en A podrán tomar valor 1 en el óptimo de los problemas (5.9) y (5.10). En ambos casos, todas las variables con índices en $S \cap B$ deben tomar valor 0 y, por tanto, el número máximo de variables con índices en B que pueden tomar el valor 1 viene dado por la expresión

$$\text{Min} \{ k_i - 1, |B| - |S \cap B| \} \text{ para } R_{r1}^D$$

$$\text{Min} \{ k_i, |B| - |S \cap B| \} \text{ para } R_{r1}^{AB}.$$

* Si $|S \cap A| = 0$

Todas las variables con índices en A tomarán valor 1 en el máximo de ambos problemas. Sólo una variable con índice en $S \cap B$ puede tomar valor 1, y, por tanto, el número máximo de variables con índices en B que pueden tomar el valor 1 viene dado por la expresión

$$\text{Min} \{ k_i - 1, |B| - |S \cap B| + 1 \} \text{ para } R_{r1}^D$$

$$\text{Min} \{ k_i, |B| - |S \cap B| + 1 \} \text{ para } R_{r1}^{AB} \quad \blacksquare$$

Observación 5.9: Los coeficientes incrementados con índices en un clique pueden pertenecer simultáneamente a una nueva restricción que sustituya a la restricción original.

5.3.2.- Caso 2: Una restricción tipo mochila como soporte.

Problema: Incrementar los coeficientes de la restricción tipo cubrimiento $x(J_r) \leq k_r$ utilizando como soporte la restricción tipo mochila $\bar{x}(J_1) \leq k_1$, con $x(J_r) = \sum_{j \in J_r} x_j$, $\bar{x}(J_1) = \sum_{j \in J_1} a_j x_j$, donde $\exists j \in J_1$ tal que $a_j > 1$ y a_j entero $\forall a_j > 1, J_r \cap J_1 \neq \emptyset$.

Sea el siguiente problema para cada $i \in J$, donde $J = J_r \cup J_1$:

$$(CM_1) \quad R_{r1}^1 = \text{Max} \{ x(J_r) / \bar{x}(J_1) \leq k_j, x_1 = 1, x \in \{0,1\}^{|J|} \}, \quad (5.11)$$

y sea el problema general

$$(CM) \quad R_{r1} = \text{Max} \{ x(J_r) / \bar{x}(J_1) \leq k_j, x \in \{0,1\}^{|J|} \}. \quad (5.12)$$

Sea J_1 el conjunto de índices correspondientes a las variables que toman el valor 1 en una solución óptima de (CM), y $J_2 = J \setminus J_1$.

Como en el problema anterior, sea

$$B = J_r \cap J_1, \quad A = J \setminus J_1, \quad D = J \setminus J_r$$

Observación 5.10: Nótese que cualquier solución óptima de (CM) y de cada (CM_1) tendrá $x_i = 1 \forall i \in A$. Por tanto, si $|A| \geq k_r$ no se verifican las hipótesis de la proposición 5.3. En este caso el procedimiento descrito no podrá incrementar ningún coeficiente.

En caso contrario, si $|A| < k_r$, sea $J = J \setminus A$, $k_r = k_r - |A|$. Por tanto, sin pérdida de generalidad, se considerará que $A = \emptyset$.

Observación 5.11: Si existe un clique C de potencial mayor que 1 tal que $\exists S \subseteq C$ con $|S| > 1$ y $S \subseteq A$, entonces $k_r = k_r - (|A| - |A \cap S| + 1)$.

Sea $n = |J|$, $a = a_i$, y sea c un vector tal que $c_j = 1$ para $j \in J_r$, y $c_j = 0$ en caso contrario. La restricción tipo cubrimiento a reforzar será $cx \leq k_r$. El programa mochila a resolver para todo $l \in J$ será:

$$(CM_1) \quad R^l = \text{Max } \{cx \mid ax \leq k_l, x_i = 1, x \in \{0,1\}^n\} \quad (5.13)$$

Con objeto de reducir el número de iteraciones de proceso y con ello su complejidad computacional, es interesante considerar los índices en J de acuerdo con un orden no creciente en los coeficientes de la restricción tipo mochila considerada. Estableciendo inicialmente este orden, se puede anticipar que si en la iteración l ya no es posible incrementar el coeficiente correspondiente, entonces ningún coeficiente podrá ser incrementado en las iteraciones posteriores, para $j=1, \dots, n$ y $c_i = c_j$. (Ver proposición 5.9).

Sea una ordenación en el conjunto de índices J , tal que

$$c_{(1)} a_{(1)} \leq c_{(2)} a_{(2)} \leq \dots \leq c_{(n)} a_{(n)} \quad (5.14)$$

es decir, situamos primero los índices correspondientes a las variables que no forman parte del cubrimiento J_r (i.e. índices en D), y a su derecha situamos los índices de J_r . Además para $c_i = c_j$, si $a_i < a_j$, el índice i precederá a j en la nueva ordenación. De este modo se ordenan los coeficientes de la restricción tipo mochila, para los índices en D y, posteriormente, de manera independiente para los índices en J_r .

Para cada binomio compuesto por una restricción tipo cubrimiento y una restricción tipo mochila, se puede efectuar la ordenación (5.14) con el algoritmo Quicksort; su complejidad es $O(n \log n)$.

Sea A_i el valor de la suma de los i primeros términos de la secuencia $c_{(1)} a_{(1)}, c_{(2)} a_{(2)}, \dots, c_{(n)} a_{(n)}$; es decir:

$$A_i = A_{i-1} + c_{(i)} a_{(i)} \quad \text{para } i=1,2,\dots,n \quad (5.15)$$

$$A_0 = 0$$

El algoritmo para resolver el problema CM (5.12) una vez determinado el vector A es el siguiente:

Algoritmo 5.3:

Paso 1: Obtener el valor de α , donde $\alpha \in \{0, \dots, n\}$ es el único entero que cumple: $A_\alpha \leq k_1 < A_{\alpha+1}$

Paso 2: $R = \text{Max} \{x(J_r) / \bar{x}(J_1) \leq k_j, x \in \{0,1\}^n\} = \sum_{i=1}^{\alpha} c_{(i)}$

Paso 3: Calcular $\rho = \text{argmax} \{0, a_j \mid \forall j \in D\}$

Paso 4: $J_1 = \{ (j) : \rho < j \leq \alpha \}$

$$J_2 = J \setminus J_1$$

El programa mochila a resolver para cada índice l tal que $(l) \in J_2$ es el siguiente:

$$\begin{aligned}
 (CM)_l \quad R^{(l)} &= c_{(l)} + \max \sum_{i \in J \setminus \{(l)\}} c_i x_i \\
 \text{s.a.} \quad & \sum_{i \in J \setminus \{(l)\}} a_i x_i \leq k_1 - a_{(l)} \\
 & x_i \in \{0, 1\} \quad \forall i \in J \setminus \{(l)\}
 \end{aligned} \quad (5.16)$$

Observación 5.12: $R^{(l)} = R$ para $(l) \in J2$ tal que $c_{(l)} a_{(l)} = c_{(f)} a_{(f)}$ para $(f) \in J1$.

El algoritmo para resolver el programa mochila (5.16) para el índice l , una vez determinado el vector A , donde A_l es el valor definido en (5.15), para $i = 1, 2, \dots, n$, es el siguiente:

Algoritmo 5.4:

Paso 1: Obtener el valor de α , donde $\alpha \in \{1, \dots, n\} \setminus \{l\}$ es el único entero para el que se cumple:

$$A_{-\alpha}^1 \leq k_1 - a_{(l)} < \bar{A}_{\alpha}^1, \text{ donde}$$

$$A_{-\alpha}^1 = \begin{cases} A_{\alpha} - c_{(l)} a_{(l)} & \text{para } l < \alpha \\ A_{\alpha} & \text{en otro caso} \end{cases} \quad (5.17)$$

y

$$\bar{A}_{\alpha}^1 = \begin{cases} A_{\alpha+1} - c_{(l)} a_{(l)} & \text{para } l < \alpha \\ A_{\alpha+2} - c_{(l)} a_{(l)} & \text{para } l = \alpha+1 \\ A_{\alpha+1} & \text{en otro caso.} \end{cases} \quad (5.18)$$

Paso 2: Calcular $R^{(l)}$, tal que

$$R^{(l)} = \begin{cases} \sum_{i=1}^{\alpha} c_{(i)} & \text{si } l < \alpha \\ \sum_{i=1}^{\alpha} c_{(i)} + c_{(l)} & \text{en otro caso.} \end{cases} \quad (5.19)$$

Paso 3: $\bar{c}_{(l)} = c_{(l)} + \max \{0, k_1 - R^{(l)}\}$

Observación 5.13: Puesto que sólo es necesario resolver el problema mochila (5.16) para cada índice l tal que $(l) \in J2$ y no verifica la condición dada en la observación 5.12, el algoritmo puede simplificarse de la siguiente manera:

Paso 1: Obtener el valor de α , donde $\alpha \in \{1, \dots, n\} \setminus \{1\}$ es el único entero para el que se cumple:

$$A_{-\alpha}^1 \leq k_1 - a_{(1)} < \bar{A}_{\alpha}^1, \text{ donde } A_{-\alpha}^1 = A_{\alpha} \text{ y}$$

$$\bar{A}_{\alpha}^1 = \begin{cases} A_{\alpha+2} - c_{(1)} a_{(1)} & \text{para } l = \alpha+1 \\ A_{\alpha+1} & \text{en otro caso.} \end{cases}$$

puesto que $1 < \alpha$ y $l \in J_2$ implica $l < \rho$.

Proposición 5.8: La complejidad del algoritmo 5.4 para resolver el programa mochila (5.16) es $O(\log n)$.

Demostración:

Determinar el valor de α y con ello el valor de $R^{(1)}$, se puede realizar mediante una búsqueda binaria similar a la utilizada en el algoritmo Quicksort y, por tanto, con complejidad $O(\log n)$. ■

Observación 5.14: El algoritmo 5.4 obtiene el coeficiente reformado \bar{c}_j que reemplaza al coeficiente c_j en la condición cubrimiento $cx \leq k_r$.

Dado el ordenamiento (5.14), y comenzando las iteraciones de incremento en orden no creciente, se puede anticipar que si en la iteración j ya no es posible incrementar el coeficiente correspondiente $c_{(l)}$ con $l = n+1-j$ ningún coeficiente podrá ser incrementado en las iteraciones posteriores $q = j+1, \dots, n$ para $c_{(l)} = c_{(n+1-q)}$. (Ver proposición 5.9). Por tanto, se efectúa el proceso de incremento en dos etapas, primero para aquellas variables cuyos índices están en J_r y luego para las correspondientes a los índices en D . En ambos casos, durante el proceso, el índice de la variable que es fijada a 1, se mueve en la dirección $(n), (n-1), \dots, (\rho+1)$ y $(\rho), (\rho-1), \dots, (1)$, donde ρ es tal que $\rho = \operatorname{argmax} \{0, a_j \mid j \in D\}$.

Proposición 5.9: * Sean $(l) = (n+1-j)$ y $(p) = (n+1-q)$ los índices correspondientes a las variables que se fuerzan a tomar el valor 1 en las iteraciones j y q respectivamente, y sea $a_{(p)} \leq a_{(l)}$, $c_{(p)} = c_{(l)}$ para $q > j$.

* Sean $R^{(l)}$ y $R^{(p)}$ los valores óptimos de la función objetivo de los programas mochila (5.16) correspondientes a las respectivas iteraciones j y q .

Entonces, resulta que $R^{(p)} \geq R^{(l)}$.

Demostración:

Para la iteración j ,

$$R^{(l)} = c_{(l)} + \max \sum_{i \in J \setminus \{(l)\}} c_i x_i$$

s.a.

$$\sum_{i \in J \setminus \{(l)\}} a_i x_i \leq k_l - a_{(l)}$$

$$x_i \in \{0,1\} \quad \forall i \in J \setminus \{(l)\}$$

(CM_l)

Para la iteración q ,

$$R^{(p)} = c_{(p)} + \max \sum_{i \in J \setminus \{(p)\}} c_i x_i$$

s.a.

$$\sum_{i \in J \setminus \{(p)\}} a_i x_i \leq k_q - a_{(p)}$$

$$x_i \in \{0,1\} \quad \forall i \in J \setminus \{(p)\}$$

(CM_p)

(5.20)

Los posibles casos son los siguientes:

1. $x_{(p)} = 1$ en la solución óptima de (CM_l) (5.20).

a. $x_{(l)} = 1$ en la solución óptima de (CM_p) (5.20).

Todas las variables que en (CM_p) toman el valor uno también lo pueden tomar en (CM_l), puesto que tienen la misma holgura en la restricción tipo mochila y el mismo coeficiente en la función objetivo para (CM_p) y (CM_l). Por tanto $R^{(p)} = R^{(l)}$.

b. $x_{(l)} = 0$ en la solución óptima de (CM_p).

Puesto que $x_{(p)} = 1$ en el óptimo de (CM_l) entonces $\exists f \geq p$

tal que $x_{(i)} = 1 \quad \forall i=1, \dots, f$ en el óptimo de (CM_p) . Puesto que $x_{(i)} = 0$ en el óptimo de (CM_p) entonces $f < 1$, y además es posible que haya un conjunto de variables, sea $\{i\}$ para $a_{(f)} \leq a_{(i)} \leq a_{(1)}$, que tomen el valor 1. Por tanto, $R^{(p)} \geq R^{(1)}$ ya que, en caso contrario, $x_{(i)}$ no hubiera tomado el valor cero en la solución óptima para (CM_p) .

2. $x_{(p)} = 0$ en la solución óptima para (CM_1) .

a. $x_{(i)} = 1$ en la solución óptima de (CM_p) . En este caso $R^{(p)} = R^{(1)}$ o la situación es **imposible**, ya que entonces la solución óptima de (CM_p) es factible para (CM_1) y mejor, en contradicción con que $x_{(p)} = 0$ en el óptimo de (CM_p) .

b. $x_{(i)} = 0$ en la solución óptima de (CM_p) .

En este caso, $\exists f < p$ tal que $x_{(i)} = 1 \quad \forall i = 1, \dots, f$ en el óptimo de (CM_1) . Estas variables también toman el valor 1 en el la solución óptima de (CM_p) .

Además, como $k_1 - a_{(p)} \geq k_1 - a_{(i)}$, al menos una de las variables i , con $a_{(f)} \leq a_{(i)} \leq a_{(1)}$, tomará el valor 1. Por tanto $R^{(p)} \geq R^{(1)}$.

Por tanto, el número de variables con valor 1 en la solución óptima del problema mochila asociado a la iteración q , nunca será inferior que el número correspondiente para el problema asociado a la iteración j . ■

Corolario 5.8a: Si $R^{(1)} \geq k_r$, resulta que $R^{(p)} \geq k_r$ en la hipótesis contemplada en la proposición 5.6.

Demostración:

Trivial. ■

La descripción del algoritmo de incremento de coeficientes de la restricción tipo cubrimiento $cx \leq k_r$ dada la restricción tipo mochila $ax \leq k_l$ es la siguiente:

Algoritmo 5.5:

Paso 1: Ordenar los coeficientes $c_j a_j$ de modo que:

$$c_{(1)} a_{(1)} \leq c_{(2)} a_{(2)} \leq \dots \leq c_{(n)} a_{(n)}$$

teniendo en cuenta que (i) precede a (j) si $a_{(i)} < a_{(j)}$ y

$$c_{(i)} = c_{(j)} = 0.$$

Paso 2: Calcular ρ tal que:

$$\rho = \operatorname{argmax} \{0, a_j \mid j \in D\}$$

Paso 3: Calcular el vector A, donde

$$A_i = A_{i-1} + c_{(i)} a_{(i)} \quad i=1,2,\dots,n \quad \text{y} \quad A_0 = 0$$

Paso 4: Obtener R, J1 y J2 según el algoritmo 5.3.

Paso 5: Para cada iteración $j = 1, 2, \dots, n - \rho$, mientras $R^{(j)} < k_r$ donde $l = n + 1 - j$ hacer: si $(l) \in J1$ ó $c_{(l)} a_{(l)} = c_{(f)} a_{(f)}$ para $(f) \in J1$, entonces $R^{(j)} = R$; de lo contrario (i.e., $l \in J2$ y $\exists (f) \in J1$ tal que $c_{(l)} a_{(l)} = c_{(f)} a_{(f)}$), ejecutar el algoritmo 5.4.

Paso 6: Para cada iteración $j = n - \rho + 1, n - \rho + 2, \dots, n$, mientras $R^{(j)} < k_r$ donde $l = n + 1 - j$, ejecutar el algoritmo 5.4.

Proposición 5.10: La complejidad del algoritmo 5.5 para obtener los

coeficientes $c_j \forall j \in J$ es $O(n \log n)$.

Demostración:

El paso 1 ejecuta el algoritmo Quicksort y, por tanto, tiene complejidad $O(n \log n)$; el paso 2 tiene complejidad $O(\log n)$, el paso 3 tiene complejidad $O(n)$ y los pasos 4 y 5 conjuntamente tienen complejidad $O(n \log n)$ pues el algoritmo 4.4 se ejecuta n veces en el peor de los casos. ■

Observación 5.15: $(l) \in J_2$ para todo $l = n+1-j$ donde $j = n-p+1, n-p+2, \dots, n$. Ver paso 4 del algoritmo 4.3. Por tanto el Paso 6 queda simplificado al no necesitar analizar si $(l) \in J_1$.

Observación 5.16: Cada ejecución del algoritmo 5.4 en los pasos 4 y 5 para la iteración j actualiza el coeficiente $c_{(l)}$ para $l = n+1-j$.

Observación 5.17: Si $(l) \in J_1$ ó $c_{(l)} a_{(l)} = c_{(f)} a_{(f)}$ para $(f) \in J_1$ entonces $R^{(l)} = R$ y $\bar{c}_{(l)} = c_{(l)} + \max \{ 0, k_i - R \}$.

Observación 5.18: El incremento de coeficientes logrado por el algoritmo 5.5 puede mejorarse en caso de contar con un clique C de potencial mayor que 1, tal que $\exists S \subseteq C$ con $|S| > 1$ y $S \subseteq J_r$. En ese caso se efectúa la ordenación (5.14) teniendo sólo en cuenta la variable x_j , tal que $a_j = \min \{ a_i / i \in S \}$, de forma que el resto de las variables con índices en S no aparezcan en la ordenación. De esta forma se realizan las iteraciones correspondientes a todas las variables, excepto las correspondientes a índices $i \in S$ con $i \neq j$. Para estos índices se sustituye la variable x_j por la variable x_i y se comienza de nuevo el algoritmo.

Observación 5.19: Cada clique C en J puede producir una condición (que sustituye a la restricción SOP dada $cx \leq k_r$), en la que se pueden incrementar, simultáneamente, todos los coeficientes correspondientes a variables que formen parte del clique y que individualmente puedan ser incrementados.

5.4.- REFORZAR RESTRICCIONES TIPO MOCHILA.

5.4.1.- Caso 1: Una restricción tipo cubrimiento como soporte.

Problema: Incrementar los coeficientes de la restricción tipo mochila $\bar{x}(J_r) \leq k_r$ utilizando como soporte la restricción tipo cubrimiento

$x(J_1) \leq k_1$, con $\bar{x}(J_r) = \sum_{j \in J_r} a_j x_j$, $x(J_1) = \sum_{j \in J_1} x_j$, donde $a_j \geq 1$ y entero

$\forall j \in J_r, \exists j \in J_1$ tal que $a_j > 1, J_r \cap J_1 \neq \emptyset$.

Sea el siguiente problema para cada $l \in J$, donde $J = J_r \cup J_1$:

$$(MC_l) \quad R_{r1}^l = \text{Max} \{ \bar{x}(J_r) / x(J_1) \leq k_1, x_1 = 1, x \in \{0,1\}^{|J|} \}, \quad (5.21)$$

y sea el problema general

$$(MC) \quad R_{r1} = \text{Max} \{ \bar{x}(J_r) / x(J_1) \leq k_1, x \in \{0,1\}^{|J|} \}. \quad (5.22)$$

Sea J_1 el conjunto de índices correspondientes a las variables que toman el valor 1 en una solución óptima de (MC), y $J_2 = J \setminus J_1$. Sean los conjuntos

$$B = J_r \cap J_1, \quad A = J \setminus J_1, \quad D = J \setminus J_r$$

Observación 5.20: Nótese que cualquier solución óptima de (MC) y de cada (MC_l) tendrá $x_1 = 1 \forall i \in A$. Por tanto, si $\sum_{j \in A} a_j \geq k_r$ no se verifican las hipótesis de la proposición 5.3. En este caso el procedimiento descrito no podrá incrementar ningún coeficiente.

En caso contrario, sea $J = J \setminus A$, $k_r = k_r - \sum_{j \in A} a_j$. Por tanto, sin pérdida de generalidad, se considerará que $A = \emptyset$.

Observación 5.21: Si existe un clique C de potencial mayor que 1, tal que

$\exists S \subseteq C$, con $|S| > 1$ y $S \subseteq A$, entonces $k_r = k_r - \sum_{j \in A \setminus S} a_j - \max \{ a_j, j \in S \}$,
 $J = J \setminus A$.

Sea $n = |J|$, $c = c$, y sea a un vector de componente i -ésima a_i si $i \in J_r$, y cero en caso contrario. La restricción tipo mochila a reforzar será $ax \leq k_r$. El programa mochila a resolver para todo $i \in J$ será:

$$(MC_i) \quad R^i = \text{Max} \{ ax / cx \leq k_i, x_i = 1, x \in (0,1)^n \}$$

Sea una ordenación en el conjunto de índices J , tal que

$$c_{(1)} a_{(1)} \geq c_{(2)} a_{(2)} \geq \dots \geq c_{(n)} a_{(n)} \quad (5.23)$$

es decir, situamos primero los índices correspondientes a las variables en J_r en orden no creciente de sus coeficientes a_i y, a su derecha, situamos (en cualquier orden) los índices en D .

La descripción del algoritmo de incremento de coeficientes de la restricción tipo mochila $ax \leq k_r$ dada la restricción tipo cubrimiento $cx \leq k_i$ es la siguiente:

Algoritmo 5.6:

Paso 1: Ordenar los coeficientes $c_j a_j$ de modo que

$$c_{(1)} a_{(1)} \geq c_{(2)} a_{(2)} \geq \dots \geq c_{(n)} a_{(n)}$$

Paso 2: Sea $R_{r1}^- = \sum_{j=1}^{k_r-1} a_{(j)}$. Si $R_{r1}^- \geq k_r$ PARAR. No se puede incrementar ningún coeficiente.

Paso 3: Sea $c_{n+1} = \infty$. Calcular $\rho = \text{argmin} \{ c_j \forall j \in D \cup \{n+1\} \}$

Paso 4: $J1 = \begin{cases} (j) : 1 \leq j < \rho & \text{si } \rho \leq k_i \\ (j) : 1 \leq j \leq k_i & \text{en otro caso} \end{cases}$

$$J2 = J \setminus J1$$

Paso 5: $R_{rl} = \sum_{j=1}^{k_1} a_{(l)}$

Paso 6: Para cada iteración $j = 1, \dots, n-\rho+1$, donde $l = n+1-j$ hacer:

$$R_{rl}^{(j)} = R_{rl}^-$$

Paso 7: Para cada iteración $j = n-\rho+2, \dots, n$, mientras $R^{(j)} < k_r$ donde $l = n+1-j$ hacer:

* si $(l) \in J_1$ ó $c_{(l)} a_{(l)} = c_{(f)} a_{(f)}$ para $(f) \in J_1$ entonces $R_{rl}^{(j)} = R_{rl}$

* en caso contrario (i.e., $(l) \in J_2$ y $\exists (f) \in J_1$ t.q. $c_{(l)} a_{(l)} = c_{(f)} a_{(f)}$)

$$\text{entonces } R_{rl}^{(j)} = a_{(l)} + \sum_{j=1}^{k_1-1} a_{(l)}$$

Observación 5.22: Cada iteración j del algoritmo 5.6 actualiza el coeficiente $a_{(l)}$ para $l = n+1-j$, de forma que

$$\bar{a}_{(l)} = a_{(l)} + \max \{ 0, k_r - R^{(j)} \}$$

Observación 5.23: Se aplica la observación 5.18 del apartado 5.2.2, tal que en cada iteración j correspondiente a un índice l perteneciente al clique C se reordenan los índices de forma que no aparezcan en la ordenación los índices $i \in C$, $i \neq l$.

5.4.2.- Caso 2: Otra restricción tipo mochila como soporte.

Sea el vector x de variables 0-1, y sean las condiciones reforzadas

$$a_r x \leq b_r \tag{5.24}$$

y

$$a_i x \leq b_i \tag{5.25}$$

donde $a_{rj} \geq 0$ y $a_{ij} \geq 0 \forall j \in J$ y enteros, son los elementos de los vectores

a_r y a_l , respectivamente, tal que $\exists g \in J_r$ con $a_{rg} > 1$, $\exists f \in J_l$ con $a_{lf} > 1$, y $b_r > 0$ y $b_l > 0$.

Se propone a continuación un método de incremento de los coeficientes de la restricción tipo mochila (5.24) de manera que la condición resultante sea equivalente a la inicial y más fuerte.

Para cada variable x_l , $l \in J$, se define el siguiente programa 0-1:

$$\begin{aligned}
 (MM_{r,l}^1): \quad R_{r,l}^1 &= a_{r,l} + \max \sum_{j \in J \setminus \{l\}} a_{rj} x_j \\
 \text{s.a.} \quad & \sum_{j \in J \setminus \{l\}} a_{lj} x_j \leq b_l - a_{ll} \\
 & x_l = 1, x_j \in \{0,1\} \quad \forall j \neq l
 \end{aligned} \tag{5.26}$$

Es decir, $R_{r,l}^1$ es el máximo valor que puede tomar el término de la izquierda de (5.24), si x es una solución factible para la restricción (5.25), tal que $x_l = 1$.

Por tanto, la nueva restricción (reforzada) será

$$\bar{a}_r x \leq b \tag{5.27}$$

donde los elementos $\bar{a}_{rj} \quad \forall j \in J$ del vector \bar{a}_r tienen la expresión:

$$\bar{a}_{rj} = \begin{cases} a_{rj} & \text{si } j \neq l \\ a_{rj} + \max \{0, b_r - R_{r,l}^j\} & \text{si } j = l \end{cases} \tag{5.28}$$

Observación 5.24: Puede ocurrir que $a_{r,l} = 0$ y $\bar{a}_{r,l} \geq 1$.

Observación 5.25: Dado un clique $C \subset J$ tal que $l \in C$, el incremento se producirá no sólo sobre el coeficiente de la variable x_l , sino sobre coeficientes correspondientes a variables en el clique C .

Dado un clique C , para cada variable $x_i, i \in C$, se define el siguiente programa 0-1:

$$\begin{aligned}
 (MM_{r,C,i}^1): \quad & R_{r,C,i}^1 = a_{ri} + \max \sum_{j \in J \setminus \{i\}} a_{rj} x_j \\
 \text{s.a.} \quad & \\
 & \sum_{j \in J \setminus \{i\}} a_{rj} x_j \leq b_i - a_{ri} \\
 & x_i = 1, \quad x_j = 0 \quad \forall j \in C \setminus \{i\} \\
 & x_j \in \{0,1\} \quad \forall j \in J \setminus C
 \end{aligned} \tag{5.29}$$

Por tanto, la nueva restricción (reforzada) será

$$\bar{a}_r x \leq b \tag{5.30}$$

donde los elementos $\bar{a}_{rj} \forall j \in J$ del vector \bar{a}_r tienen la expresión:

$$\bar{a}_{rj} = \begin{cases} a_{rj} & \text{si } j \notin C \\ a_{rj} + \max \{0, b_r - R_{r,C,i}^j\} & \text{si } j \in C \end{cases} \tag{5.31}$$

Observación 5.26: Para unas restricciones dadas r e i , y para un clique C , el programa (5.29) se repite $|C|$ veces con pequeñas modificaciones. De hecho, $(MM_{r,C,i}^p)$ y $(MM_{r,C,i}^q)$ únicamente se diferencian en que se impone $x_p = 1$ para $(MM_{r,C,i}^p)$ y $x_q = 1$ para $(MM_{r,C,i}^q)$ para $p, q \in C$.

Observación 5.27: Para la resolución del problema (5.26) y (5.29) se puede utilizar el algoritmo para optimización del problema de la mochila descrito en Martello y Toth (1984), (1990).

5.4.3.- Caso 3: Misma restricción como soporte

Sea el siguiente problema para el índice $p \in C$, como un caso particular del problema (5.29),

$$(S_p): R^p = \max \{ ax / ax \leq b, x_p = 1, x \in \{0,1\}^n \} \quad (5.32)$$

Una vez obtenido R^p , no siempre es necesario resolver un nuevo problema para obtener R^q , con $p \neq q$. Sea la ordenación no decreciente de los coeficientes del vector a . Entonces es posible determinar R^q bajo ciertas circunstancias sin resolver el nuevo problema, para $q > p$, una vez obtenido R^p . (Ver proposiciones 5.11 y 5.12).

Observación 5.28: $p < q$ si $a_p \leq a_q$.

Proposición 5.11: Sea R^p el valor óptimo de (S_p) (5.32). Se trata de determinar el valor de R^q para $q > p$, con $q \notin C_p, p \in C_p, x(C_p) \leq 1$.

Entonces:

Caso 1: $x_q^* = 0$ en la solución óptima x^{*p} del problema (S_p) , con valor óptimo R^p . Si además $0 \leq \delta \equiv a_q - a_p \leq b - R^p$, entonces $R^q = R^p + \delta$.

Caso 2: $x_q^* = 1$ en la solución óptima x^{*p} de (S_p) , con valor óptimo R^p . Entonces $R^q \geq R^p$.

Demostración:

Caso 1: $x_q^* = 0$ en el óptimo x^{*p} para (S_p) .

Dada una solución factible x^{ap} para el problema (S_p) , sea $N_p^a \subset J$ el conjunto de índices tales que $i \in N_p^a$ si y sólo si $x_i^{ap} = 1$ en x^{ap} . Entonces, $p \in N_p^*$, $q \notin N_p^*$, donde N_p^* es el conjunto de índices correspondientes a las variables que toman el valor uno en la solución óptima para (S_p) .

a) $x_p^* = 0$ en el óptimo para (S_q) .

Entonces $p \notin N_q^*$, $q \in N_q^*$. Como $0 \leq \delta \leq b - R^p$ por hipótesis,

$N_q^a = N_p^* - \{p\} + \{q\}$ es tal que la solución x^a con

$$\left. \begin{array}{l} x_i^a = 1 \quad \forall i \in N_q^a \\ x_i^a = 0 \quad \forall i \notin N_q^a \end{array} \right\} \text{ es factible para el problema } (S_q).$$

Veamos que esta solución es óptima para (S_q) . Su valor en la función objetivo es ax^a , con $x_q^a = 1$.

Si no es óptima es porque $\exists N'_q \neq N_q^a$ con $p \notin N'_q$, $q \in N'_q$ que proporciona una solución factible, tal que $ax' > ax^a$.

En este caso, $N'_q + \{p\} - \{q\}$ proporciona una solución factible para el problema (S_p) ($a_p \leq a_q$), y con mejor valor en la función objetivo que el proporcionado por N_p^* , en contradicción con que N_p^* proporcionaba el valor óptimo.

Por tanto la solución dada por N_q^a es óptima, $N_q^a = N_q^*$, y

$$R^q = R^p - a_p + a_q = R^p + \delta \quad \text{c.q.d.}$$

b) $x_p^* = 1$ en el óptimo x^{*q} para (S_q) .

Esto significa que x^{*q} es una solución factible para (S_p) , y, por tanto, si es óptima para un problema también lo es para el otro. Pero eso contradice la hipótesis de que $x_q^* = 0$ en el óptimo para (S_p) . Por tanto es **imposible**.

Caso 2: $x_q^* = 1$ en el óptimo para (S_p) .

a) $x_p^* = 1$ en el óptimo x^{*q} para (S_q) .

- Si $R^p < R^q$ entonces la solución x^{*p} no es óptima para (S_p) , ya que la solución óptima x^{*q} para (S_q) es factible para (S_p) y mejor.

- Si $R^p > R^q$ entonces la solución x^{*q} para (S_q) no es óptima por una razón análoga.

Por tanto $R^p = R^q$

b) $x_p^* = 0$ en el óptimo x^{*q} para (S_q) .

- Si $R^q < R^p$ entonces la solución x^{*q} no es óptima para (S_q) , ya que la solución óptima para (S_p) es factible para (S_q) y mejor.

Por tanto $R^q \geq R^p$. ■

Corolario 5.10a: Supuestas las hipótesis de la proposición 5.11, si $R^p = b$ entonces no es necesario resolver el problema para ningún q tal que $x_q^* = 1$ en el óptimo para (S_p) .

Demostración:

Estamos en las hipótesis del caso 2 de la proposición 5.11.

Por tanto $R^q \geq R^p = b$. Como el problema es un SSP, $R^q \leq b \forall q \Rightarrow R^q = b$ y por tanto no se podrá incrementar su coeficiente a_q . Así, no es necesario resolver explícitamente el problema. ■

Proposición 5.12: Sea R^p el valor óptimo de (S_p) (5.32). Se trata de determinar el valor de R^q para $q > p$, con $p, q \in C$, $x(C) \leq 1$, y $0 \leq \delta \equiv a_q - a_p \leq b - R^p$.

Entonces $R^q = R^p + \delta$.

Demostración:

Dado que $p, q \in C$ entonces:

En el óptimo para (S_p) $x_p^* = 1 \Rightarrow x_q^* = 0$ }
 En el óptimo para (S_q) $x_q^* = 1 \Rightarrow x_p^* = 0$ }

Se verifican por tanto las hipótesis del apartado a) del caso 1 de la proposición 5.11 y, por tanto, $R^q = R^p + \delta$. ■

Observación 5.29: Sea $J \subset J$ el conjunto de índices $\{i\}$ tal que $x_i = 1$ en una solución óptima para el problema

$$(S): R = \text{Max} \{ ax \mid ax \leq b, x \in \{0,1\}^n \} \quad (5.33)$$

Se verifica entonces que $R_p = R \quad \forall p \in J$.

Observación 5.30: Para la resolución del problema (5.32) y (5.33) se puede utilizar el algoritmo para optimización del problema SSP descrito en Martello y Toth (1988), (1990).

5.5.- IDENTIFICACION DE CLIQUES MAXIMALES.

Problema: Dada una condición tipo mochila

$$\sum_{j \in J} a_j x_j \leq b \quad (5.34)$$

identificar todos los cliques maximales inducidos por ella.

Sea una ordenación en el conjunto de índices J , con $n = |J|$, tal que

$$a_{(1)} \leq a_{(2)} \leq \dots \leq a_{(n)} \leq b \quad (5.35)$$

Para cada índice $j \in \{1, \dots, n\}$ se define

$$\gamma(j) = \begin{cases} \min \{ k \mid 1 \leq k \leq n, a_j + a_k > b, \text{ con } j \neq k \} \\ n+1, \text{ en otro caso} \end{cases} \quad (5.36)$$

Obviamente si $\gamma(j) = n+1$, la variable x_j no pertenece a ningún clique no trivial implicado por la condición (5.34). En otro caso, (i.e., $\gamma(j) \neq n+1$), el conjunto $C(j) = \{j\} \cup \{k \mid k \geq \gamma(j)\}$ define un clique.

Proposición 5.13: Sean $i, j \in \{1, \dots, n\}$ dos índices tales que $i < j$ y $\gamma(j) \neq n+1$. Entonces

(a) $\gamma(i) \geq \gamma(j)$

(b) $\gamma(i) = \gamma(j)$ si y sólo si $a_j + a_{\gamma(i)-1} \leq b$.

Demostración:

Ver Dietrich, Escudero, Garín y Pérez (1993).

A partir de este resultado es posible obtener todos los cliques maximales inducidos por la condición (5.34) sin repeticiones.

Algoritmo 5.7:

Paso 1: Ordenar los coeficientes de la condición mochila en orden no decreciente, de modo que

$$a_{(1)} \leq a_{(2)} \leq \dots \leq a_{(n)} \leq b$$

Paso 2: Calcular $\gamma(n)$ según la expresión (5.36).

Sea $h = \min \{ n, \gamma(n) \}$

Si $h = n$ PARAR. No existe ningún clique no trivial inducido por la condición mochila dada.

Paso 3: Sea $g = \min \{ k / h \leq k \leq n, a_k + a_h > b \}$

Paso 4: Se define el clique $C(h) = \{h\} \cup \{ k / k \geq g \}$

Paso 5: Si $g = h+1$ PARAR. Se han identificado todos los cliques maximales.

En caso contrario hacer $h = h+1$.

Paso 6: Si $g = h+1$ ó $a_h + a_{g-1} \leq b$ ir al paso 4.

Paso 7: Hacer $g = g-1$ e ir al paso 6.

5.6.- EJEMPLO ILUSTRATIVO.

Sea el grafo dirigido $G = (V,A)$ de la figura 5.1, y sea el digrafo de precedencias $P = (V,\Pi)$ tal que $\{(5,6),(6,7),(7,8),(7,4),(4,2),(10,8)\} \subseteq \Pi$ y Π es cerrado transitivamente (ver figura 5.2). El nudo 1 hace las veces de nudo auxiliar, y, por tanto, $(1,v) \in \Pi \forall v \in V$.

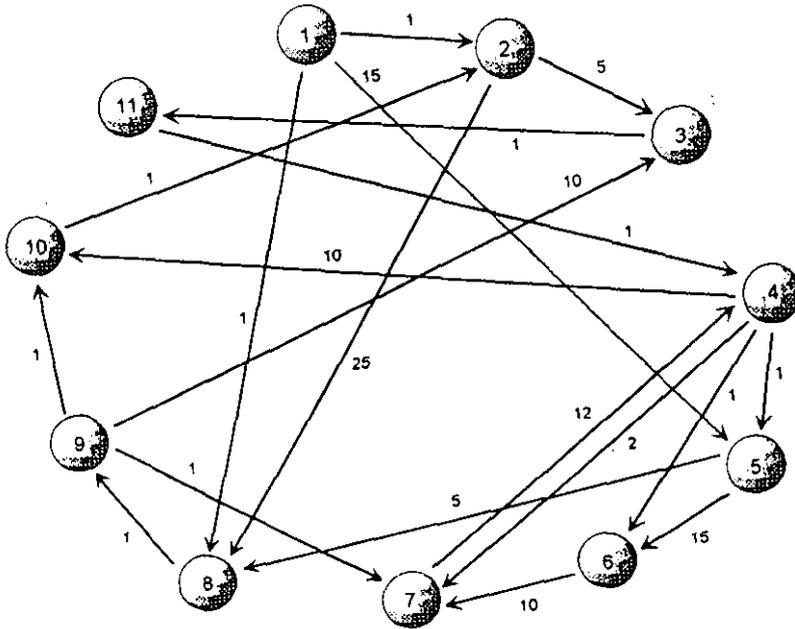


Figura 5.1

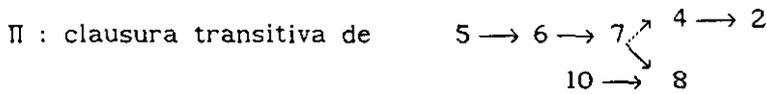
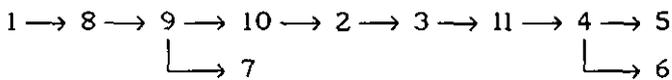


Figura 5.2

La arborescencia de mínimo peso en el grafo G es:



La precedencia (5,8) es violada por esta solución. Se obtiene así el corte de tipo I:

$$x_{89} + x_{910} + x_{102} + x_{23} + x_{311} + x_{114} + x_{45} + x_{46} + x_{47} \leq 6 \quad (5.37)$$

Se obtiene así el corte reforzado tipo mochila (incrementando coeficientes en las variables del clique (5.41), según la observación 5.6)

$$2x_{89} + 2x_{910} + x_{102} + x_{23} + x_{311} + x_{114} + x_{45} + x_{46} + x_{47} \leq 6 \quad (5.42)$$

El corte (5.38), de tipo cubrimiento, no se podía reforzar utilizando como soporte el cubrimiento (5.37). En cambio, si utilizamos como soporte el corte (5.42), reforzamiento del corte (5.37), que es una condición mochila, y se aplica el algoritmo 5.5,

Paso 0: $|A| = 1 \Rightarrow k_r = k_r - 1 = 6$

Paso 1: Se ordenan los coeficientes, de forma que

$$\begin{array}{lllll} (1) = 102 & (2) = 23 & (3) = 311 & (4) = 114 & (5) = 45 \\ (6) = 46 & (7) = 47 & (8) = 89 & (9) = 910 & \end{array}$$

Paso 2: $\rho = 0$

$$\begin{array}{lllll} \text{Paso 3: } A_0 = 0 & A_1 = 1 & A_2 = 2 & A_3 = 3 & A_4 = 4 \\ & A_5 = 5 & A_6 = 6 & A_7 = 7 & A_8 = 9 & A_9 = 11 \end{array}$$

Paso 4: $A_6 \leq k_1 < A_7 \Rightarrow \alpha = 6$. Según el algoritmo 5.3 se obtiene:

$$R = 6$$

$$J1 = \{ (1), (2), (3), (4), (5), (6) \}$$

$$J2 = \{ (7), (8), (9) \}$$

Paso 5: Aplicando el algoritmo 5.4 para $l = (9)$ y $l = (8)$ se obtiene

$$\alpha = 4$$

$$R^{(9)} = \sum_{i=1}^{\alpha} c_{(i)} + c_{(9)} = 4 + 1 = 5 < 6 = k_r$$

$$R^{(8)} = 5 < 6$$

$$\text{Por tanto } a'_{(9)} = c_{(9)} + (6 - 5) = 2, \quad a'_{(8)} = c_{(8)} + (6 - 5) = 2$$

Por otra parte,

$$R^{(7)} = R^{(6)} = R^{(5)} = R^{(4)} = R^{(3)} = R^{(2)} = R^{(1)} = 6 \geq k_r$$

Obteniéndose el corte (tipo mochila) reforzado siguiente (incrementando

coeficientes en las variables del clique (5.41))

$$x_{18} + 2x_{89} + 2x_{910} + x_{102} + x_{23} + x_{311} + x_{114} + x_{45} + x_{46} + x_{47} \leq 7 \quad (5.43)$$

Intentemos ahora reforzar la condición mochila (5.43) utilizando el cubrimiento (5.40) como soporte (algoritmo 5.6).

Paso 0: $A = \{ 18, 89, 910, 102 \}$. $\sum_{j \in A} a_j = 6 < 7 = k_r$. Por tanto

$$J = J \setminus A, \quad k_r = k_r - 6 = 1$$

Paso 1: Se ordenan los coeficientes, de forma que

$$\begin{aligned} (1) &= 23 & (2) &= 311 & (3) &= 114 & (4) &= 45 \\ (5) &= 46 & (6) &= 47 \end{aligned}$$

Paso 2: $R_{ri}^- = 1 \geq k_r$.

Por tanto no se puede incrementar ningún coeficiente.

Sin embargo, utilizando la restricción tipo clique (5.41) según la observación 5.21, el paso 0 pasa a ser

Paso 0: $J = J \setminus A$, $k_r = k_r - 2 - 2 = 3$

Y por tanto $R_{ri}^- = 1 < k_r$

Así, aplicando el algoritmo, se obtiene la siguiente condición reforzada,

$$x_{18} + 3x_{89} + 3x_{910} + x_{102} + x_{23} + x_{311} + x_{114} + x_{45} + x_{46} + x_{47} \leq 7 \quad (5.44)$$

que sustituye a la condición tipo mochila (5.43).

Estimando los multiplicadores de Lagrange de todos estos cortes y calculando la solución del nuevo problema relajado se obtiene la arborescencia

$$1 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 4 \rightarrow 10 \rightarrow 2 \rightarrow 8 \rightarrow 9 \rightarrow 3 \rightarrow 11$$

que no viola ninguna restricción y es, por tanto, la solución óptima del problema.

Apéndice 1

Descomposición de Benders

1.- INTRODUCCION

En la práctica, las dimensiones de muchos problemas de programación lineal son demasiado grandes para ser resueltos con los ordenadores actuales. El tipo de técnica desarrollada por Benders (1962) permite resolver problemas de grandes dimensiones en aquellos casos en los que se pueda identificar un conjunto de condiciones cuya resolución no ofrezca gran dificultad. Para ello se utilizará cierta dualización del otro conjunto de condiciones (denominadas *condiciones difíciles*) cuya estructura general no va a ser explotada.

Sea el siguiente problema (P) de programación lineal:

$$z^* = \min cx$$

s.a.

$$Ax = b$$

$$x \in X = \{ x : Dx \leq d, x \geq 0 \}$$

donde c es el vector de coeficientes, A y D son las matrices de condiciones, b y d los correspondientes términos independientes y x el vector de variables, todos ellos con las dimensiones apropiadas. Se supone que X es un conjunto no vacío y acotado. Sea $Ax = b$ el sistema de condiciones difíciles, y X el conjunto de soluciones que puede obtenerse de forma simple, bien porque está definido por subsistemas independientes con dimensiones relativamente pequeñas, o existen algoritmos polinomiales para su obtención, o cualquier otra razón que permita su consideración de forma explícita.

2.- ESQUEMA GENERAL EN LA DESCOMPOSICION DE BENDERS.

Sea (D) el problema dual del problema (P):

$$\begin{aligned} z^* &= \max wb + vd \\ \text{s.a.} \quad & wA + vD \leq c \\ & v \leq 0 \end{aligned}$$

donde w y v son vectores de variables con las dimensiones apropiadas.

Para una realización del vector w , sea el problema (D_w):

$$\begin{aligned} z_w &= wb + \max vd \\ \text{s.a.} \quad & vD \leq c - wA \\ & v \leq 0 \end{aligned}$$

Por tanto el problema (D) se puede escribir de la forma siguiente:

$$\begin{aligned} z^* &= \max_w z_w = \max_w wb + \max vd \\ \text{s.a.} \quad & vD \leq c - wA \\ & v \leq 0 \end{aligned}$$

Utilizando argumentos simples de teoría de la dualidad, resulta la siguiente formulación para el problema (D):

$$\begin{aligned} z^* &= \max_w wb + \min (c - wA)x \\ \text{s.a.} \quad & x \in X \end{aligned}$$

Dado que el conjunto X es no vacío y acotado, resulta que el punto óptimo para el problema $\{ \min (c-wA)x / x \in X \}$ para un w dado es un punto extremo de X . Sea $T = \{ x_1, x_2, \dots, x_t \}$ el conjunto (finito) de puntos extremos de X . Por tanto el problema (D) se puede reescribir como

$$z^* = \max_w wb + \min_{j=1,2,\dots,t} \{ (c-wA)x_j \}$$

y, en forma explícita, el problema (D) es equivalente al denominado *Problema Maestro* (PM)

$$\begin{aligned} &\max z \\ &\text{s.a.} \\ &z \leq wb + (c-wA)x_j \quad \forall j = 1, 2, \dots, t \end{aligned}$$

donde z es la variable a maximizar. Sea (z^*, w^*) su correspondiente valor óptimo.

Dado que el número t de puntos extremos de X puede ser finitamente grande no sería práctica la resolución explícita del problema (PM). Sin embargo, empíricamente se observa que una gran mayoría de las condiciones en (PM) se satisfacen con desigualdad estricta. Así, existe un conjunto $S \subseteq T$ de puntos extremos cuyas correspondientes condiciones en (PM) son suficientes para resolver el problema; es decir, existe un *problema maestro relajado* tal que obtiene el mismo vector óptimo (z^*, w^*) considerando un subconjunto de condiciones (i.e., puntos extremos) del problema maestro.

Sea (PMR) una relajación de este tipo para un número determinado, sea \bar{t} , de puntos extremos en X :

$$\begin{aligned} &\max z \\ &\text{s.a.} \\ &z \leq wb + (c-wA)x_j \quad \forall j = 1, 2, \dots, \bar{t}, \quad \bar{t} \leq t \end{aligned}$$

Sea (\bar{z}, \bar{w}) la correspondiente solución óptima.

Observación 1: \bar{z} es un límite superior para z^* .

Observación 2: $z^* = \bar{z}$ si y sólo si (\bar{z}, \bar{w}) satisface también las condiciones en (PM) que explícitamente no se han incluido en (PMR).

Observación 3: $z^* < \bar{z}$ si hay alguna condición en (PM) y, obviamente, no en (PMR) que sea violada por el vector (\bar{z}, \bar{w}) .

Para comprobar que todas las condiciones en (PM) son satisfechas por el vector (\bar{z}, \bar{w}) , es decir, para comprobar la condición

$$\bar{z} \leq \bar{w}b + (c - \bar{w}A)x_j \quad \forall j = 1, 2, \dots, t$$

o su equivalente
$$\bar{z} \leq \bar{w}b + \min_{j=1, \dots, t} \{ (c - \bar{w}A)x_j \}$$

es preciso resolver el siguiente *problema auxiliar* (PA):

$$\hat{z} = \bar{w}b + \min_{x \in X} \{ (c - \bar{w}A)x \}$$

donde x recoge el punto extremo correspondiente a la condición en (PM) con el menor valor en el término de la derecha. Sea dicho valor \hat{z} , tal que si $\hat{z} = \bar{z}$, entonces $\bar{z} = z^*$. En caso contrario (i.e., $\bar{z} > \hat{z}$) al menos el punto óptimo \hat{x} en (PA), sea x_k , corresponde a una condición violada en (PM), sea $z \leq \bar{w}b + (c - \bar{w}A)x_k$, que es preciso añadir a (PMR).

Dado que t es un número finito, el algoritmo que consiste en resolver iterativamente los problemas (PMR) aumentado y (PA) converge en un número finito de iteraciones.

Observación 4: Aunque se añade una condición al problema (PMR) en cada iteración, sería fácil instrumentar un esquema en el que se eliminasen condiciones que de una forma repetida sean fuertemente satisfechas, evitando siempre el problema del zig-zag.

Observación 5: Dada la estructura de las condiciones que definen el conjunto X , un paso crucial en los algoritmos basados en la descomposición de Benders es la rapidez con la cual debe resolverse el problema (PA).

3.- VARIABLES DUALES EN EL PROBLEMA MAESTRO RELAJADO.

El problema (PMR) puede también expresarse de la forma siguiente:

$$\begin{aligned} & \max w0 + z \\ & \text{s.a. } w(Ax_j - b) + z \leq cx_j \quad \forall j = 1, 2, \dots, \bar{t}, \quad \bar{t} \leq t \end{aligned}$$

donde 0 es el vector nulo, y el punto extremo x_j se supone conocido $\forall j = 1, 2, \dots, \bar{t}$.

Algebraicamente, el problema puede escribirse como

$$\begin{aligned} & \max (0, 1) \begin{pmatrix} w^t \\ z \end{pmatrix} \\ & \text{s.a.} \\ & \begin{pmatrix} (Ax_j - b)^t, 1 \end{pmatrix} \begin{pmatrix} w^t \\ z \end{pmatrix} \leq cw_j \end{aligned}$$

Sea (PDMR) el problema dual de (PMR), con la siguiente expresión:

$$\begin{aligned} & \bar{z} = \min \sum_j (cx_j)\lambda_j \\ & \text{s.a.} \\ & \begin{pmatrix} Ax - b \\ 1 \end{pmatrix} \lambda = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \end{aligned}$$

donde 1 es un vector de unos y λ es el vector dual tal que λ_j es la variable dual de la condición en (PMR) correspondiente al punto extremo x_j .

El sistema de condiciones en (PDMR) también puede expresarse como

$$\begin{aligned} & \sum_j (Ax_j)\lambda_j = b \sum_j \lambda_j = b \\ & \sum_j \lambda_j = 1 \\ & \lambda_j \geq 0 \quad \forall j = 1, 2, \dots, \bar{t} \end{aligned}$$

de donde resulta

$$\begin{aligned} \bar{z} &= \min \sum_j c_j \lambda_j \\ \text{s.á.} \quad & A \sum_j (x_j \lambda_j) = b \\ & \sum_j \lambda_j = 1 \end{aligned}$$

Cuando $\bar{z} = z^*$, se tiene que el vector que da la solución óptima del problema dual al (PM), y por tanto la solución al problema original, es el vector resultante de la combinación lineal de los puntos extremos x_1, \dots, x_t en X activos en el (PMR), tal que los coeficientes de ponderación son los valores de las variables duales de las condiciones del (PMR).

Así, en general se verifica que

$\bar{z} \geq z^* \geq \hat{z}$, con $\bar{x} = \sum_j x_j \lambda_j$, y la solución óptima para el problema (P) original se alcanza cuando $\bar{z} = z^* = \hat{z}$, en el punto $x^* = \sum_j x_j \lambda_j$.

Finalmente, es oportuno destacar que el esquema primal de la descomposición de Benders es la descomposición introducida independientemente en Dantzig y Wolfe (1960).

REFERENCIAS

- R. Aboudi, A. Hallefjord and K. Jornsten (1991), A facet generation and relaxation technique to an assignment problem with side constraints, *European Journal of Operational Research*, 50, pp. 335-344.
- R.K. Ahuja, T.L. Magnanti and J.B. Orlin (1989), Network flows, en: *Optimization*, G.L. Nemhauser, A.H.G. Rinnooy-Kan and M.J. Todd, eds., North-Holland, Amsterdam, pp. 211-369.
- R.K. Ahuja, T.L. Magnanti and J.B. Orlin, (1993), *Network Flows: Theory, Algorithms and Applications*, Prentice Hall, Englewood Cliffs, NY.
- N. Ascheuer, L.F. Escudero, M. Grötschel and M. Stoer (1993), A Cutting Plane Approach to the Sequential Ordering Problem (With Applications to Job Scheduling in Manufacturing), *SIAM J. on Optimization*, 3, pp. 25-42.
- E. Balas, M. Fischetti and W.R. Pulleyblank (1994), The Precedence-constrained Asymmetric Traveling Salesman Polytope, *Management Science Report MSRR-582*, Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburg, PA. (Aceptado para publicación en *Mathematical Programming*).
- J.F. Benders (1962), Partitioning Procedures for Solving Mixed Variables Programming Problems, *Numerische Methematik*, 4, pp. 238-252.
- D.P. Bertsekas (1982), *Constrained Optimization and Lagrange Multipliers methods*, Academic Press, NY.
- D.P. Bertsekas (1991), *Linear Network Optimization*, MIT Press, Cambridge, MA.
- G.B. Dantzig, D.R. Fulkerson and S.M. Johnson (1954), Solution of a Large-Scale Traveling Salesman Problem, *Operations Research* 2, pp. 393-410.
- G.B. Dantzig and P. Wolfe (1960), A decomposition principle for linear programs, *Operations Research* 8, pp. 101-111.

- B.L. Dietrich and L.F. Escudero (1992), On Tightening Cover Induced Inequalities, *European Journal of Operational Research*, 60, pp. 335-343.
- B.L. Dietrich, L.F. Escudero, A. Garín and G. Pérez (1993), $O(n)$ Procedures for Identifying Maximal Cliques and Non-Dominated Extensions of Consecutive Minimal Covers and Alternates, *Top*, 1, pp. 139-160.
- J. Edmonds (1967), Optimum Branching, *Journal of Research of the National Bureau of Standards B-71*, pp. 233-240.
- L.F. Escudero (1988a), An Inexact Algorithm for the Sequential Ordering Problem, *European Journal of Operational Research*, 37, pp. 236-253.
- L.F. Escudero (1988b), On the Implementation of an Algorithm for Improving a Solution to the Sequential Ordering Problem, *Trabajos de Investigación Operativa*, 3, pp. 117-140.
- L.F. Escudero (1989), A Production Planning Problem in FMS, *Annals of Operations Research*, 17, pp. 69-104.
- L.F. Escudero and A. Sciomachen (1993a), A $O(n^3)$ Approximate Algorithm for Finding a Feasible Solution for the Sequential Ordering Problem, *Journal of Combinatorics, Information and System Sciences*, (aceptado para publicación).
- L.F. Escudero and A. Sciomachen (1993b), Local Search Procedures for Improving Feasible Solutions to the Sequential Ordering Problem, *Annals of Operations Research*, 43, pp. 397-416.
- M. Fischetti and P. Toth (1993), An Efficient Algorithm for the Min-Sum Arborescence Problem on Complete Digraphs, *ORSA Journal on Computing*, 5, pp. 426-434.
- M.L. Fisher (1981), The Lagrangian relaxation method for solving integer programming problems, *Management Science*, 27, pp. 1-18.
- L.R. Ford and D.R. Fulkerson (1956), Maximal Flow Through a Network, *Canadian Journal of Mathematics* 8, pp. 399-404.
- L.R. Ford and D.R. Fulkerson (1962), *Flows in Networks*, Princeton University Press, Princeton, NJ.
- G. Gallo and S. Pallottino (1988), Shortest Path Algorithms, *Annals of Operations Research*, 13, pp. 3-79.

- M.R. Garey and D.S. Johnson (1979), *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, San Francisco.
- A.M. Geoffrion (1974), Lagrangian Relaxation and its Uses in Integer Programming, *Math. Programming Study*, 2, pp. 82-114.
- A.V. Goldberg and R.E. Tarjan (1988), A New Approach to the Maximum Flow Problem, *Journal of ACM*, 35, pp. 921-940.
- M. Grötschel (1977), *Polyedrische Charakterisierungen Kombinatorischer Optimierungsprobleme*, Hain, Meisenheim am Glan, Germany.
- M. Grötschel, M. Junger and G. Reinelt (1984), A Cutting Plane Algorithm for the Linear Ordering Problem, *Operations Research*, 32, pp. 1195-1220.
- M. Grötschel, L. Lovasz and A. Schrijver (1988), *Geometric Algorithms and Combinatorial Optimization*, Springer-Verlag, Berlin.
- M. Grötschel and O. Holland (1991), Solution of Large-scale Symmetric Travelling Salesman Problem, *Mathematical Programming* 51, pp. 191-202.
- M. Guignard and S. Kim (1987), Lagrangean Decomposition: A Model Yielding Stronger Lagrangean Bounds, *Mathematical Programming* 39, pp. 215-228.
- M. Held and R.M. Karp (1970), The Traveling Salesman Problem and Minimum Spanning Trees, *Operations Research*, 18, pp. 1138-1162.
- M. Held and R.M. Karp (1971), The Traveling Salesman Problem and Minimum Spanning Trees: Part II, *Mathematical Programming*, 1, pp. 6-25.
- M. Held, P. Wolfe and H.P. Crowder (1974), Validation of Subgradient Optimization, *Mathematical Programming*, 6, pp. 62-68.
- P.C. Kanellakis and C.C. Papadimitriou (1978), Local Search for the Asymmetric Travelling Salesman Problem, *Operations Research*, 26, pp. 1086-1098.
- N. Karmarkar (1984), A New Polynomial Time Algorithm for Linear Programming, *Combinatorica*, 4, pp. 375-395.
- L.G. Khachian (1979), A Polynomial Algorithm in Linear Programming, *Soviet Mathematics Doklady* 20, pp. 191-194.
- J.B. Kruskal (1956), On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem, *Proceedings of the American Mathematical Society* 7, pp. 48-50.

- E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan and D.B. Shmoys, eds. (1985) *The Traveling Salesman Problem. A guided tour of combinatorial optimization*, J. Wiley, NY.
- S. Martello and P. Toth (1984), A Mixture of Dynamic Programming and Branch-and-bound for the Subset Sum Problem, *Management Science*, **30**, pp. 765-771.
- S. Martello and P. Toth (1988), A New Algorithm for the 0-1 Knapsack Problem, *Management Science*, **34**, pp. 633-644.
- S. Martello and P. Toth (1990), *Knapsack Problems: Algorithms and Computer Implementations*, J. Wiley, NY.
- J.B. Orlin (1993), A Faster Polynomial Minimum Cost Flow Algorithm, *Operations Research*, **4**, pp. 338-350.
- M. Padberg and M. Grötschel (1985), Polyhedral Computations, en: E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy-Kan and D.B. Shmoys, eds., *The Traveling Salesman Problem. A guided tour of combinatorial optimization*, J. Wiley, pp. 251-360.
- M.W. Padberg and M.R. Rao (1982), Odd Minimum Cut-Sets and b-Matchings, *Mathematics of Operations Research* **7**, pp. 67-80.
- M.W. Padberg and G. Rinaldi (1990), An Efficient Algorithm for the Minimum Capacity Cut Problem, *Mathematical Programming* **47**, pp. 19-36.
- M.W. Padberg and G. Rinaldi (1991), A Branch-and-cut Algorithm for the Resolution of Large-scale Symmetric Salesman Problems, *JAM Review* **33**, pp. 60-100.
- C.H. Papadimitriou and K. Steiglitz (1982), *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall, NY.
- C.R. Reeves (1993), *Modern heuristic techniques for Combinatorial Problems*, Blackwell Scientific Publications, Oxford.