UNIVERSIDAD DE VALENCIA

Facultad de Matemáticas

"Circuitos Eulerianos óptimos en grafos mixtos: El problema del Cartero Mixto."

TESIS DOCTORAL

Presentada por

Vicente Campos Aucejo

Dirigida por el Catedrático

D. Marco Antonio López Cerdá

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

CIRCUITOS EULERIANOS OPTIMOS EN GRAFOS MIXTOS : EL PROBLEMA DEL CARTERO MIXTO.

Memoria presentada por D.Vicente Campos Aucejo para optar al grado de Doctor en Ciencias Matemáticas.
Realizada bajo la dirección de D.Marco Antonio López Cerdá, Catedrático de Investigación Operativa de la Facultad de Matemáticas de la Universidad de Valencia.

A Alicia.

CERTIFICADO

MARCO A. LOPEZ CERDA, Catedrático de Investigación Operativa de la Facultad de Matemáticas de la Universidad de Valencia.

CERTIFICA: Que la presente memoria "*Circuitos Eulerianos óptimos en grafos mixtos: El problema del Cartero Mixto*", ha sido realizada bajo su dirección en el Departamento de Estadística Matemática de la Facultad de Matemáticas de la Universidad de Valencia por el Sr.D.Vicente Campos Aucejo, y constituye su tesis para optar al grado de Doctor en Ciencias Matemáticas.

Y para que conste en cumplimiento de la legislación vigente, presenta ante la Facultad de Matemáticas de la Universidad de Valencia, a 19 de agosto de 1982.

Firmado:

Marco A. López Cerdá

Nuestro agradecimiento a todas las personas que nos han ayudado y estimulado, y en particular, al personal del Centro de Cálculo de la Universidad Politécnica de Valencia, sin cuya amable colaboración no habría sido posible la realización de una parte importante de este trabajo.

INDICE

RESUMEN

En 1736 Euler plantea y resuelve el problema de la existencia, o no, de un circuito que recorra exactamente una vez todas las aristas de un grafo no dirigido, como extensión del famoso problema de los puentes de Köningsberg reconocido como uno-de los problemas iniciadores de la Teoría de Grafos. Una generalización fue propuesta por Kwan-Mei-Ko en el año 1962, con lo -- que más tarde se conocería como el problema del Cartero Chino -- (CPP): dado un grafo conexo y no dirigido, con costes no negativos asociados a sus aristas, hallar un circuito que recorra to-- das las aristas al menos una vez con el mínimo coste. Cuando el (CPP) se plantea sobre un grafo "dirigido" , se conoce con el -- nombre del problema del Cartero Chino Dirigido (DCPP). Para am-- bos problemas se conocen algoritmos exactos de resolución. No -- así, ocurre cuando el grafo sobre el que se intenta minimizar el coste del circuito, contiene líneas que pueden ser recorridas en ambos sentidos y líneas donde sólo se permite uno, los grafos -- con estas características son llamados grafos mixtos, y el pro-- blema del Cartero Mixto (MCPP) es un problema abierto a la investigación en su caso más general.

La presente memoria ofrece el estudio del (MCPP), así como un algoritmo exacto de resolución basado en técnicas de la Programación Lineal Entera y algoritmos de Teoría de Grafos.

Dado que toda la memoria apunta hacia un mismo objetivo, hemos dividido la misma en Secciones que están fuertemente interrelacionadas.

En la Sección 1 ofrecemos algunos conceptos y re-- sultados conocidos de la Teoría de Grafos, marcando las diferencias existentes entre conceptos y propiedades relativos a grafos no dirigidos, dirigidos y mixtos, estos últimos apenas referidos en la mayoría de los textos que tratan sobre grafos. Además se - incluye un resumen de la Teoría de la Relajación Lagrangiana, -- que será utilizada en este trabajo.

En la Sección 2 comenzamos comentando los antece--

dentes históricos del (MCPP), junto con una exposición del estado actual del problema que comporta la descripción de los algoritmos que producen soluciones posibles (no necesariamente - óptimas) al (MCPP), así como las distintas propuestas de resolución exacta del problema.

En la Sección 3 formulamos el (MCPP) como un problema de Programación Lineal Entera (P.L.E.) y un estudio de - la conectividad del grafo mixto original, que puede ser trasladado también a grafos derivados de él. Al final de esta Sección se considera la existencia de un conjunto de restricciones redundantes a la formulación dada del (MCPP), que serán utilizadas para construir un problema relajado en la forma lagrangiana.

En la Sección 4 proponemos un algoritmo heurístico que proporciona soluciones posibles al (MCPP), comparándolo con el obtenido a partir de la solución óptima del problema lineal, obtenido al eliminar las restricciones de integridad sobre las variables del problema lineal entero que resuelve el-- (MCPP).

En la Sección 5 estudiamos tres relajaciones del problema, de las que dos son obtenidas por métodos lagrangianos y que suponen respectivamente, la resolución de un problema lineal (RLin), la de un problema de 1-acoplamiento y la de un problema de flujo de coste mínimo. Los valores óptimos de - cada una de las relajaciones, producen cotas inferiores al valor óptimo del (MCPP).

En la Sección 6 y última, describimos las partes fundamentales de las que consta el algoritmo de Branch and --- Bound diseñado para resolver el (MCPP), ofreciendo el comportamiento del mismo en una colección de 35 problemas de distintos tamaños, de hasta 50 vértices, 66 arcos y 39 aristas, obteniendo la solución óptima en no más de 500 segundos de C.P.U en un computador UNIVAC 1100/60. La Sección concluye con los resultados computacionales y una discusión de los mismos.

S E C C I O N 1

INTRODUCCION

1.1 TEORIA DE GRAFOS

A causa de la escasa standarización existente en los conceptos utilizados en Teoría de Grafos, incluso en la definición de grafo, es necesario ofrecer un resumen de las definiciones y resultados más importantes de los que se hace uso en esta memoria, con el fin de evitar ambigüedades y posibles interpretaciones erróneas, facilitando así la lectura de ésta. Por otra parte, en los textos más utilizados en Teoría de Grafos, como los de Christofides (1975), Berge (1962), Harary (1969), etc..., no se hace referencia explícita a los grafos mixtos, lo cual pretendemos subsanar en esta introducción.

1.1.1 Definiciones y conceptos básicos

Un grafo G, es una colección de puntos ó vértices $i_1,i_2,\ldots,i_n$ que representaremos por el conjunto N, y una colección de líneas $t_1,t_2,\ldots,t_m$ representadas por el conjunto T, que unen todos o algunos de estos puntos. La línea que une dos puntos (no necesariamente distintos) de un grafo, se dice que es incidente con ellos. Si algunas de las líneas de T tienen asignada una dirección, reciben el nombre de arcos, y al conjunto de todos ellos lo representaremos por A. Las líneas de T que no tienen una dirección asignada se llaman aristas, y el conjunto de todas ellas lo representaremos por L. Una arista será representada por medio de un par no ordenado de la forma $\{i_p,i_q\}$, siendo i_p e i_q los vértices que une dicha arista,que son llamados terminales. Un arco será representado por un par ordenado de la forma (i_p,i_q), si la dirección asignada es de i_p a i_q. Cuando exista más de un arco o arista entre un mismo par de vértices de un grafo, los diferenciaremos utilizando subíndices, así $(i_p,i_q)_r$, representa el arco r-ésimo dirigido del vértice inicial (o de

salida) i_p, al vértice final (o de llegada) i_q. Una arista es fácilmente transformable en un arco, asignándole uno de sus dos sentidos posibles, y recíprocamente un arco puede convertirse en una arista si se ignora su dirección. Un grafo G en el que todas sus líneas sean aristas se llama grafo no dirigido, y será representado por el par (N,L). Un grafo G en el que todas sus líneas sean arcos se llama grafo dirigido, y se representa por el par (N,A). Un grafo G que contenga arcos y aristas se llama grafo mixto, y se representa por la terna (N,A,L). En la representación gráfica de un grafo, las líneas que representan a los arcos poseen unas flechas que indican la dirección asignada a cada uno de ellos. Un grafo no dirigido tal que, entre cada dos vértices distintos exista una arista, se llama completo. Una arista que una un vértice consigo mismo, recibe el nombre de bucle.

En las definiciones que se van a dar a continuación, cuando nos refiramos a un grafo mixto, entenderemos que posee conjuntos no vacíos de arcos y aristas, no obstante, la mayoría de ellas son fácilmente reducibles a grafos dirigidos y no dirigidos, desprendiéndose del contexto de la definición.

Diremos que un grafo mixto es simple, si no contiene bucles, y además, dados dos vértices distintos i_p, i_q cualesquiera, no existe entre ellos: ni más de una arista (representada indistintamente por $\{i_p,i_q\}$ ó $\{i_q,i_p\}$), ni más de un arco en la misma dirección. Un grafo mixto que no es simple, se llama multigrafo.

Un camino dirigido en un grafo mixto, es una sucesión de arcos $(a_1,a_2,\ldots,a_q)$, en donde el vértice final de uno cualquiera de ellos es el vértice inicial del siguiente. Un camino no dirigido en un grafo mixto, es una sucesión de aristas $(l_1,l_2,\ldots,l_r)$, en la que cada arista l_i,

excepto quizás la primera y la última, está conectada a las aristas l_{i-1} y l_{i+1}, por sus dos vértices terminales. Un camino en un grafo mixto, es una sucesión de arcos y aristas (no necesariamente distintos) $(a_1, l_2, \ldots, l_{q-1}, a_q)$, de forma que, asignando una y solo una de las dos direcciones posibles a cada arista (no necesariamente la misma para todas sus apariciones), y convirtiéndola por lo tanto en arco, se obtiene un camino dirigido. El término camino, será utilizado también cuando nos refiramos a grafos dirigidos o no dirigidos, sin más adjetivos; por otro lado, cuando en lo sucesivo se dé alguna definición y/o propiedad, referente a un grafo G, que sea independiente de si G es dirigido, no dirigido o mixto, nos referiremos a G simplemente por el término "grafo".

Un circuito en un grafo, es un camino cuyos vértices inicial y final coinciden. Si el circuito no utiliza el mismo arco, o arista, más de una vez, se llamará circuito simple. Un circuito simple que utilice todas las líneas de un grafo, se llama circuito euleriano. Un grafo que admita un circuito euleriano se llama unicursal o euleriano.

Dado un grafo G definimos para cada vértice i de él:

grado del vértice i, representado por $d^G(i)$, como el número de líneas de G incidentes con i.

grado de entrada del vértice i, representado por $d_e^G(i)$, es el número de arcos cuyo vértice final es i.

grado de salida del vértice i, representado por $d_s^G(i)$, es el número de arcos cuyo vértice inicial es i.

grado respecto a aristas del vértice i, representado por $d_a^G(i)$, es el número de aristas incidentes con i.

Tenemos que $d^G(i) = d_e^G(i) + d_s^G(i) + d_a^G(i)$, para todo $i \in N$. Cuando no exista ambigüedad acerca del grafo de referencia, no colocaremos el su

praíndice. Diremos que la paridad de un vértice i es par (impar), si d(i) es par (impar). Si la paridad de un vértice es par (impar), diremos que el vértice es par (impar). Diremos que un vértice i es pseudosimétrico, si $d_e(i) = d_s(i)$. Un grafo donde todos sus vértices sean pseudosimétricos, se llama simétrico.

Teorema 1.1

En un grafo cualquiera, el número de vértices impares es par.

Dado un grafo mixto G=(N,A,L), llamaremos grafo parcial propio G_p de G, a cualquier grafo $G_p=(N,A_p,L_p)$, siendo $A_p \subset A$ y $L_p \subset L$ (i.e: un grafo con el mismo conjunto de vértices, pero con solamente un subconjunto propio de arcos y aristas del grafo dado). Diremos que un grafo G es maximal con respecto a cierta propiedad $\mathcal{P}$ referente a la estructura de G, si cumple dicha propiedad, y no existe ningún grafo parcial propio de él que la posea. Un subgrafo G_S del grafo G, es todo grafo formado por un subconjunto de vértices S del grafo G y por todos los arcos y/o aristas que son incidentes con algún vértice de S. Diremos que un vértice i_q es alcanzable desde el vértice i_p, si existe un camino con vértice inicial i_p y vértice final en i_q. Al conjunto de vértices alcanzables desde un vértice i, se le llama conjunto de accesibilidad de i, y se representa por R(i). Por definición $i \in R(i)$, para todo i. Dos vértices i_p, i_q, se dice que están conectados, si son mutuamente alcanzables (i.e: $i_p \in R(i_q)$ y $i_q \in R(i_p)$). Diremos que un grafo G, no dirigido, es conexo, si todos los pares de vértices de él están conectados. Diremos que un grafo G, mixto (dirigido), es fuertemente conexo, si todos los pares de vértices de él están conectados. Diremos que un grafo G, mixto (dirigido), es débilmente conexo, si el grafo no dirigido inducido por todas las líneas de G, ignorando la dirección de los arcos, es conexo. Un subgrafo maximal respecto a la propiedad de la

"conexión fuerte", se llama componente conexa.

Teorema 1.2

Un grafo mixto G=(N,A,L) es euleriano sii:

(a) G es fuertemente conexo.

(b) G es par.

(c) Para todo subconjunto $X \subset N$, la diferencia entre el número de arcos dirigidos de X a $\overline{X}$ (complementario de X) y el número de arcos de $\overline{X}$ a X es menor o igual que el número de aristas conectando vértices de X con vértices de $\overline{X}$.

Teorema 1.3

Un grafo no dirigido G=(N,L) es euleriano sii:

(a) G es conexo.

(b) G es par.

Teorema 1.4

Un grafo dirigido G=(N,A) es euleriano sii:

(a) G es fuertemente conexo.

(b) G es simétrico.

La condición a de los Teoremas anteriores, caracteriza a los grafos que admiten un circuito (no necesariamente simple), que contenga a todas las líneas de ellos; a un circuito tal, si existe, le llamaremos tour. Los términos "circuito euleriano" y "tour euleriano" son sinónimos.

Un matching o acoplamiento en un grafo no dirigido, es cualquier subconjunto M del conjunto de aristas del grafo, tal que, no existen dos aristas en M que tengan un vértice terminal común. Dado un matching M, un camino alternado respecto de M es un camino simple cuyas aristas están alternativamente en M y no en M. Un vértice que no sea ter

minal de ninguna arista en M, se llama vértice aislado respecto de M. Un camino aumentado es un camino alternado cuyos vértices inicial y final son aislados. Diremos que un matching M es perfecto, respecto a un grafo G, si todo vértice de G no es aislado. Un 1-matching es un matching M de forma que entre dos vértices cualesquiera existe, a lo sumo, una arista de M.

Con frecuencia, los grafos llevan asociados en cada una de sus líneas, unos números que llamaremos costes y que usualmente representan la distancia, o tiempo, de recorrer la línea en cuestión. Los problemas de localización y de rutas de vehículos, por ejemplo, tienen una clara escenificación en Teoría de Grafos, utilizando grafos que representan un plano de un área determinada, y en donde se intenta minimizar una función dependiente de los costes. Los conceptos y algoritmos que siguen,se refieren a grafos con costes c_{ij}, asociados a sus arcos y/o aristas.

Dado un grafo G cualquiera, representaremos por c(G) a la suma de todos los costes asociados a las líneas de G. Si P es un camino, c(P) representa la suma de todos los costes de los arcos y/o aristas que aparezcan en él.

El problema del camino más corto

El problema de encontrar el camino de coste mínimo entre dos vértices especificados de un grafo, se conoce con el nombre de problema del camino más corto (en grafos mixtos y dirigidos, el camino más corto de ir de i_p a i_q puede no ser el mismo que el camino más corto de ir de i_q a i_p), y puede extenderse al problema de encontrar los caminos más cortos entre todos, o un subconjunto de vértices del grafo.

En la actualidad existen algoritmos eficientes (i.e:acotados polinomialmente en tiempo de resolución) para resolver éstos problemas, así pues, el algoritmo de Dijsktra es el más apropiado para hallar los cami

nos más cortos entre dos vértices, o un vértice y los del resto del grafo , o entre un subconjunto relativamente pequeño de vértices, mientras que los algoritmos de Floyd, Murchland y Dreyfus son más eficaces en los demás casos. Abreviadamente, representaremos por c.m.c(i,j) al coste del camino más corto de i a j, en este orden.

Problemas de acoplamientos

El problema de encontrar un 1-acoplamiento perfecto de mínimo coste en un grafo no dirigido G, con costes asociados c_j a cada una de sus aristas l_j, puede formularse como un problema de Programación Lineal Entera (P.L.E) de la forma siguiente:

$$\text{Min } z = \sum_j c_j x_j$$

sujeto a:

$$\sum_{a_j \in L(i)} x_j = 1 \qquad \forall i \in N \qquad (1.1)$$

siendo N el conjunto de vértices en el grafo G, L(i) el conjunto de todas las aristas que son incidentes con el vértice i, y x_j es una variable binaria que vale 0 ó 1, dependiendo de si la arista asociada a_j no está, o sí forma parte de las aristas que forman el acoplamiento.

También existen algoritmos eficientes para este problema, muy estudiados y elaborados, que pueden encontrarse en Christofides(1975) y especialmente en Lawler(1976), quién dedica dos capítulos de su libro a los problemas de acoplamientos y sus aplicaciones.

Problemas de flujo de coste mínimo

Dado un grafo dirigido G=(N,A), en el que para cada vértice i existe un número entero b_i que llamaremos "demanda" de i (si $b_i < 0$), u "oferta" de i (si $b_i \geq 0$), y asociado con cada arco (i,j) la variable x_{ij} representa el número de unidades de flujo que circula por él, siendo c_{ij}

el coste de transportar una unidad de flujo por dicho arco.

El problema de flujo de coste mínimo, consiste en transportar toda la oferta disponible desde los vértices "fuentes" (i.e: con $b_i > 0$) hacia los vértices "sumideros" (i.e: con $b_i < 0$), para satisfacer la demanda total con el mínimo coste, donde se supone que la oferta y demanda totales son iguales, sin pérdida de generalidad. El problema, admite una formulación por P.L.E definida por:

$$\text{Min} \quad \sum_{A} c_{ij}\, x_{ij}$$

sujeto a:

$$\sum_{j} x_{ij} - \sum_{j} x_{ji} = b_i \qquad \forall i \in N \qquad (1.2)$$

$$x_{ij} \geq 0 \qquad \forall (i,j) \in A \qquad (1.3)$$

donde las restricciones (1.2) se llaman "de conservación del flujo" o de Kirchoff. Mientras que las restricciones (1.3) son, a menudo, reemplazadas por

$$l_{ij} \leq x_{ij} \leq q_{ij} \qquad \forall (i,j) \in A \qquad (1.4)$$

siendo l_{ij}, q_{ij} dos números enteros no negativos que representan el número mínimo y máximo, respectivamente, de unidades de flujo que han de ser transportadas por cada arco (i,j).

Fulkerson en 1961 diseñó algoritmos eficientes para resolver este problema, entre los que destaca el algoritmo "out of kilter", similar al método primal-dual del simplex, si bien, difiere de éste en que no siempre mantiene las condiciones del Teorema de la Holgura Complementaria. Remitimos a Ford y Fulkerson (1962) y Bazaraa(1977) para una información más detallada.

Transformación de un multigrafo en un grafo simple

Para los propósitos de esta memoria,y con objeto de simplificar la notación, es conveniente relacionar de una forma biunívoca cada arco y/o arista, por los dos vértices que unen; es por esto que ofrecemos una fácil transformación de un multigrafo en un grafo simple, sin que se alteren las propiedades de conectividad del mismo y donde cualquier camino en el grafo transformado, puede ser convertido en un camino del grafo original y viceversa, de una forma trivial.

La Figura 1.1.a muestra un multigrafo mixto con cuatro vértices, tres arcos representados por $a_1 = (1,2)_1$, $a_2 = (1,2)_2$, $a_3 = (2,3)$, y cuatro aristas $l_1 = \{1,4\}$, $l_2 = \{2,4\}$, $l_3 = \{3,4\}_1$, $l_4 = \{3,4\}_2$.

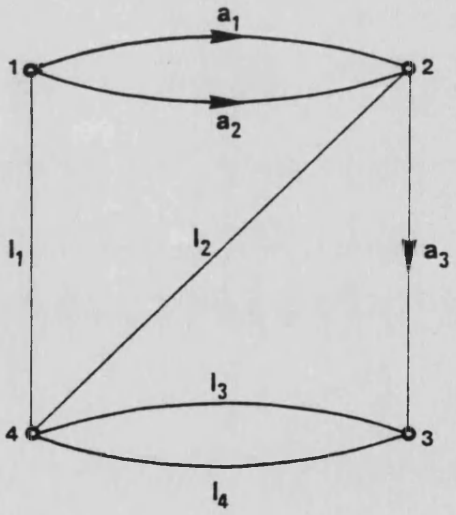


Figura 1.1.a:Ejemplo de multigrafo mixto

Se obtiene un grafo simple, convirtiendo todos excepto uno de los arcos en la misma dirección, o aristas que aparecen entre un mismo par de vértices en el multigrafo, en dos arcos dirigidos en el mismo sentido del arco que sustituyen, o en dos aristas, que tienen un nuevo vértice en común. En el multigrafo de la Figura 1.1.a, puesto que, existen dos arcos en paralelo

a_1 y a_2 entre el mismo par de vértices 1 y 2, incorporamos un nuevo vértice 5, convirtiendo el arco a_2 en los arcos $a_2' = (1,5)$ y $a_2'' = (5,2)$; de la misma forma incorporando un nuevo vértice 6 y sustituyendo la arista 1_4 por las aristas $1_4' = \{3,6\}$ y $1_4'' = \{6,4\}$, obtenemos el grafo simple que se muestra en la Figura 1.1.b.

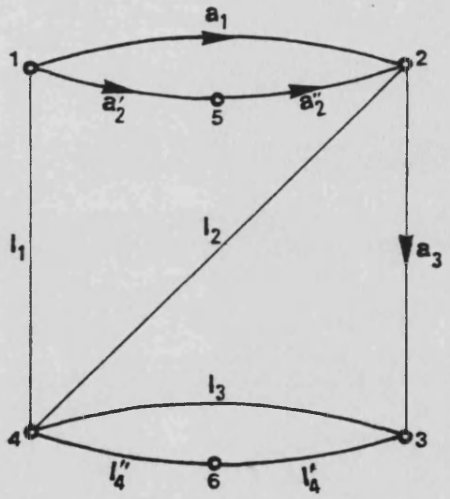


Figura 1.1.b: Grafo transformado simple.

Si además, el grafo original tiene costes asociados a sus líneas, podemos realizar la misma transformación,asignando a uno de los dos arcos o aristas que sustituyen a un arco o arista del multigrafo original, el coste de éste, y al otro coste cero, con lo que no varían las distancias entre dos puntos cualesquiera del multigrafo original.

1.2 LA RELAJACION LAGRANGIANA

Consideremos el problema de minimización discreta P definido por

(P) Min cx

sujeto a:

$$Ax \geq b$$

$$Bx = d$$

$$x \geq 0,\ x_j \text{ entero } \forall j \in I$$

donde c,x,b,d son vectores, y A,B matrices de dimensiones adecuadas, mientras que cx representa la función objetivo (producto escalar) a minimizar. Representaremos por F(P) al conjunto discreto de soluciones posibles de P. Supondremos que las restricciones $Ax \geq b$, son las que dificultan la resolución de P, y que el conjunto $\{x: x \geq 0,\ Bx = d \text{ y } x_j \text{ entero}, \forall j \in J\}$, tiene una estructura especial que pretendemos explotar.

La dualización del problema P respecto a las restricciones $Ax \geq b$, consiste en incorporarlas a la función objetivo, mediante los multiplicadores de Lagrange $\lambda \geq 0$, generando el problema lagrangiano:

PR(x,λ) Min $cx + \lambda(b - Ax)$

sujeto a:

$$Bx = d$$

$$x \geq 0,\ x_j \text{ entero}, \forall j \in I$$

La utilidad de este problema relajado, depende en cada caso específico, de las características especiales de los conjuntos de restricciones $Ax \geq b$, $Bx = d$, y de la elección de los multiplicadores λ , donde evidentemente la relajación tendrá sentido si el problema PR(x,λ) es más fácil de resolver que P, por "más fácil" entenderemos, que puede resolverse mediante un algoritmo cuyo tiempo de resolución está acotado por un polinomio que es fun-

ción de los datos de entrada del problema.

Al problema PR(x,λ) se le suele llamar según la definición dada por Geoffrion, relajación lagrangiana de P con respecto a las restricciones $Ax \geq b$, y al vector λ. La expresión relajación hace referencia a unas relaciones entre el conjunto de soluciones posibles del problema de optimización P y su problema relajado PR. Se tiene por lo tanto la siguiente definición.

Definición 1.5

Un problema de minimización (max.) PR se dice que es una relajación de un problema de minimización (max.) P sii:

(a) $F(P) \subseteq F(PR)$.

(b) $v(P) \geqslant v(PR)$ en el caso de minimización ó

$v(P) \leqslant v(PR)$ en el caso de maximización.

donde v(.) representa el valor óptimo del problema (.) y F(.) representa el conjunto de soluciones posibles de (.).

De acuerdo con la Definición 1.5, se tiene que, del problema P se obtiene una relajación inmediata, eliminando las condiciones de integridad sobre las variables x, obteniendo la relajación lineal definida por el problema

$(\overline{P})$ Min cx

sujeto a:

$Ax \geq b$

$Bx = d$

$x \geq 0$

La eficacia de cualquier relajación viene determinada por las condiciones siguientes:

(i) La dificultad de resolver el problema PR. Donde la dificultad

se expresa usualmente, con una cota superior al tiempo de resolución.

(ii) Proximidad entre v(P) y v(PR).

(iii) La posibilidad de obtener "buenas" soluciones posibles de P a partir de soluciones óptimas de PR.

Relativo a la condición (ii), existe el problema de elegir un "buen" vector de multiplicadores λ, es decir, de forma que $v(PR(x,\lambda))$ esté lo más próximo posible a v(P), y el mejor es aquél vector λ^*, tal que:

$$(D) \qquad \underset{\lambda \geq 0}{\text{Max}} \; v(PR(x,\lambda)) = v(PR(x,\lambda^*))$$

es decir aquél que proporciona la máxima cota inferior, aunque para ello hay que resolver el problema cóncavo D, que a su vez está relacionado con otra relajación importante de P, consistente en sustituir el conjunto $\{x: x \geq 0, Bx = d, x_j \text{ entero}, j \in I\}$ por su envoltura convexa, dando origen a otro problema P^*, cuya formulación es:

$$(P^*) \qquad \text{Min } cx$$

sujeto a:

$$Ax \geq b$$

$$x \in \text{Co}\{x: x \geq 0, Bx=d, x_j \text{ entero}, j \in I\}$$

El problema P^* puede considerarse como un problema de programación lineal, si bien, la envoltura convexa referida, no es fácil de representar por medio de un sistema de desigüaldades ó inecuaciones. Los problemas (P^*) y (D) son, esencialmente, problemas duales respecto de las restricciones - - $Ax \geq b$, pues el mínimo de una función lineal en un compacto no cambia si se sustituye por su envoltura convexa. Las relaciones entre los problemas relajados, definidos anteriormente, vienen dados por el siguiente Teorema debido a Geoffrion en 1974.

Teorema 1.6

(a) $F(P) \subseteq F(P^*) \subseteq F(\overline{P})$

$F(P) \subseteq F(PR(x,\lambda))$

$v(P) \geqslant v(P^*) \geqslant v(\overline{P})$

$v(P) \geqslant v(PR(x,\lambda)) \qquad \lambda \geqq 0.$

(b) Si $(\overline{P})$ es posible, entonces $v(\overline{P}) \leqslant v(PR(x,\overline{\lambda}))$, siendo $\overline{\lambda}$ un vector óptimo de la solución del problema dual de $(\overline{P})$, asociado con las restricciones $Ax \geqq b$.

(c) Si para un $\lambda \geqq 0$ dado, un vector $\hat{x}$ satisface las condiciones:

(i) $\hat{x}$ es óptimo para $PR(x,\lambda)$.

(ii) $\lambda(b - A\hat{x}) = 0.$

(iii) $A\hat{x} \geqq b.$

entonces $\hat{x}$ es una solución óptima de P.

Puede ocurrir, que dado un vector λ, la solución del problema lagrangiano $PR(x,\lambda)$ nos proporcione un vector $\hat{x}$ que cumpla las condiciones (i) e (iii), pero no (ii), en cuyo caso $\hat{x}$ es una solución ε-óptima de P, siendo $\varepsilon = \lambda(A\hat{x} - b) > 0$. De aquí se deduce que $PR(x,\lambda)$ puede proporcionar soluciones posibles para P, cuya calidad estará limitada por el distanciamiento entre v(D) y v(P) si es que existe.

(d) Si P^* es posible, entonces:

$$v(D) = \max_{\lambda \geqq 0} v(PR(x,\lambda)) = v(PR(x,\lambda^*)) = v(P^*).$$

La parte (d) del Teorema, establece que la relajación lagrangiana puede proporcionar cotas tan buenas pero no mejores, que (P^*), y como consecuencia de ello, la posición de $v(P^*)$ en el intervalo $[v(\overline{P}), v(P)]$ es fundamental para analizar el valor potencial de la relajación lagrangiana aplicada a un problema en particular.

Propiedad de la Integridad

Diremos que el problema $PR(x,\lambda)$ tiene la propiedad de la integridad sii el valor óptimo de $PR(x,\lambda)$ es el mismo, ignorando la restricción de que sus variables tomen valores enteros, i.e: $v(PR(x,\lambda)) = v(\overline{PR(x,\lambda)})$.

Teorema 1.7

Si P es posible, y $PR(x,\lambda)$ tiene la propiedad de la integridad, entonces:

(i) P^* es posible.

(ii) $v(\overline{P}) = v(PR(x,\overline{\lambda})) = v(D) = v(PR(x,\lambda^*)) = v(P^*)$.

Como consecuencia de este Teorema, tenemos que si la partición de las restricciones del problema P, en los conjuntos $Ax \geq b$ y $Bx = d$ es tal que éste último satisface la propiedad de la integridad, entonces la relajación lagrangiana no puede proporcionar cotas inferiores mejores que la relajación lineal. La relajación lagrangiana es interesante en este caso, solamente si se pueden obtener soluciones ε-óptimas para el problema D, por métodos especializados, de forma más eficiente que resolver $\overline{P}$ por los métodos usuales de la Programación Lineal.

El método del subgradiente

El método más comúnmente utilizado para determinar v(D), fue el empleado por Held y Karp en 1970, para resolver el problema del agente viajero (TSP), denominado "método del subgradiente", es una aplicación del método más general de Agmon-Motzkin-Schoenberg, para resolver sistemas con inecuaciones lineales. En un trabajo posterior, Held, Wolfe y Crowder en [36], estudian el método con detalle, presentando resultados de su aplicación en varios problemas de optimización discreta. Algunos de los resultados más importantes, y estudio de la convergencia del método del subgradiente, pueden encontrarse en [58] y [59].

Fundamentalmente, el método consiste en partir de un vector inicial $\lambda^{\circ} \geqq 0$ (que puede ser el vector nulo), para determinar una sucesión $\{\lambda^{\nu}\}$ de acuerdo con la regla siguiente:

$$\lambda^{\nu+1} = \text{Max } \{\lambda^{\nu} + \theta^{\nu}(b - Ax^{\nu}) \ , \ 0\}$$

donde el máximo se aplica a cada una de las componentes del vector λ , θ^{ν} es un escalar positivo, y x^{ν} representa una solución óptima de $PR(x,\lambda^{\nu})$.

El resultado básico sobre este método viene expresado por el siguiente Teorema.

Teorema 1.8

$$v(PR(x,\lambda^{\nu})) \longrightarrow v(D) \quad \text{si } \theta^{\nu} \to 0 \quad \text{y} \quad \sum_{i=0}^{\infty} \theta^{i} \to \infty \ .$$

El valor de θ^{ν} que se utiliza generalmente es:

$$\theta^{\nu} = \frac{t_{\nu} \ (z^* - v(PR(x,\lambda^{\nu})))}{\|Ax^{\nu} - b\|^2}$$

donde t_{ν} es un escalar tal que $0 < t_{\nu} \leqslant 2$, que corrientemente se denomina "paso del subgradiente", $\| \ \|$ representa la norma euclídea, mientras que z^* es una cota superior del valor v(D), obtenida frecuentemente aplicando un procedimiento heurístico, que proporcione una solución posible de P. La justificación de este resultado se encuentra en [36] .

La sucesión de valores $v(PR(x,\lambda^{\nu}))$ no es, sin embargo, necesariamente monótona y por lo tanto el método termina frecuentemente sin alcanzar el valor v(D) después de un número arbitrario de iteraciones. A menudo, la sucesión de "pasos" $\{t_{\nu}\}$ se determina tomando $t_o = 2$, y dividiendo dicho valor por 2, cada vez que $v(PR(x,\lambda^{\nu}))$ no aumentado en un número prefijado de iteraciones.

S E C C I O N 2

ESTADO ACTUAL DEL PROBLEMA. PROBLEMAS RELACIONADOS

2.1 CIRCUITOS EULERIANOS OPTIMOS. PROBLEMAS RELACIONADOS

Con el nombre genérico de problemas de cartero, hacemos alusión a una colección de problemas de Optimización Combinatorial, cuyo objetivo consiste en recorrer de forma óptima (i.e: con el mínimo coste) todas, o solamente una parte de las líneas de un grafo con costes asociados. Cuando el grafo no es euleriano (ver Teoremas 1.2,1.3,1.4), el problema en cualquiera de los casos no es trivial, y los distintos problemas de cartero surgen de la separación entre los casos en que el grafo sea no dirigido, dirigido, o mixto, o bien si se han de recorrer todas, o únicamente un subconjunto de líneas requeridas.

El problema de recorrer todas las aristas de un grafo no dirigido, conexo y con costes no negativos asociados, de forma óptima, se conoce por el nombre del problema del cartero chino (CPP), en honor al primer trabajo sobre el tema,que apareció publicado en un periódico chino, firmado por Kwan Mei-Ko [43] . El problema anterior cuando se plantea sobre un grafo dirigido y fuertemente conexo,se representa abreviadamente por (DCPP),es más fácil de resolver que el (CPP) usando algoritmos de flujos, que son de hecho más eficientes que los algoritmos de 1-matching propuestos por Edmonds y Johnson en [18] para resolver el caso no dirigido.

El problema del cartero chino en un grafo mixto (MCPP), es claramente una extensión de el (CPP) y (DCPP) en tanto que un grafo no dirigido, o dirigido, puede considerarse como un caso particular de grafo mixto en el que, o bien el conjunto de arcos, o el de aristas,es vacío. El (MCPP) solamente se puede reducir al (DCPP),sobre un grafo transformado, cuando el grafo mixto original es par (ver Minieka (1978), pp.251-258), -en tanto que para el caso general en que el grafo no sea par, Papadimi-

triou [57] ha demostrado que el (MCPP) es un problema perteneciente a la clase NP-completo, lo cual significa que si se encontrase un algoritmo acotado polinomialmente para resolver el (MCPP), un gran número de problemas cuyo tiempo de resolución crece exponencialmente en función del tamaño de ellos, podría resolverse también en tiempo polinomial, lo cual indicaría que las clases P y NP son en realidad una sola y este hecho es muy improbable que suceda.

Otros problemas de complejidad similar a la del (MCPP), surgen, como ya hemos dicho, al obligar únicamente a un subconjunto de aristas (en el caso no dirigido), o de arcos (en el caso dirigido) a ser recorridas por el tour solución, y se les llaman problemas del cartero rural, en el caso no dirigido (RPP) y dirigido (DRPP); ambos problemas han sido estudiados por A.Corberán [12] y E.Mota [54] respectivamente, proporcionando algoritmos de Branch and Bound utilizando la Relajación Lagrangiana para producir cotas inferiores en los nudos del árbol de ramificación. Orloff [55], considera inmersos a todos estos problemas en lo que él califica problemas generales de rutas (GRP) proponiendo simplificaciones que son incorrectas.

La confección de algoritmos exactos para resolver los problemas de cartero y ,en especial, para el (MCPP) supone una herramienta muy útil para abordar problemas que surgen con frecuencia en la vida real, tales como: recogida de basura, reparto de correo, inspección de líneas de ferrocarril o de alumbrado, limpieza y reparación de gaseoductos y oleoductos,etc..., de los que una mayoría de ellos son resueltos de una forma heurística con el consiguiente detrimento económico.

Una prueba de la importancia de estos problemas y de la atención cada vez mayor que están recibiendo, es el International Workshop on the Routing and Scheduling of Vehicles and Crews, que se celebró

en 1979 en la Universidad de Maryland (U.S.A), cuyos trabajos han sido publicados en un número especial de la revista Networks en el volumen del año 1981. Beltrami y Bodín (1974), por ejemplo, consideran algunos problemas relativos a la recogida de basura y desarrollan procedimientos heurísticos que fueron aplicados en las ciudades de Nueva York y Washington; Male y Liebman (1978) describen también un esquema de rutas para la recogida de basura que fue aplicado en la ciudad de Knoxville (Tenn.U.S.A); Stern y Dror (1979) elaboran un algoritmo para el problema, relacionado con el (CPP) múltiple (i.e: varios circuitos deben recorrer todas las aristas del grafo suponiendo la existencia de unas capacidades máximas), para el diseño de rutas de los lectores de contadores de luz, aplicándolo a la ciudad de Beersheva (Israel), obteniendo reducciones de hasta el 40% en el número de rutas de trabajo.

La mayoría de estos trabajos se ha realizado simplificando la estructura del plano del área de optimización, ignorando en ocasiones las direcciones de las calles a recorrer; por ejemplo, en el caso de la recogida de basura con vehículos, las calles estrechas,en donde la basura se recoge en ambos lados de la calle simultáneamente, corresponderían a aristas en un grafo mixto, combinándola con la existencia de arcos, representativos de calles de una dirección, que deben ser recorridos en un solo sentido, o dos arcos en direcciones opuestas representando avenidas que deben recorrerse en ambos sentidos (no necesariamente con el mismo coste para ambos).

La obtención de algoritmos exactos para resolver el problema del cartero chino en un grafo mixto proporcionarán soluciones más fieles a la realidad que las que proporcionan los algoritmos para los casos dirigido y no dirigido.

2.2 ANTECEDENTES

El (MCPP) puede considerarse resuelto únicamente cuando el grafo mixto sobre el que se pretende encontrar la solución es par, el algoritmo que resuelve este caso particular fue elaborado por Edmonds y Johnson (1973) , más tarde perfeccionado en [22] , se basa en la resolución de un problema de flujos sobre un grafo transformado. Para resolver el (MCPP) en el caso más general, se han elaborado solamente heurísticos y propuesto dos métodos basados en problemas de flujos con ganancias, y examinando todos los puntos extremos de un poliedro, respectivamente; ambas vías de resolución han sido calificadas por sus propios autores de " no elegante" y "computacionalmente intratable en el peor de los casos e ineficiente en el mejor". En consecuencia no se conoce ningún algoritmo específico y exacto que resuelva el (MCPP). Por esta razón, el presente trabajo resulta novedoso al ofrecer un algoritmo exacto para resolver éste problema en su caso más general, acompañado de una serie de resultados computacionales que dan la potencia del mismo para resolver problemas de tamaño real.

En las tres subsecciones siguientes se comentan someramente los resultados más relevantes obtenidos del estudio del (MCPP), recogidos fundamentalmente en tres artículos debidos a Frederickson, Minieka y Kappauf y Koehler , publicados todos ellos en 1979.

2.2.1 Algoritmos heurísticos para el (MCPP)

Algoritmo de Edmonds y Johnson modificado. MIXED1

Este algoritmo consiste esencialmente en tres etapas, que son descritas por medio de subrutinas de la forma siguiente:

Algoritmo MIXED1

Entrada: Un grafo mixto G=(N,A,L) con costes no negativos.

Salida: Un circuito que recorre todas las líneas de G.

1. Call Evendegree
2. Call Inoutdegree (con la salida de Evendegree)
3. Call Evenparity (con la salida de Inoutdegree)

El algoritmo Evendegree realiza un 1-acoplamiento perfecto de mínimo coste entre todos los vértices impares del grafo G, utilizando -- las distancias de los caminos más cortos entre ellos, ignorando las direc-- ciones de los arcos. El acoplamiento proporciona un conjunto A' de arcos y un conjunto L' de aristas que son añadidas al grafo G, obteniendo un grafo mixto aumentado que es par.

El algoritmo Inoutdegree aplicado a un grafo mixto cualquiera G (y, en particular, al grafo resultante de la aplicación del algo-- ritmo Evendegree), produce un conjunto M que contiene algunas copias de los arcos de G, junto con algunas aristas orientadas, de forma que el grafo resultante de añadir M a G tiene todos sus vértices simétricos.

El algoritmo Inoutdegree fue creado originalmente por Ed-- monds y Johnson en 1973 para resolver el (MCPP) cuando el grafo mixto ori-- ginal es par, consiguiendo la simetría de todos sus vértices mediante un -- flujo de mínimo coste, no obstante, este algoritmo puede no conservar la -- paridad de los vértices producida por el algoritmo Evendegree, sin obtener, por lo tanto, un grafo par y simétrico del que se pueda conseguir una solu-

ción posible al (MCPP). Lo único que demuestran Edmonds y Johnson es la -- existencia de un flujo de mínimo coste que conserva la propiedad de que cada vértice tenga grado par. Para resolver este último problema, el algoritmo Evenparity transforma el grafo resultante de añadir al grafo G el conjunto M (formado por arcos adicionales y aristas orientadas), formando caminos que alternativamente tienen sus líneas en M y U (conjunto de aristas que no han sido orientadas), entre los vértices impares del grafo aumentado. Estos caminos son transformados en circuitos, añadiéndoles la arista de U que los une, y finalmente asignándoles una dirección conveniente, eliminando aquellas copias de arcos, o aristas dirigidas en sentido opuesto a la -- del circuito, y orientando adecuadamente (i.e: en el sentido del circuito) a las aristas de U en él. El resultado final es un grafo dirigido euleriano que contiene a todos los arcos y aristas (con una determinada dirección) del grafo original. En este grafo euleriano es sencillo construir un circuito euleriano por medio del algoritmo de Aardenne-Ehrenfest-Bruin que puede ser estudiado en [18].

En la Figura 2.1.a se presenta un grafo mixto original con costes asociados que es par, por lo que el algoritmo Evendegree no lo modifica. El algoritmo Inoutdegree produce el grafo simétrico pero no par de la Figura 2.1.b. En la Figura 2.1.c se representa un circuito, con líneas en M y U alternativamente, uniendo los vértices impares 1 y 4, al que se le asigna la dirección indicada por las flechas. El algoritmo Evenparity elimina una de las copias del arco (3,1) y duplica el arco (2,4) produciendo el grafo euleriano de la Figura 2.1.d obteniendo un circuito euleriano dado por la sucesión: (1,2),(2,4),(4,3),(3,1),(1,2),(2,4),(4,3),(3,2),(2,4),(4,1).

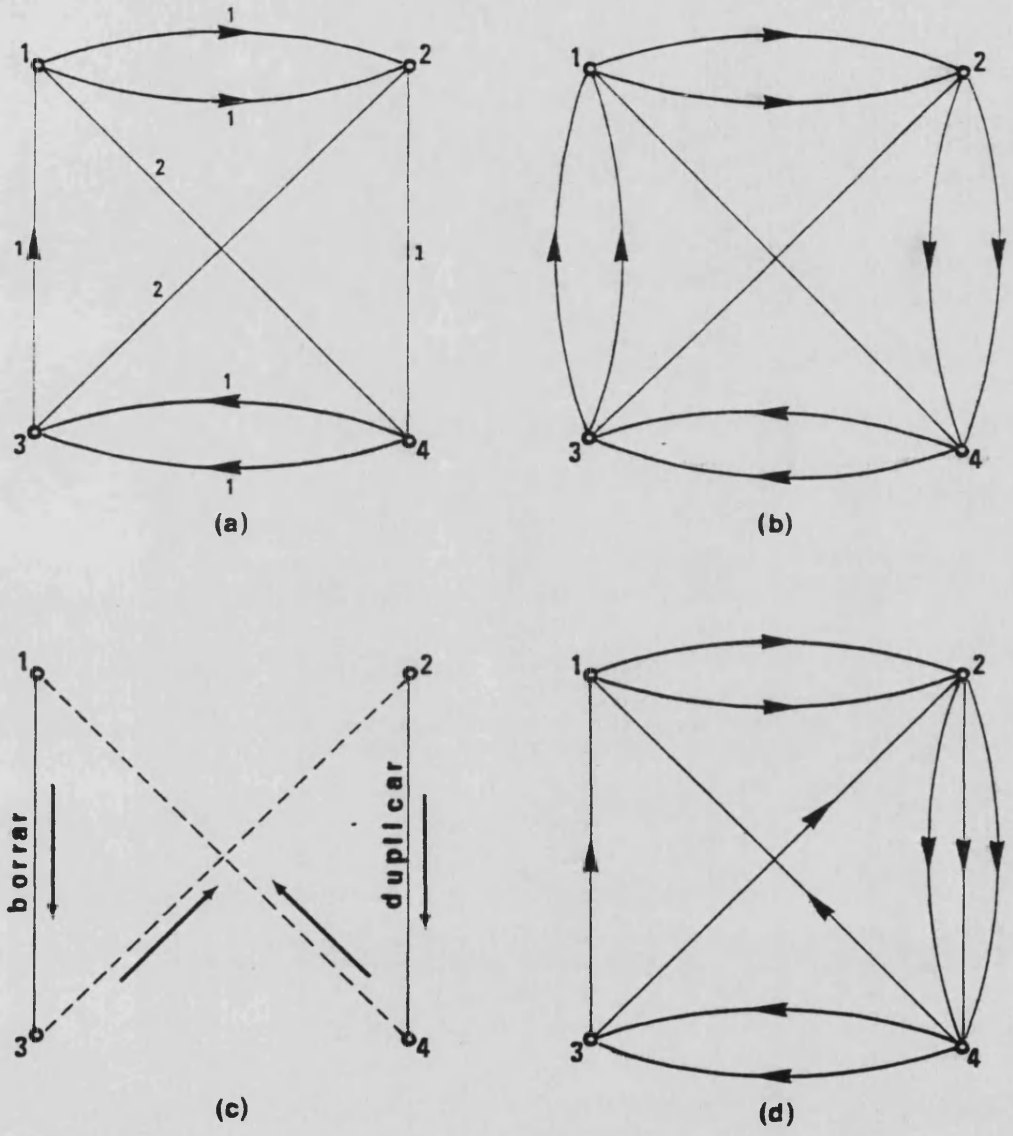


Figura 2.1: Algoritmo MIXED1 aplicado a un grafo.

Algoritmo de Frederickson.MIXED2

Este algoritmo consiste esencialmente en alterar el orden de las dos etapas principales del algoritmo MIXED1. En aquél se construía primero un grafo par y a continuación se le hacía simétrico, mientras que en el MIXED2 se invierte este orden.

Algoritmo MIXED2

Entrada: Un grafo mixto G=(N,A,L) con costes no negativos.

Salida: Un circuito que recorre todas las aristas de G.

1. Call Inoutdegree
2. Call Largecycles (con la salida de Inoutdegree)

El algoritmo Largecycles trabaja con el grafo G=(N,A,L) junto con los conjuntos M y U proporcionados por el algoritmo Inoutdegree. Sobre los vértices de grado impar en el grafo no dirigido G' = (N,U) se realiza un 1-acoplamiento de coste mínimo con las distancias calculadas en el grafo G" = (N,L); las aristas que forman el acoplamiento son añadidas a U obteniendo un grafo par y simétrico donde es fácil obtener una dirección a las aristas no dirigidas por un algoritmo descrito en [18].

Análisis de los peores casos en MIXED1 y MIXED2

En [22] se demuestra que las soluciones obtenidas por los algoritmos MIXED1 y MIXED2, no pueden tener un coste superior a dos veces el coste de la solución óptima y que esta cota superior no puede reducirse. Para apoyar la última aserción se ofrecen ejemplos de mal comportamiento, así, en la Figura 2.2.a se representa el grafo original, en la Figura 2.2.b la solución obtenida por el algoritmo MIXED1, y en la Figura 2.2.c tenemos la solución óptima, siendo ε un valor positivo tan pequeño como queramos.

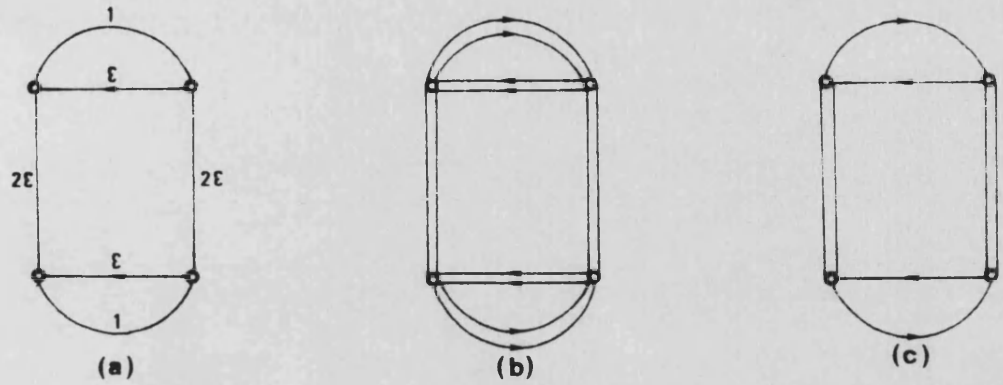


Figura 2.2: Ejemplo del peor caso de MIXED1

Obsérvese que si ε tiende a cero, el coste de la solución producida por MIXED1 tiende a ser dos veces la de la óptima.

El peor comportamiento de MIXED2 puede observarse en el ejemplo mostrado en la Figura 2.3.

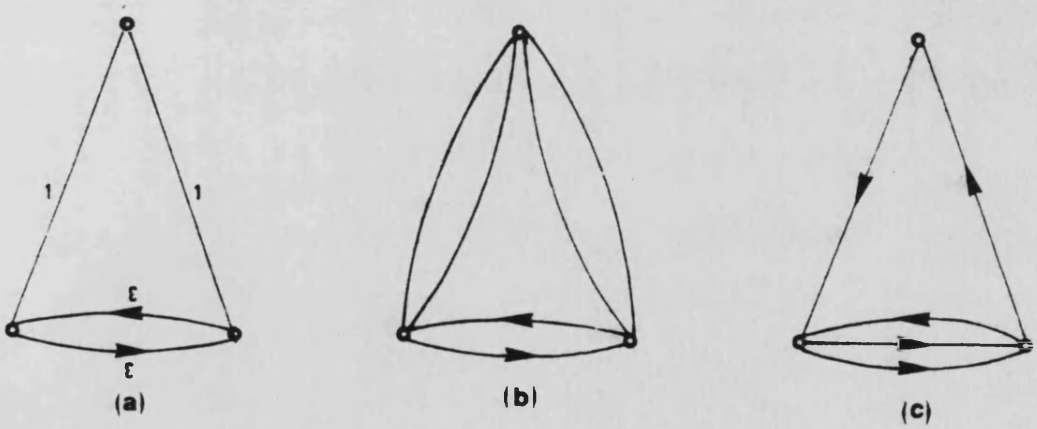


Figura 2.3: Ejemplo del peor caso de MIXED2

En la Figura 2.3.a se muestra el grafo original, en la Figura 2.3.b está la solución proporcionada por MIXED2, y en la Figura 2.3.c la solución óptima. De nuevo, cuando ε tiende a cero, el coste de la solución heurística tiende a ser dos veces la de la óptima.

A raiz de estos ejemplos, es curioso que para el ejemplo donde se da el peor caso de MIXED1, el algoritmo MIXED2 proporciona el valor óptimo y viceversa, si bien, este hecho no siempre ocurre.

Se demuestra en [22] que la mejor solución obtenida de entre las dos producidas por ambos algoritmos, tiene un coste no superior a 5/3 del coste de la solución óptima. Frederickson finalmente propone la utilización de una "distancia más informada" para realizar el acoplamiento inmerso en el algoritmo MIXED2, esta distancia, sin embargo, no es utilizable puesto que puede producir soluciones no posibles según se deduce del contraejemplo mostrado en la Figura 2.4.

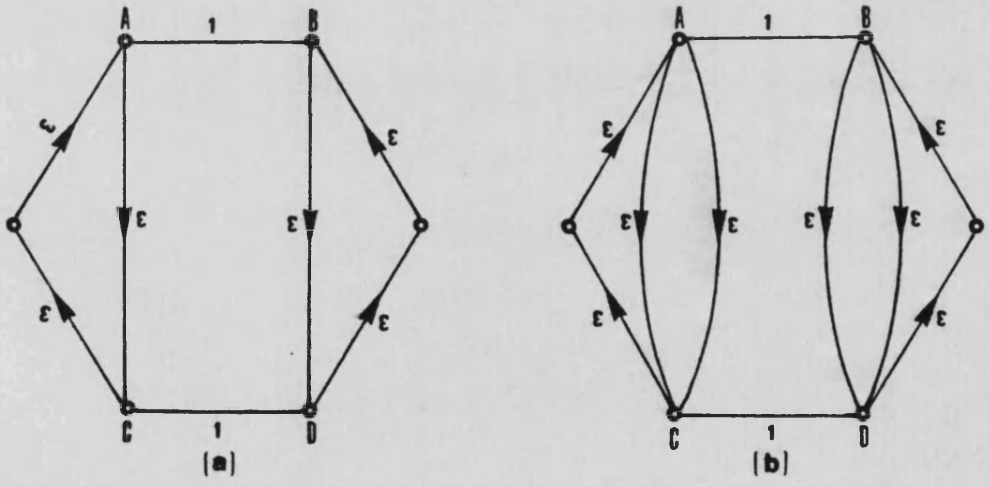


Figura 2.4: Contraejemplo a la "distancia más informada" de Frederickson

En la Figura 2.4.a, se muestra el grafo original, mientras que aplicando la distancia c' definida por:

$$c'(v,w) = \min \left\{ c(v,w),\ \max \{ c.m.c(v,w), c.m.c(w,v) \} \right\}$$

donde c.m.c(v,w) es el coste del camino más corto de v a w considerando todas las líneas del grafo, y c(v,w) representa el coste del camino más corto de v a w, considerando únicamente aquellos caminos formados sólo por aristas, si un camino tal no existe,se supone que su coste es ∞.En nuestro ejemplo, el acoplamiento perfecto de mínimo coste entre los vértices impares A,B,C,D del grafo de la Figura 2.4.a, está formado por los caminos más cortos que conectan A con C y B con D, en este orden, con un coste mínimo total de 2ε, obteniendo el grafo de la Figura 2.4.b que no es euleriano (notar que la condición (c) del Teorema 1.2 no se cumple). En consecuencia el algoritmo MIXED2 no produce una solución posible al (MCPP) en este ejemplo.

2.2.2 El (MCPP) como un problema de flujos con ganancias

En 1979, Minieka propone un algoritmo para resolver el (MCPP) cuya dificultad principal estriba en resolver un problema de flujos con ganancias, en el que la solución debe ser entera. El algoritmo puede resumirse en las siguientes etapas:

Etapa 1. Asignar direcciones arbitrarias a cada arista del grafo mixto original. Para cada vértice i, determinar $d_e(i)$ y $d_s(i)$, pero no contabilizar ninguna arista en el cálculo de $d_e(i)$. A continuación calcular, para cada vértice i, $D(i) = d_e(i) - d_s(i)$ que representa la demanda del vértice i. Si D(i) es positiva, el vértice i es una fuente, mientras que si es negativa, i es un sumidero.

Etapa 2. Por cada arista {i,j}, agrandar el grafo original añadiendo dos vértices más y cinco arcos, con los tripletes asociados (coste,capacidad,ganancia), tal como se representa en la Figura 2.5. Los demás arcos del grafo tienen su coste original, capacidad infinita y ganancia +1. Cada nuevo vértice s es una fuente con una oferta de una unidad.

Etapa 3. Hallar un flujo de coste mínimo "con valores enteros", en el que todas las fuentes descarguen sus ofertas satisfaciendo las demandas de los sumideros.

Etapa 4. (Obtención de la solución)

(a) Si un arco original es recorrido por unidades de flujo, el tour solución debe recorrer dicho arco una vez por cada unidad de flujo que haya circulado por él.

(b) Si la unidad de flujo de la fuente s asociada a una arista {i,j} recorre (s,j), entonces la dirección arbitraria dada a {i,j} no se cambia. Si recorre el arco (s,v), entonces la dirección arbitraria de {i,j} se cambia. Por último, si la unidad de flujo recorre el arco (s,i), entonces se recorre la arista {i,j} una vez en cada dirección.

(coste, capacidad, ganancia)

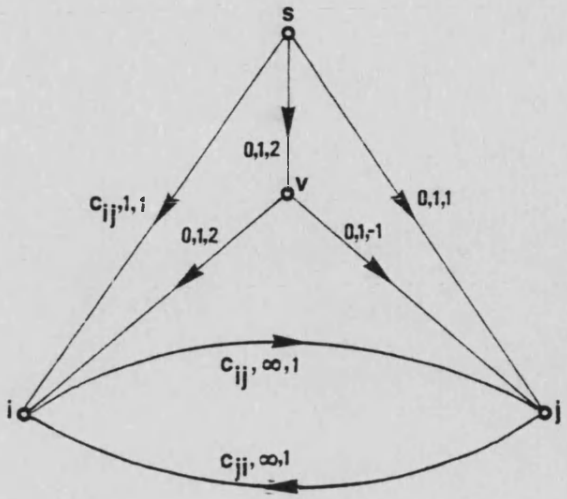


Figura 2.5: Agrandamiento del grafo por cada arista {i,j}.

Algoritmos de flujos con ganancias, han sido desarrollados por Jewell (1962) y Maurrás (1972) en [39] y [50], respectivamente. Desafortunadamente, ni el algoritmo de Jewell, ni ningún otro diseñado para el problema de redes generalizado (que es un caso más general que el nuestro), garantizan que el flujo óptimo sea entero (ver [27] y [19]). La resolución del (MCPP) utilizando este método, supondría la resolución de un problema de programación lineal con restricciones de integridad sobre sus variables, que pueden ser impuestas utilizando técnicas de planos de corte.

La solución propuesta por Minieka, es viable, pero poco esperanzadora para resolver el (MCPP), añadiendo a la dificultad del problema, la creación de cinco nuevos arcos y dos nuevos vértices por cada arista original.

2.2.3 El (MCPP) como un problema de examen exhaustivo de los puntos extremos de un poliedro

Kappauf y Koehler ofrecen una formulación al (MCPP) como un problema de programación lineal entera equivalente a la que se presentará en la Sección 3 de esta memoria. Como ya hemos visto anteriormente, para encontrar la solución óptima al (MCPP) es suficiente con encontrar un grafo dirigido que contenga todas las aristas de G al menos una vez, así como todos los arcos de G con posibles copias, por lo tanto tenemos la siguiente definición:

Definición 2.1

Dado un grafo mixto G = (N,A,L), se dice que el grafo G' = (N,A') es un grafo euleriano asignado de G sii:

(i) G' es euleriano.

(ii) A' está formado por todos los arcos de A junto con los arcos resultantes de asignar direcciones a las aristas de L, con posibles repeticiones.

(iii) Cada arista de L tiene al menos una copia como arco en A'.

Un tour solución al (MCPP) puede construirse a partir de un grafo euleriano asignado cualquiera, por lo que el procedimiento general de solución del (MCPP) puede dividirse en dos etapas:

Etapa 1. Determinar: (a) el número de copias de cada arista de G, (b) la apropiada dirección asignada de las copias necesarias para construir un grafo euleriano asignado de mínimo coste.

Etapa 2. Encontrar un circuito euleriano sobre el grafo asignado y a partir de él un tour de G.

Para resolver la Etapa 2 existen algoritmos sencillos que pueden encontrar

se en [18] de los que ya hicimos mención en la Subsección 2.2.1. Por otra parte, el problema de encontrar un grafo euleriano asignado de mínimo coste puede formularse como un problema de P.L.E. Remitimos a [40] los detalles de esta formulación.

El resultado más importante de [40], es el de establecer una correspondencia biunívoca entre los puntos extremos del poliedro de soluciones posibles, obtenido al eliminar las condiciones de integridad de las variables del problema lineal entero, y una clase especial de grafos asignados llamados principales. De esta forma redondeando al valor de 1 las coordenadas no enteras de los puntos extremos, que se demuestra sólo pueden valer 0, 1/2, ó un entero positivo, se puede obtener una solución posible al (MCPP) (en la Sección 4 se verá cómo). Este último hecho, unido al de que algún grafo euleriano asignado principal es óptimo, permite resolver el (MCPP) examinando todos los puntos extremos de un poliedro.

El resultado comentado en el párrafo anterior tiene más importancia teórica que práctica, pues el analizar todos los puntos extremos de un poliedro puede ser ineficiente en el mejor de los casos e intratable computacionalmente en el mejor, utilizando las palabras de Kappauf y Koehler; además, dado que, el (MCPP) presenta una degeneracion grande, el problema de pasar de un punto extremo a otro resulta una "dura tarea" pues un punto extremo puede tener muchas bases asociadas.

S E C C I O N 3

FORMULACION Y PROPIEDADES DEL (MCPP)

3.1 DESCRIPCION Y FORMULACION DEL (MCPP)

Dado un grafo mixto G = (N,A,L) fuertemente conexo, donde N es un conjunto de n vértices, A es un conjunto de m arcos, y L un conjunto de p aristas. Supondremos, sin pérdida de generalidad (véase pg.14), que G es simple, lo que nos permitirá representar a cada arco $a \in A$ dirigido del vértice i al j, por el par ordenado (i,j), y a cada arista $l \in L$ cuyos vértices terminales sean el i y el j, por el par no ordenado {i,j}. Supondremos además que cada arco y cada arista del grafo G tiene asociado un coste no negativo c_{ij}. El (MCPP) consiste en hallar un tour de G de coste mínimo.

La existencia de un circuito euleriano en un grafo mixto,fue caracterizada en el Teorema 1.2 debido a Ford y Fulkerson en 1962. Si un circuito tal, existe en G, entonces éste es la solución óptima al (MCPP). Si por el contrario G no es euleriano, entonces cualquier tour de G, requerirá la repetición de algunas de las líneas de G. Como ya se ha visto en la Subsección 2.2.3, el (MCPP) puede resolverse encontrando un grafo euleriano asignado de G (ver Definición 2.1) de mínimo coste, y a partir de él construir un tour sobre G por medio de una regla sencilla dada en [18].

Notación 3.1

$A^s(i)$ representa el conjunto de todos los vértices finales de los arcos que 'salen' del vértice i.

$A^e(i)$ representa el conjunto de todos los vértices iniciales de los arcos que 'entran' en el vértice i.

L(i) representa el conjunto de todos los vértices que están conectados al vértice i por medio de una arista.

El problema de encontrar un grafo euleriano asignado de G de coste mínimo puede considerarse equivalente al de resolver el (MCPP), por lo que el (MCPP) puede ser formulado como el problema de minimización lineal entera siguiente:

$$(P) \quad \text{Min} \left\{ \sum_{A} c_{ij}x_{ij} + \sum_{L} c_{ij}(y_{ij} + y_{ji}) + \sum_{A} c_{ij} \right\} \qquad (0)$$

sujeto a:

$$\sum_{A^s(i)} (1+x_{ij}) + \sum_{L(i)} y_{ij} = \sum_{A^e(i)} (1+x_{ji}) + \sum_{L(i)} y_{ji} \qquad \forall i \quad (1)$$

$$y_{ij} + y_{ji} \geq 1 \qquad \forall l=\{i,j\} \in L \quad (2)$$

$$x_{ij},\ y_{ij},\ y_{ji} \geq 0 \qquad \forall (i,j) \in A, \quad \forall l=\{i,j\} \in L \quad (3)$$

$$x_{ij},\ y_{ij},\ y_{ji} \text{ enteras} \qquad \forall (i,j) \in A, \quad \forall l=\{i,j\} \in L \quad (4)$$

Donde los sumatorios existen solamente para los arcos y aristas existentes en G.

La solución óptima del problema (P), permite construir un grafo dirigido G^a del modo siguiente:

(i) Hacer $A^a = A$, y añadir al conjunto A^a, x_{ij} copias del arco (i,j).

(ii) Por cada $y_{ij} > 0$, añadir y_{ij} arcos dirigidos de i a j al conjunto A^a.

El grafo G^a construido por el proceso anterior, es simétrico en virtud de las restricciones (1), que genéricamente serán llamadas restricciones de simetría. Además, G^a es al menos débilmente conexo por construcción, existiendo en él un arco, como mínimo, por cada una de las líneas de G.

Las restricciones de simetría, pueden transformarse en:

$$\sum_{A^s(i)} x_{ij} + \sum_{L(i)} y_{ij} - \sum_{A^e(i)} x_{ji} - \sum_{L(i)} y_{ji} = D(i) \qquad \forall i \quad (1')$$

y representan el hecho de que la demanda de cada vértice i, expresada por:

$$D(i) = \left|A^e(i)\right| - \left|A^s(i)\right| = d_e^G(i) - d_s^G(i)$$

debe satisfacerse con las duplicaciones de arcos incidentes con él, junto con las apariciones, en uno u otro sentido, de las aristas con vértice terminal en i.

Las variables x_{ij} representan el número de veces extra que su arco asociado debe ser recorrido por el tour solución. Las variables y_{ij}, y_{ji} representan el número de veces que la arista asociada debe recorrerse en el tour solución, en una u otra dirección, respectivamente. Las restricciones (2) que llamaremos restricciones de obligatoriedad, obligan a que se recorra cada una de las aristas del grafo G al menos una vez en alguna, o ambas de sus direcciones posibles.

A continuación vamos a demostrar que G^a es un grafo euleriano asignado de G, para lo cual veamos en primer lugar el siguiente Lema.

Lema

Sea $G^a = (N, A^a)$ el grafo asociado a una solución de P. Sea $S \subset N$, cualquier subconjunto propio de vértices, siendo G_S el subgrafo de G^a inducido por S. Se tiene entonces que:

" El número de arcos que entran en S es igual al número de arcos que salen de S ".

Demostración

Sea $d_s^{G^a}(i)$=grado de salida del vértice i en G^a.

y $d_e^{G^a}(i)$=grado de entrada del vértice i en G^a.

Puesto que G^a es simétrico tenemos que:

$$\sum_{i \in S} d_s^{G^a}(i) = \sum_{i \in S} d_e^{G^a}(i) \qquad (3.1)$$

y por otra parte, en cualquier grafo G_S se sabe que:

$$\sum_{i \in S} d_s^{G_S}(i) = \sum_{i \in S} d_e^{G_S}(i) \qquad (3.2)$$

siendo $d_s^{G_S}(i)$ y $d_e^{G_S}(i)$ los grados de salida y de entrada del vértice i en G_S.

Representando a $\sigma(S)$ y $\partial(S)$ como el número de arcos que salen y entran en S respectivamente, obtenemos:

$$\sum_{i \in S} d_s^{G_S}(i) = \sum_{i \in S} d_s^{G^a}(i) - \sigma(S), \text{ y análogamente}$$

tenemos que:

$$\sum_{i\in S} d_e^{G_S}(i) = \sum_{i\in S} d_e^{G^a}(i) - \partial(S)$$

y teniendo en cuenta la expresión (3.2),considerando las dos últimas ecuaciones resulta que:

$$\sum_{i\in S} d_s^{G^a}(i) - \sigma(S) = \sum_{i\in S} d_e^{G^a}(i) - \partial(S)$$

y finalmente por (3.1), $\sigma(S)=\partial(S)$. ●

Para la demostración anterior solamente hemos utilizado que el grafo G^a era simétrico, ahora tenemos el siguiente Teorema.

Teorema 3.1

G^a es un grafo euleriano asignado de G.

Demostración

Unicamente necesitamos demostrar que G^a es fuertemente conexo en virtud del Teorema 1.4. La demostración procede por reducción al absurdo:

Supongamos que G^a no es fuertemente conexo, y sean $C_1,\ldots,C_k$ las k ($k\geqslant 2$) componentes conexas de G^a. Puesto que G^a es , por construcción, débilmente conexo, dada una componente cualquiera C_i,existe al menos un arco (i_1,j_1) tal que $i_1\varepsilon C_i$ y $j_1\varepsilon C_j$ (con $i\neq j$). Aplicando el Lema anterior a los vértices de la componente C_j, debe existir al menos un arco (j_2,h_2) tal que $j_2\varepsilon C_j$ y $h_2\varepsilon C_h$. Tenemos las siguientes posibilidades:

(a) Si $C_h=C_i$, la componente C_i no es maximal respecto a la propiedad de la conexión fuerte, al estar contenida en la componente conexa inducida por los arcos de $C_i\cup C_j\cup(i_1,j_1)\cup(j_2,h_2)$lo cual es absurdo.

(b) Si $C_h\neq C_i$, repitiendo el razonamiento anterior un número finito de veces (menor o igual que k), volveríamos a conectar la componente C_j con la C_i llegando a un absurdo. ●

Comentario 3.1

Se ha demostrado que una solución del problema P, proporciona un grafo euleriano asignado de G, del que fácilmente se obtiene un tour de G. El coste de dicho tour, viene dado por el valor de la función objetivo (0) y que coincide con el coste de G^a.

La formulación dada para el (MCPP) es fácilmente adaptable a la modificación propuesta por Minieka [53], en donde c_{ij} representa el coste de recorrer la arista $\{i,j\}$ en el sentido de i a j, mientras que c_{ji} representa el coste de recorrerla de j a i, pudiendo ocurrir que $c_{ij} \neq c_{ji}$. La sustitución del segundo término de (0) por $\sum_{L(i)} (c_{ij}y_{ij} + c_{ji}y_{ji})$ sería suficiente en este caso, sin embargo, esta suposición invalidaría incluso la técnica de resolución basada en el cálculo de un 1-matching para el (CPP), tal como se ilustra en la Figura 3.1

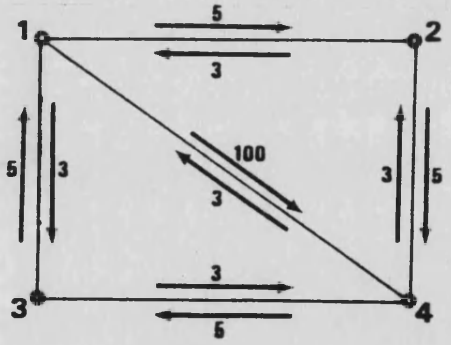


Figura 3.1: Grafo ejemplo con $c_{ij} \neq c_{ji}$.

El acoplamiento de coste mínimo sobre los vértices impares 1 y 4 da como solución la duplicación de la arista $\{1,4\}$, en dirección de 4 a 1. El tour de coste mínimo que pasa por $\{1,4\}$ dos veces viene dado por la sucesión (1,2),(2,4), -(4,1),(1,3),(3,4),(4,1) con coste 22, mientras que el tour óptimo es (1,3),(3,4),(4,1),(1,3),(3,4),(4,2),(2,1) de coste 21. En consecuencia, aunque la formulación dada, sea adaptable a la -

variación propuesta, las consecuencias que conlleva son difíciles de adaptar a los subproblemas que se presentarán en las secciones siguientes de esta memoria, por lo que supondremos que el coste de recorrer una arista en cualquiera de sus sentidos es el mismo.

El Teorema que presentamos a continuación, fue demostrado por Minieka [53] y Kappauf y Koelher [40] simultáneamente. Nosotros hemos querido presentarlo en esta Sección con una fácil demostración.

Teorema 3.2

Si en una solución óptima del (MCPP) la pareja de variables asociadas a una arista cualquiera {i,j} son distintas de cero, entonces ambas toman el valor uno.

Demostración (por reducción al absurdo)
Supongamos que en una solución óptima de P tenemos que $y_{ij}=2$ e $y_{ji}=1$ para alguna arista {i,j}(esta condición es suficiente para nuestros propósitos, puesto que los demás casos son claramente reducibles a éste).

Si hacemos $y_{ij}=1$ e $y_{ji}=0$, esto es equivalente a eliminar una copia del arco (i,j) y otra del arco (j,i) en G^a (grafo asignado a la solución óptima). Con las siguientes posibilidades:

(a) Que $c_{ij}>0$, en cuyo caso el grafo G^* obtenido por la eliminación de las copias de arcos no puede admitir un circuito euleriano (por optimalidad), lo cual es absurdo,pues G^* sigue siendo débilmente conexo, simétrico y con al menos un arco en sustitución de cada arista de G, por lo que en virtud del Teorema 3.1, G^* es un grafo euleriano asignado de G con coste estrictamente menor que G^a.

(b) Que $c_{ij}=0$, en cuyo caso el grafo G^*, construido en el apartado a, proporciona un tour del mismo coste que G^a, esto es, sin pérdida de optimalidad. ●

3.2 ANALISIS DE LA CONECTIVIDAD DEL GRAFO G

En esta Subsección presentamos una serie de resultados que pueden transformar algunas de las aristas del grafo mixto G en arcos, así como la obligación de que algunos de los arcos deben recorrerse un mínimo número de veces en cualquier tour solución al (MCPP). Por otra parte veremos en la Sección 6 que la conectividad inicial de G se modifica durante el proceso de resolución del (MCPP), al particionar el problema P en una colección de subproblemas que resultan al imponer restricciones sobre la dirección en que debe recorrerse cada arista del grafo G, en donde la aplicación de las Proposiciones que se ofrecen a continuación serán de importancia vital. Veamos en primer lugar cuatro definiciones aplicables a cualquier grafo mixto fuertemente conexo y, en particular, al grafo G.

Definición 3.1

Diremos que el par ordenado $(S,\overline{S})$ es un corte que separa a los vértices i,j del grafo $G=(N,A,L)$ sii $i\in S$ y $j\in\overline{S}$, siendo $S\subset N$ y $\overline{S}$ el complementario de S respecto de N.

Definición 3.2

Diremos que el arco $(i,j)\in A$ es de corte en G sii al eliminar (i,j) de G obtenemos un grafo $G'= G-(i,j)$ que no es fuertemente conexo.

Definición 3.3

Diremos que la arista $\{i,j\}$ es de corte en sentido de i a j en G sii al sustituir dicha arista por el arco (j,i) el grafo resultante no es fuertemente conexo.

Definición 3.4

Diremos que la arista $\{i,j\}$ es de corte en G sii es de corte en ambos sentidos.

Las siguientes Proposiciones caracterizan a los arcos de corte y a las aristas de corte en virtud de los cortes que las definen.

Proposición 3.3

Un arco $(i,j)\in A$ es de corte en G sii existe un corte $(S,\overline{S})$ que separa a los vértices i, j de G tal que:

(i) El arco (i,j) es el único dirigido de S a $\overline{S}$.

(ii) No existe ninguna arista entre S y $\overline{S}$.

Demostración

⟹ En efecto, sea G'=G-(i,j) el grafo resultante al eliminar el arco (i,j) de G. En G' construimos el conjunto de vértices alcanzables desde i, R'(i). Por definición sabemos que $i\in R'(i)$, y además $j\notin R'(i)$, pues en caso contrario G' sería fuertemente conexo.

El corte $(R'(i),\overline{R'(i)})$ separa a los vértices i,j;cumpliéndose además que R'(i) es maximal respecto a la propiedad de alcanzabilidad (i.e: no puede existir ningún vértice en $\overline{R'(i)}$ alcanzable desde alguno de R'(i)), con lo que se satisfacen las condiciones (i) e (ii).

⟸ La condición suficiente es obvia, pues si existe un corte que cumpla (i) e (ii), no existe ningún camino en G' de i a j, por lo que (i,j) sería de corte en G. ●

Proposición 3.4

La arista {i,j} es de corte en sentido de i a j en G sii existe un corte $(S,\overline{S})$ que separa a los vértices i, j de G tal que:

(i) No existe ningún arco dirigido de S a $\overline{S}$.

(ii) La arista {i,j} es la única entre S y $\overline{S}$.

Demostración

Sea G'=(N,A',L') el grafo resultante al sustituir la arista {i,j}por el arco (j,i), es decir: $A'=A\cup(j,i)$ y $L'=L-\{i,j\}$, ahora la demostración es análoga a la de la Proposición 3.3 construyendo el corte $(R'(i),\overline{R'(i)})$ en G'. ●

Proposición 3.5

La arista {i,j} es de corte sii existe un corte $(S,\overline{S})$ en G que separa a los vértices i,j tal que {i,j} es la única línea entre S y $\overline{S}$.

Demostración

⟶ Puesto que la arista {i,j} es de corte en ambos sentidos, las condiciones (i) e (ii) de la Proposición 3.4 se cumplen en ambos sentidos, faltando demostrar únicamente que un único corte $(S,\overline{S})$ sirve para ambos casos.

En efecto, sea G'=(N,A',L') siendo A'=A∪(j,i) y L'=L-{i,j}, y sea también el grafo G"=(N,A",L") donde A"=A∪(i,j) y L"=L'. Se tiene que los cortes $(R'(i),\overline{R'(i)})$ y $(R"(j),\overline{R"(j)})$ construidos sobre G' y G" cumplen las condiciones (i) e (ii) de la Proposición 3.3. Demostraremos ahora que ambos cortes que separan obviamente a los vértices i,j son iguales al invertir el orden de uno cualquiera de ellos, es decir que $R'(i)=\overline{R"(j)}$, demostraremos la doble inclusión entre estos dos conjuntos:

(a) $R'(i)\subset\overline{R"(j)}$ (por reducción al absurdo)

Supongamos que $\exists k\in R'(i)\cap R"(j)$, obviamente $k\neq i,j$ y como $k\in R'(i)$ entonces existe un camino P'(i,k) de i a k en G', análogamente existirá un camino P"(j,k) de j a k en G" (ver Figura 3.2)

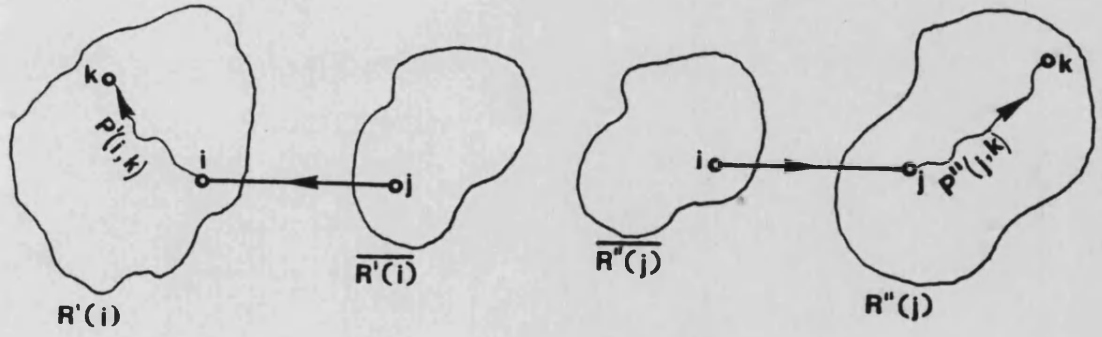


Figura 3.2: Esquema de los grafos G' y G" con los caminos P'(i,k) y P"(j,k).

El camino P'(i,k) no utiliza la arista {i,j}ni en sentido de i a j (por no existir el arco (i,j) en G'),ni en sentido contrario (pues j sería alcanzable desde i en G'). Por idéntico razonamiento P"(j,k) no utiliza la arista {i,j} en ningún sentido. Por otra parte, en el grafo G todo camino P(k,j) debe utilizar la arista {i,j} ya que en otro caso,el camino formado por la unión de los caminos P'(i,k) y P(k,j) sería un camino posible en G' lo cual es absurdo. Análogamente en G todo camino de k a i debe utilizar la arista {i,j} en el sentido de j a i, pues en caso contrario el camino formado por la unión de los caminos P"(j,k) y P(k,i) sería un camino posible en G", lo cual es absurdo.

En el grafo G se han construido los caminos

$$P(k,j)=P'(k,i),(i,j) \quad y \quad P(k,i)=P"(k,j),(j,i)$$

donde P'(k,i) y P"(k,j) son posibles en los subgrafos generados por R'(i) y R"(j) en G' y G" respectivamente. En G se produce la situación esquematizada en la Figura 3.3.

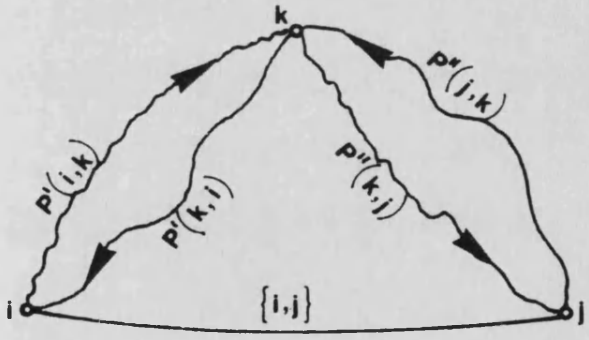


Figura 3.3: Situación grafo G.

Los caminos P(i,j)=P'(i,k),P"(k,j) y P(j,i)= P"(j,k),P'(k,i) existen en G sin utilizar la arista {i,j} en ninguno de sus sentidos, por lo que al eliminarla, el grafo G continuaría siendo fuertemente conexo, lo cual es absurdo.

(b) $R"(j) \subset \overline{R'(i)}$ (por reducción al absurdo)

Supongamos que $h \in \overline{R"(j)} \cap \overline{R'(i)}$, es decir que el vértice h no es alcanzable ni desde j en G", ni desde i en G'. Evidentemente se tiene que $h \neq i,j$ por la definición de conjunto alcanzable. Por no ser alcanzable h desde i en G' entonces

h es alcanzable desde i en el grafo G,solamente a través de un camino P(i,k) que utilice la arista {i,j} en dirección de i a j. Luego P(i,k)= (i,j),P'(j,h), donde P'(j,h) existe en G'. Además por no ser h alcanzable desde j en G", tenemos que podemos construir el camino P(j,h)=(j,i),P"(i,h) donde el camino P"(i,h) existe en G".

Puesto que G es fuertemente conexo, dados los vértices h,i existe el camino P(h,i) de h a i, con las siguientes posibilidades:

1. Si P(h,i) utiliza la arista {i,j} en el sentido de i a j, entonces los caminos P(h,i)=P"(h,j),(j,i) y P"(i,h),P"(h,j) serían posibles en G", lo cual es absurdo (ver Figura 3.4.a).
2. Si P(h,i) no utiliza la arista {i,j} en el sentido de j a i, entonces el camino P'(j,h),P(h,i) sería posible en G', lo cual es absurdo (ver Figura 3.4.b)

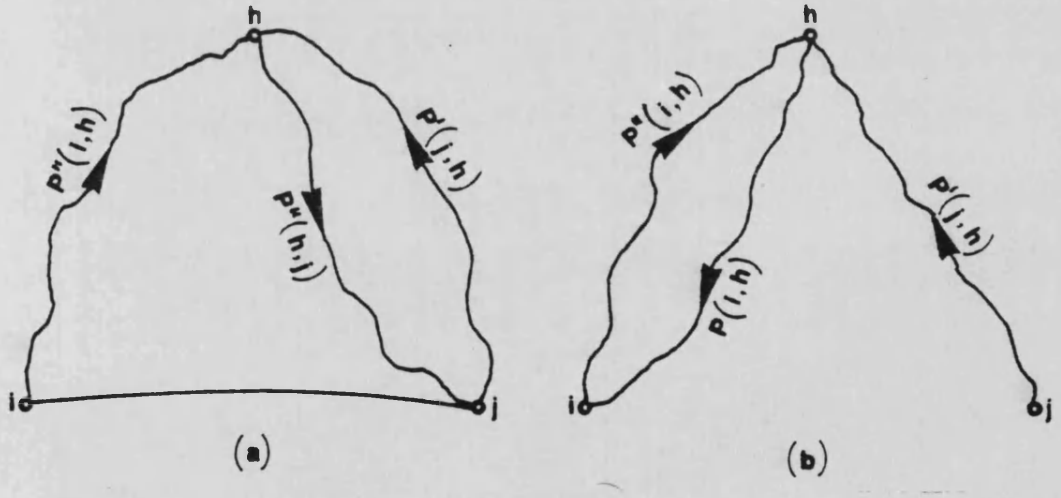


Figura 3.4: (a) Posibilidad 1. para P(h,i), (b) Posibilidad 2.

Por lo tanto hemos demostrado que:

$\overline{R'(i)} \cap \overline{R''(j)} = \emptyset$ que implica que $\overline{R''(j)} \subset R'(i)$

Por las demostraciones de (a) y (b) tenemos que el corte $(S,\overline{S}) = (R'(i), \overline{R'(i)}) = (\overline{R''(j)}, R''(j))$ separa a i y j en el grafo G, existiendo únicamente la arista {i,j} entre S y $\overline{S}$, con lo que finaliza la demostración de la condición necesaria.

⟵ Recíprocamente si en el grafo G existe un corte $(S,\bar{S})$ que separa a los vértices i,j de forma que sólo existe la arista {i,j} entre S y $\bar{S}$, se tiene evidentemente que {i,j} es una arista de corte (i.e: en ambos sentidos).●

Proposición 3.6

Si una arista {i,j} del grafo G es de corte únicamente en un solo sentido entonces no será recorrida en sentido opuesto en cualquier solución óptima del (MCPP).

Demostración (por reducción al absurdo)

Sean y_{ij}, y_{ji} la pareja de variables asociadas a la arista {i,j} que suponemos es de corte únicamente en el sentido de i a j. Supongamos que en una solución óptima $y_{ji}\neq 0$ (negación), y sea G^a el grafo euleriano asignado de G obtenido de ella. Puesto que la arista es de corte en sentido de i a j se tiene que $y_{ij}\geq 1$, y por el Teorema 3.2 sabemos que $y_{ij}=y_{ji}=1$ sin pérdida de optimalidad. Consideremos ahora un corte $(S,\bar{S})$ que caracterice a la arista de corte en sentido de i a j. En virtud de las condiciones (i) e (ii) de la Proposición 3.4, se deduce que el número de arcos que salen del conjunto S en el grafo G^a es estrictamente mayor que el número de arcos que salen de S en G^a, lo cual está en contradicción con el Lema (ver p.41).●

Proposición 3.7

Si una arista {i,j} del grafo G es de corte entonces será recorrida exactamente una vez en cada sentido en cualquier solución óptima del (MCPP).

Demostración

Surge como consecuencia inmediata de la Proposición 3.5 y del Teorema 3.2, observando que la pareja de variables y_{ij}, y_{ji}, asociadas a la arista {i,j} valen uno.●

Comentario 3.2

Los resultados obtenidos en las Proposiciones 3.3 a 3.7 resultantes del estudio de la conectividad del grafo original G, son útiles en tanto que la determinación de aristas de corte en uno, o ambos sentidos, proporciona una disminución del número de restricciones de obligatoriedad (conjunto de restricciones (2)), así como la disminución del número de variables 'y' asociadas a las aristas de G. Particularizando, supongamos que la arista {i,j} es de corte en sentido de i a j, en virtud de la Proposición 3.6 sabemos que se recorrerá siempre en sentido de i a j, en una solución óptima cualquiera del (MCPP), por lo tanto esta arista puede considerarse como un arco (i,j), que será de corte en el grafo resultante de sustituir {i,j} por (i,j). En la formulación de P, este hecho produce la desaparición de la pareja de variables y_{ij}, y_{ji} asociadas a ésta arista, apareciendo una única variable x_{ij} que indicará el número extra de veces que el arco (i,j) será recorrido en la solución óptima al (MCPP), sujeta a las mismas restricciones que cualquier otra variable x asociada a cualquier arco ya existente en el grafo G. Por supuesto, el término constante que figura en la función objetivo se verá incrementado en el coste de este nuevo arco. Por otra parte, supongamos ahora que la arista {i,j} es de corte en ambos sentidos en G, en virtud de la Proposición 3.7 sabemos que dicha arista será recorrida exactamente una vez en cada uno de sus dos sentidos posibles, en cualquier solución óptima al (MCPP), teniendo como consecuencia inmediata la desaparición de las dos variables y_{ij}, y_{ji} asociadas a ella en la formulación de P, y por otro lado, la posibilidad de resolver el (MCPP) por separado en cada uno de los dos subgrafos fuertemente conexos (la demostración está implícita en la demostración de la Proposición 3.4), que genera el corte $(S,\bar{S})$ que separa a los vértices i,j, construido en la demostración de la Proposición 3.4. La importancia de estas consideraciones reside en que la complejidad en tiempo de resolución de P, decrece exponencialmente con la eliminación de la ambigüedad del sentido en que debe ser recorrida cada una de las aristas de G por un tour óptimo (solu-

ción óptima del (MCPP)), puesto que si éste sentido se conociera 'a priori' para toda arista de G, tendríamos que todas las aristas se sustituirían por arcos dirigidos en sentido óptimo reduciendo el (MCPP) al (DCPP) que sabemos se resuelve eficientemente por un algoritmo de flujo de coste mínimo. Veamos ahora la aplicación de los resultados en un ejemplo concreto.

Consideremos el grafo mixto G de la Figura 3.5

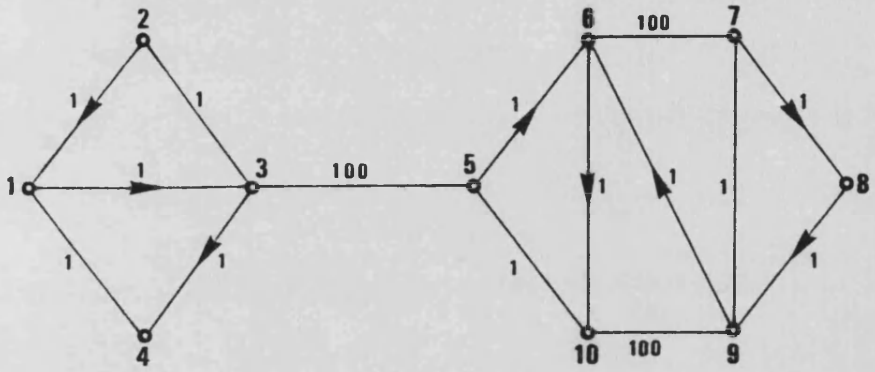


Figura 3.5: Grafo ejemplo G.

El grafo G consta de 7 aristas y 8 arcos, y en pricipio desconocemos el sentido en que será recorrida cada una de sus siete aristas, sin embargo considerando que:

La arista {3,5} es de corte en ambos sentidos, determinado por el corte ({1,2,3,4} ,{5,6,7,8,9,10}), podemos descomponer el grafo G en dos subgrafos G_1 y G_2 que pueden ser optimizados por separado, ver Figura 3.6

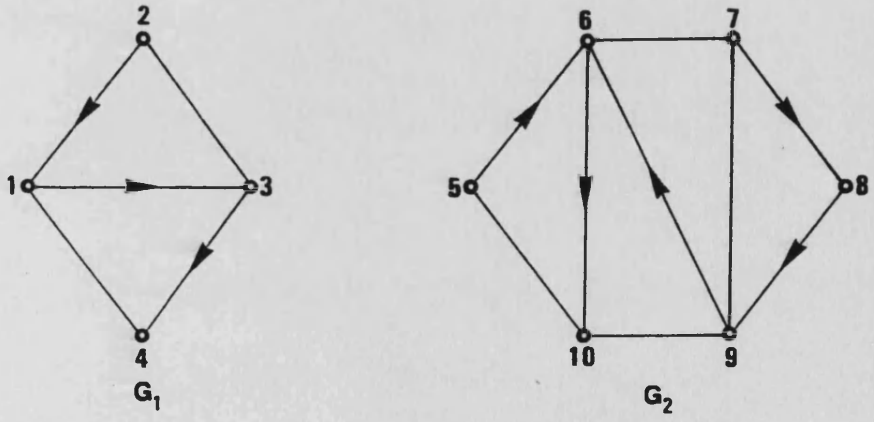
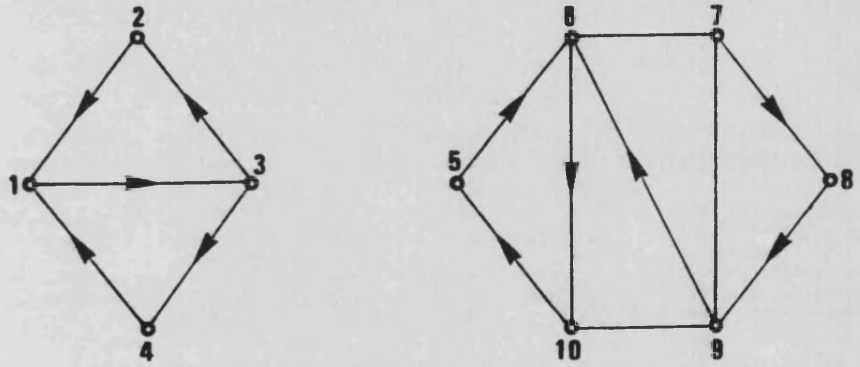


Figura 3.6: Grafos G_1 y G_2.

En el grafo G_1 se observa que la arista {1,4} es de corte en el sentido de 4 a 1, viniendo determinado por el corte ({4} ,{1,2,3}) y análogamente la arista {2,3} sería de corte en el sentido de 3 a 2, con el corte asociado ({1,3,4} , {2}), por lo que ambas aristas pueden transformarse en arcos de corte, dirigidos en en sentido del corte, dando lugar al grafo G_1^T que es dirigido (ver Figura 3.7). Por otra parte, en el grafo G_2 tenemos que la arista {5,10} es de corte en sentido de 10 a 5, definido por el corte ({6,7,8,9,10} ,{5}) por lo que el grafo transformado G_2^T es todavía un grafo mixto.



G_1^T G_2^T

Figura 3.7: Los subgrafos transformados.

Los grafos eulerianos asignados a los subgrafos G_1^T y G_2^T , proporcionan la solución óptima del (MCPP) sobre G, uniendo éstos por la incorporación de los arcos (3,5) y (5,3) correspondientes a la arista de corte {3,5}. Ver Figura 3.8.

Podemos observar también que no son las aristas de corte las únicas que se deben recorrer en ambos sentidos en una solución óptima al (MCPP).

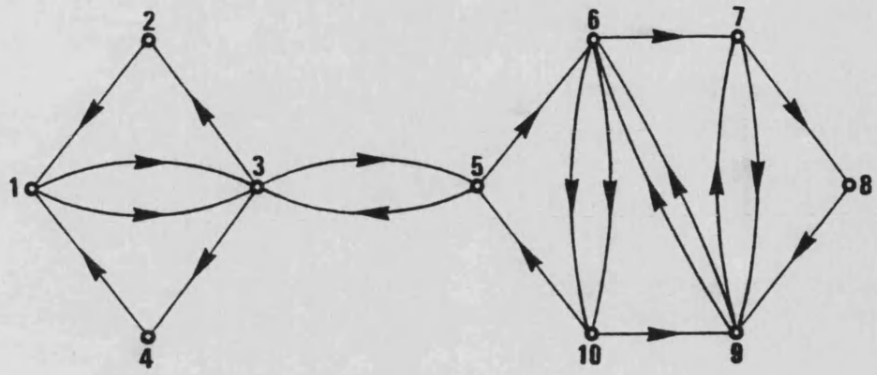


Figura 3.8: La solución óptima al (MCPP) sobre el grafo G.

Podría argumentarse que lo que en principio le -- ocurría al grafo G, es que contenía "falsas aristas" que en -- realidad eran arcos, lo cual no deja de ser cierto, pero con - la importante salvedad de que los resultados de las proposicio nes de 3.3 a 3.7, no solamente pueden ser aplicadas al grafo G, sino a cualquier grafo derivado de él,en el que las direccio-- nes de algunas de sus aristas han sido fijadas, pudiendo origi nar en el grafo derivado, la aparición de nuevas aristas de -- corte que en G no existían. La potencia de estos resultados se verá realzada, cuando en el algoritmo de Branch and Bound, des crito en la Sección 6, sean fijadas las direcciones de las -- aristas de G de acuerdo con la estrategia de separación elegi- da. Por último, los arcos de corte existentes en G (que pueden ser incrementados en número si se determina la existencia de - aristas de corte), tienen variables asociadas x_{ij} a las que se les puede ajustar una cota inferior superior a la trivial de - cero, determinada por el corte que las caracteriza. Este resul tado se justifica en las siguientes proposiciones.

Proposición 3.8

Si el arco (i,j) es de corte en G, entonces su va riable asociada x_{ij} cumple que $x_{ij} \geq \partial(S) - 1$, en cualquier so- lución óptima de (P), siendo S el conjunto de vértices que - define un corte $(S,\overline{S})$ que separe a los vértices i,j cumpliendo las condiciones (i) e (ii) de la Proposición 3.3.

Demostración

Es inmediata en virtud del Lema (ver p.41), y de la Proposición 3.3. ●

Proposición 3.9

(a) Si el arco (i,j) es el único que sale del vértice i y $d_a^G(i)=0$, entonces su variable asociada x_{ij}, cumple la desigualdad: $x_{ij} \geq d_e^G(i)-1$.

(b) Si el arco (i,j) es el único que entra en el vértice j y $d_a^G(j)=0$, entonces su variable asociada x_{ij}, cumple la desigualdad: $x_{ij} \geq d_s^G(j)-1$.

Demostración

Es inmediata, considerando que bajo estas hipótesis, el arco (i,j) es de corte y luego aplicando el resultado de la Proposición 3.8. ●

Comentario 3.3

La existencia de uno o más arcos de corte en G, o bien en un grafo derivado de él por las transformaciones de aristas de corte en arcos, viene determinada por uno o varios cortes $(S,\bar{S})$, que separan a los vértices incidentes con ellos, cumpliendo las condiciones (i) e (ii) de la Proposición 3.3. Esta situación se esquematiza en la Figura 3.9.

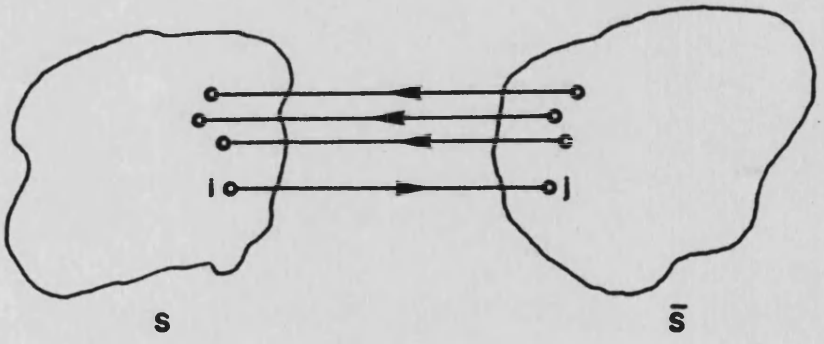


Figura 3.9: Esquema de un arco de corte.

En la Figura 3.9 hemos representado a un arco de corte génerico, incidente con los vértices i y j, junto con - uno de los cortes que lo define. El valor de $\partial(S)$ (recordamos que se definió en la demostración del Lema), representa en -- cierto modo, la "oferta" o número de arcos mínimo que deben - salir del conjunto S para construir un grafo euleriano asigna do de G. Si $\partial(S)$ vale 1, entonces la cota inferior para la va riable x_{ij} seguirá siendo cero, pero si $\partial(S)$ es mayor o igual que dos, entonces la cota inferior será como mínimo de - $\partial(S)-1 \geqslant 1$, incorporando el coste de estas copias al término - constante de la función objetivo de (P), pasando la variable x_{ij} a representar el número adicional de copias (a parte de - las $\partial(S)-1$), que deberán ser colocadas en el grafo euleriano asignado de G de mínimo coste.

3.3 RESTRICCIONES ADICIONALES AL PROBLEMA.

En la formulación del problema P, se ha procurado incluir el mínimo número de restricciones que fueran suficientes para obtener la solución óptima al (MCPP), sin embargo, un conjunto de restricciones que fuera redundante con las restricciones (1),(2),(3) y (4) que definen el conjunto de soluciones posibles del problema P, podrían no serlo en alguna de sus relajaciones, por lo que la inclusión de las mismas supone una reducción del número de soluciones de los problemas relajados, incrementando así el valor de las cotas inferiores que producen.

3.3.1 Restricciones de paridad.

Las restricciones de paridad asociadas a cada uno de los vértices de G hacen referencia a que la paridad de cada vértice en el grafo euleriano asignado de G que proporciona la solución óptima al (MCPP) debe ser par. Este tipo de restricciones viene definido por el siguiente sistema de congruencias en módulo 2:

$$\sum_j (1+x_{ij}) + \sum_j (1+x_{ji}) + \sum_j (y_{ij}+y_{ji}) \equiv 0 \pmod 2 \quad \forall i \quad (3.3)$$

donde los índices j de los sumatorios existen únicamente cuando las variables x_{ij}, y_{ij}, y_{ji} existen con un vértice i fijado, en consecuencia la expresión (3.3) es equivalente a:

$$\sum_j x_{ij} + \sum_j x_{ji} + \sum_j (y_{ij} + y_{ji} - 1) + d^G(i) \equiv 0 \pmod 2 \quad \forall i$$

donde $d^G(i)$ representa el grado del vértice i en G.

Queremos ahora transformar las congruencias en ecuaciones, para lo cual definimos variables binarias b_i de la forma siguiente:

$$\begin{cases} b_i = 0 \text{ si el vértice i es par.} \\ b_i = 1 \text{ si el vértice i es impar.} \end{cases}$$

por la definición dada, tenemos que obviamente $b_i \equiv d^G(i) \pmod 2$ por lo que si $b_i = 0$ para todo vértice i, el grafo G sería par.

Introduciendo las variables enteras y no negativas w_i, asociadas a cada una de las congruencias, obtenemos que:

$$\sum_j x_{ij} + \sum_j x_{ji} + \sum_j (y_{ij}+y_{ji}-1) - 2w_i = b_i \qquad \forall i \qquad (5)$$

$$w_i \geq 0, \text{ enteras} \qquad \forall i \qquad (6)$$

Las restricciones (5) y (6) son equivalentes al sistema de congruencias de la expresión (3.3). Ahora vamos a demostrar que dicho sistema de congruencias es redundante con las restricciones que definen al problema P.

Proposición 3.10

Toda solución posible de P satisface los conjuntos de restricciones (5) y (6)

Demostración

A causa de la equivalencia entre el sistema de congruencias de la expresión (3.3) con el sistema (5) y (6), tenemos que en virtud de las restricciones de simetría (1):

$$\sum_j (1+x_{ij}) + \sum_j y_{ij} + \sum_j (1+x_{ji}) + \sum_j y_{ji} \equiv 0 \pmod 2 \qquad \forall i$$

y puesto que existe un par de variables y_{ij}, y_{ji} asociadas a cada arista $\{i,j\}$ tenemos que:

$$\sum_j y_{ij} + \sum_j y_{ji} = \sum_j (y_{ij} + y_{ji})$$

por lo que las restricciones de simetría son redundantes con (5) y (6). ●

Notar que solamente hemos hecho uso de las restricciones de simetría para probar el resultado anterior. Por otra parte es fácil ver que las condiciones de integridad y no negatividad de las variables x_{ij}, y_{ij}, y_{ji} $\forall ij$, junto con las restricciones de obligatoriedad implican que las variables w_i sean

no negativas, pues si consideramos la expresión

$$2w_i = \sum_j x_{ij} + \sum_j x_{ji} + \sum_j (y_{ij}+y_{ji}-1) - b_i \quad \text{dado } i \qquad (3.4)$$

obtenida de las restricciones (5), tenemos que el segundo miembro de (3.4) solamente puede tomar los valores $\{-1,0,1,2,\ldots\}$ ya que b_i vale 0 ó 1 y cada uno de los sumandos que aparecen en él es un entero no negativo, en virtud de las restricciones (2),(3),(4), pero notemos que el valor -1 no es posible, a causa de la restricción de integridad de la variable w_i.

S E C C I O N 4

COTA SUPERIOR.SOLUCIONES POSIBLES

El algoritmo heurístico que presentamos en esta Sección para resolver el (MCPP), es una alternativa a los ya - presentados en la Sección 2 y proporciona una cota superior al valor óptimo del problema (P), que será utilizada en el algo-- rítmo de Branch and Bound que veremos en la Sección 6. Este -- heurístico proporciona en una primera etapa, una información so bre las posibles aristas de corte en el grafo original, al -- igual que posibles repeticiones de arcos de corte que deben -- aparecer en cualquier solución posible al (MCPP). La primera - etapa podría ser eliminada del heurístico sin variar el valor de la solución posible que proporciona, sin embargo es de fundamental importancia incluida en el algoritmo exacto que pre-- sentaremos puesto que puede disminuir el número de aristas del grafo original, y dicho algoritmo se verá que es de crecimiento exponencial en el número de aristas.

Por otra parte, baste decir que la cota superior producida con la mejora introducida en la Subsección 4.2., ha proporcionado el valor óptimo en doce de los treinta y cinco - problemas testados, con un porcentaje medio de alejamiento del valor óptimo inferior al 3%. Además dicho heurístico es comparado con el implícitamente obtenido de la solución a la relaja ción lineal de (P) en la Subsección 4.3.

4.1 OBTENCION DE UNA COTA SUPERIOR AL (MCPP)

Sea G = (N,A,L) el grafo mixto original. El algoritmo para obtener una cota superior a la solución óptima del (MCPP), es descrito por etapas en la forma siguiente:

Etapa 1: Consiste en encontrar todas las aristas de corte en G, transformando éstas (si existen) en arcos dirigidos. Para comprobar si una arista $\{i,j\}$ es de corte en sentido de i a j, se sustituye la arista $\{i,j\}$ por el arco (j,i) tal como se vió en la Proposición 3.4, y en el grafo resultante se comprueba si $j \in R(i)$; en caso afirmativo, la arista $\{i,j\}$ no es de corte en dicho sentido, mientras que si $j \notin R(i)$ esto significa que $\{i,j\}$ es de corte en ese sentido, y por lo tanto se sustituye por el arco (i,j), en cuyo caso se comprueba si también es de corte en el sentido contrario verificando si $\{i,j\}$ es la única línea entre R(i) y $\overline{R(i)}$. El método anterior se realiza para cada una de las aristas del grafo,utilizando un sencillo algoritmo descrito en Christofides (1975), que sirve para determinar las componentes conexas de un grafo, obteniendo el conjunto R(i). Al finalizar el análisis para todas las aristas del grafo G, se comprueba para cada vértice del grafo si cumple las condiciones de la Proposición 3.9, obteniendo el grafo transformado (si se han encontrado aristas de corte, o determinado la existencia de arcos que deben duplicarse) $G^T = (N,A^T,L^T)$, donde A^T, L^T representan los nuevos conjuntos de arcos y aristas.

Etapa 2: Si $L^T = \phi$ (i.e: todas las aristas eran de corte en alguno o en ambos sentidos), el grafo G^T es dirigido, con lo que aplicando sobre él el algoritmo de flujo de coste mínimo, que resuelve el (DCPP), obtendríamos la solución óptima al (MCPP). Si $L^T \neq \phi$, entonces asignar una dirección cualquiera a cada una de las aristas de L^T, obteniendo un grafo dirigido que llamaremos $G_d = (N,A_d)$, siendo A_d el conjunto de los arcos existentes en A^T, junto con los arcos resultantes de asignar una única dirección a cada una de las aristas de L^T.

Etapa 3: Calcular en G_d los grados de entrada $d_e(i)$, y de salida $d_s(i)$ de cada uno de los vértices i del grafo, clasificándolos de la siguiente forma:

(a) Si $D(i)=d_e(i)-d_s(i) > 0$, entonces i es una fuente con oferta D(i).

(b) Si $D(i)=d_e(i)-d_s(i) < 0$, entonces i es un sumidero con demanda -D(i)

(c) Si $D(i)=d_e(i)-d_s(i) = 0$, entonces i es un vértice intermedio

Etapa 4: Construir el grafo dirigido G'=(N,A') de la forma siguiente:

(a) Por cada arco $(i,j) \in A^T$ poner un arco (i,j) en el conjunto A' con la misma dirección y coste, y su capacidad será infinita.

(b) Por cada arista $\{i,j\} \in L^T$ poner dos arcos en paralelo, uno dirigido de i a j, y el otro en sentido opuesto, en el conjunto A'; cada uno de estos arcos tiene capacidad infinita y coste igual al que tenía la arista {i,j} en el grafo original. Estos arcos serán llamados "arcos en paralelo".

(c) Por cada arista $\{i,j\} \in L^T$ que ahora figura como arco dirigido en G_d, poner un arco dirigido en sentido opuesto en A'. Estos arcos serán llamados "arcos artificiales", y tienen capacidad dos y coste cero.

Etapa 5: Utilizando las ofertas y demandas de cada vértice i obtenidas en la Etapa 3, resolver el problema de flujo de coste mínimo sobre el grafo G', definido por el problema de minimización (F) siguiente:

$$(F) \qquad \text{Min} \sum_{(i,j)\in H_1} c_{ij}\, f(i,j) \qquad (0_F)$$

sujeto a:

$$\sum_j f(i,j) + \sum_j f^g(i,j) - \sum_j f(j,i) - \sum_j f^g(j,i) = D(i) \quad \forall i \quad (1_F)$$

$$\left.\begin{array}{ll} 0 \leqslant f(i,j) & \forall (i,j) \in H_1 \\ 0 \leqslant f^g(i,j) \leqslant 2 & \forall (i,j) \in H_2 \end{array}\right\} \quad (2_F)$$

Los sumatorios existen sobre los arcos del grafo - G', siendo H_1 el conjunto de todos los arcos "no artificiales", mientras que H_2 es el conjunto de todos los arcos "artificiales". El problema de flujo de mínimo coste (F), siempre tiene solución puesto que G' es por construcción un grafo fuertemente conexo.

Sea una solución óptima del problema (F), siendo - $\hat{f}(i,j)$, $\hat{f}^g(i,j)$, los flujos óptimos que circulan por cada arco $(i,j) \in A'$, dependiendo de si el arco (i,j) es no artificial, o artificial, respectivamente.

Etapa 6: A partir de la solución óptima al problema (F), cons-- truir el grafo $\tilde{G}=(N,\tilde{A})$ del siguiente modo:

(a) Si $\hat{f}^g(i,j)=0$, entonces poner en el conjunto $\tilde{A}$ tan-- tas copias del arco (j,i) como indique el valor de $1+\hat{f}(j,i)$, siendo $\hat{f}(j,i)$ el número de unidades de -- flujo que han circulado por el arco "en paralelo" y dirigido en sentido opuesto al del arco artificial (i,j) asociado.

(b) Si $\hat{f}^g(i,j)=2$, entonces poner en el conjunto $\tilde{A}$ tan-- tas copias del arco (i,j) como indique el valor de $1+\hat{f}(i,j)$, siendo $\hat{f}(i,j)$ el número de unidades de -- flujo que han circulado por el arco "en paralelo" y dirigido en el mismo sentido que el del arco artifi cial (i,j) asociado.

(c) Si $\hat{f}^g(i,j)=1$, entonces poner en el conjunto $\tilde{A}$ una - copia del arco (i,j) y otra del arco (j,i). Obsérve se que se puede suponer en este caso que $\hat{f}(i,j)=\hat{f}(j,i)=0$, para los dos arcos en paralelo con el arco artificial (i,j) asociado sin pérdida de optimalidad.

El algoritmo anterior finaliza con la obtención de un grafo $\tilde{G}=(N,\tilde{A})$, que demostraremos es un grafo euleriano asignado de G y por lo tanto proporciona una solución posible al (MCPP). Por otra parte, las Etapas de 2 a 6 pueden considerarse como una extensión del algoritmo propuesto por Minieka (1978), para obtener la solución óptima al (MCPP) cuando el grafo original G es par, en donde los flujos por los arcos artificiales son siempre o cero o un número par. En el caso en que el grafo G no sea par, obtenemos una solución posible como demuestra el siguiente Teorema.

Notación 4.1

Representaremos por:

$d_e(i)$ el grado de entrada del vértice i en G_d

$d_s(i)$ el grado de salida del vértice i en G_d

$\tilde{d}_e(i)$ el grado de entrada del vértice i en $\tilde{G}$

$\tilde{d}_s(i)$ el grado de salida del vértice i en $\tilde{G}$

$D(i)=d_e(i)-d_s(i)$

$\tilde{D}(i)=\tilde{d}_e(i)-\tilde{d}_s(i)$

$\Delta D(i)=\tilde{D}(i)-D(i)$ incremento total de la demanda del vértice i.

Teorema 4.1

El grafo $\tilde{G}$ es un grafo euleriano asignado de G.

Demostración

Por construcción, en $\tilde{G}$ existe al menos un arco, en una dirección, en sustitución de cada una de las aristas de G, por lo que es suficiente con demostrar que G es un grafo simétrico.

En efecto, los incrementos totales, $\Delta D(i)$, en la demanda de los vértices producidos por los cambios en la Etapa 6 del algoritmo, son debidos a la suma de los incrementos parciales que ocasionan en la demanda del vértice i, cada uno de los arcos artificiales incidentes con él.

Para cada arista l incidente en el grafo G con el vértice i, sea $\Delta^l D(i)$ el incremento sufrido por D(i), cuando se considera la dirección del arco artificial asociado en G, en lugar de la dirección que originalmente tenía en G_d. Se tiene por lo tanto que $\Delta D(i) = \sum_l \Delta^l D(i)$, donde el sumatorio se realiza únicamente para las aristas l incidentes con el vértice i en G.

Dada una arista l= {u,v} en el grafo G, podemos distinguir los siguientes casos:

Caso 1: En G_d la arista l figuraba como el arco (u,v) pudiendo ocurrir:

(1.a) Que en $\tilde{G}$ figure el arco (u,v) sii $\hat{f}^g(v,u)=0$
En este caso, los grados de entrada y de salida de los vértices u,v, son los mismos en ambos grafos (considerando solamente este arco artificial), por lo tanto tendremos:
$\Delta D^l(u)=0$ y también $\Delta D^l(v)=0$.

(1.b) Que en $\tilde{G}$ figure el arco (v,u) sii $\hat{f}^g(v,u)=2$
En este caso, tenemos que:
$\Delta D^l(u)=2$ mientras que $\Delta D^l(v)=-2$.

(1.c) Que en $\tilde{G}$ figuren los arcos (u,v), (v,u) sii $\hat{f}^g(v,u)=1$.
En este caso tenemos que:
$\Delta D^l(u)=1$, mientras que $\Delta D^l(v)=-1$.

Caso 2: En G_d la arista l figuraba como el arco (v,u).

En este caso, la dirección inicial en G_d de la arista {u,v} es la opuesta a la del Caso 1, y los incrementos parciales de u y v se intercambian.

Los seis subcasos existentes se resumen en la Tabla 4.1

Asig. en G_d	Dirección en $\tilde{G}$	$\Delta D^1(u)$	Flujo por arco art.
(u,v)	(u,v)	0	$\hat{f}^g(v,u)=0$
	(v,u)	2	$\hat{f}^g(v,u)=2$
	(u,v),(v,u)	1	$\hat{f}^g(v,u)=1$
(v,u)	(v,u)	0	$\hat{f}^g(u,v)=0$
	(u,v)	-2	$\hat{f}^g(u,v)=2$
	(u,v),(v,u)	-1	$\hat{f}^g(u,v)=1$

Tabla 4.1: Cambios en las demandas y relación con los flujos -- por arcos artificiales.

En consecuencia tenemos que:

$$\Delta D(i)=\sum_1 \Delta D^1(i)=\sum_j \hat{f}^g(j,i) - \sum_j \hat{f}^g(i,j) \qquad \forall i \qquad (4.1)$$

En virtud de las ecuaciones (1_F) y la expresión (4.1), obtenemos

$$\tilde{D}(i)=\tilde{d}_e(i)-\tilde{d}_s(i)=D(i)+\sum_j \hat{f}(j,i)-\sum_j \hat{f}(i,j)+\Delta D(i)=0$$

y finalmente $\tilde{d}_e(i)=\tilde{d}_s(i)$ $\forall i$, con lo que el grafo $\tilde{G}$ es simétrico. ●

4.2 MEJORA DE LA COTA SUPERIOR

Aplicando el algoritmo heurístico explicado en la Subsección anterior, obtenemos una cota superior cuyo valor es la suma de los costes de los arcos del grafo euleriano $\tilde{G}$. Esta cota puede ser mejorada,en ocasiones, sustituyendo en $\tilde{G}$ algunos de sus arcos por el camino más corto entre sus pares de vértices incidentes. Recordemos que el problema de flujo (F), resuelto para la obtención de $\tilde{G}$, transforma cada arista $\{i,j\}$ por donde ha circulado una unidad de flujo (esto se realiza con coste cero en la resolución de (F)), en dos arcos dirigidos en sentidos opuestos (i,j), (j,i), cuyo coste en $\tilde{G}$ es el asociado a la arista $\{i,j\}$. El hecho anterior da origen a una primera mejora consistente en sustituir uno de los dos arcos (i,j) ó (j,i) (nunca ambos), por el camino más corto de i a j, ó de j a i respectivamente, siempre que ello represente una disminución del coste de $\tilde{G}$. Por lo tanto, supongamos que c_{ij} es el coste original de la arista $\{i,j\}$, y sean c.m.c(i,j), c.m.c(j,i) los costes de los caminos más cortos de i a j, y de j a i respectivamente. Sean $d_{ij} = c_{ij}$ - c.m.c(i,j) y $d_{ji} = c_{ji}$ - c.m.c(j,i), las diferencias entre el coste de la arista $\{i,j\}$ y los costes de los caminos más cortos entre los vértices terminales de ella. Representaremos por $D_{ij} = d_{ij} - d_{ji}$, teniendo la Proposición siguiente:

Proposición 4.2

(a) Si $D_{ij}>0$, sustituyendo el arco $(i,j)\in\tilde{A}$, por el --- c.m.c(i,j) se obtiene una solución posible mejorada con valor $c(\tilde{G})-d_{ij}$.

(b) Si $D_{ij}<0$, sustituyendo el arco $(j,i)\in\tilde{A}$, por el --- c.m.c(j,i) se obtiene una solución posible mejorada con valor $c(\tilde{G})-d_{ji}$.

(c) Si $D_{ij}=0$, pueden ocurrir dos casos:

(c1) $d_{ij}=d_{ji}=0$, en este caso no puede ser mejorada la solución en la forma descrita y no se efectúan cambios.

(c2) $d_{ij}=d_{ji}>0$, entonces sustituyendo uno de los dos arcos que existen entre los vértices i,j por el camino más corto correspondiente, obtenemos una solución posible mejorada,con valor $c(\tilde{G})-d_{ij}$.

Demostración

Obvia. ●

Las sustituciones que se proponen en la Proposición 4.2,son también aplicables a cualquier camino formado únicamente por aristas (que en G figuran como parejas de arcos dirigidos en sentidos opuestos), por las que ha circulado un flujo de una unidad por el arco artificial correspondiente, no siendo aplicables a otro tipo de arcos y/o caminos, puesto que éstos figuraban en (F) con su coste original y por lo tanto la optimalidad del flujo impide que se sustituyan.

Después de la mejora anterior, únicamente queda una mejora adicional en el caso en que al sustituir arcos y/o caminos por caminos más cortos,dé lugar a la aparición de una arista representada en el grafo transformado de G por tres o más arcos,de los cuales hay dos en direcciones opuestas, con lo que se podrían eliminar pares de repeticiones tal como se estableció en el Teorema 3.2.

La aplicación del algoritmo heurístico descrito en esta Subsección para el cálculo de la cota superior,que más tarde,será utilizada en el algoritmo de Branch and Bound se ilustra con el siguiente ejemplo mostrado en la Figura 4.1,correspondiente al problema número 12 del total de los 35 testados.

Ejemplo de obtención de la cota superior

Consideremos el grafo mixto G, representado en - la Figura 4.1

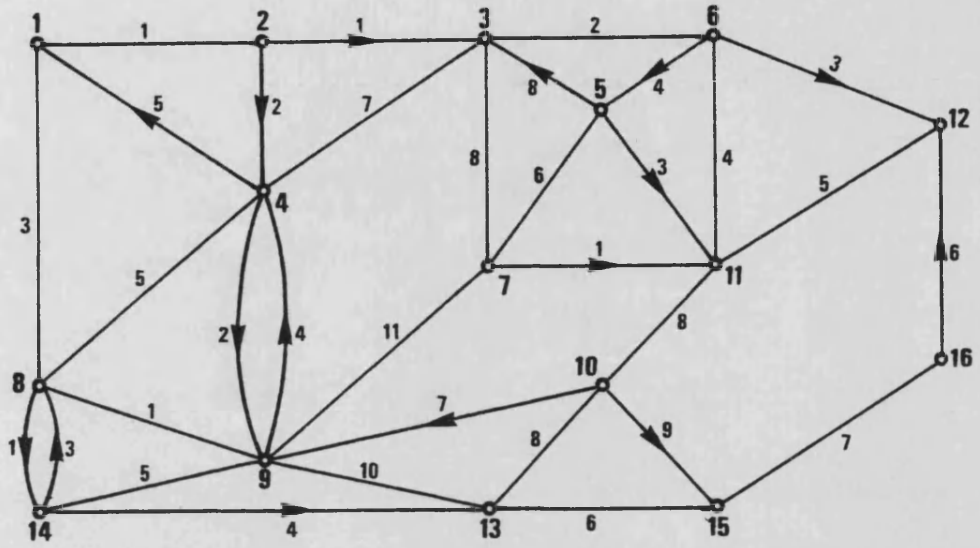


Figura 4.1: Grafo mixto original G.

En Etapa 1 del algoritmo, se detectan las aristas de corte en G, tal como ya fué explicado en la Subsección 3.2, por la aplicación de las Proposiciones de 3.3. a 3.7, y a continuación se hacen las copias necesarias de los arcos de corte en virtud de las Proposiciones 3.8 y 3.9. En este caso, tenemos -- que el grafo ejemplo de la Figura 4.1 presenta tres aristas de corte: la arista $\{1,2\}$ es de corte en sentido de 1 a 2, por - lo que se transforma en el arco de corte (1,2), la arista -- $\{11,12\}$ es de corte en sentido de 12 a 11, transformándose en - el arco (12,11), y la arista $\{15,16\}$ es de corte en sentido de 15 a 16, obteniendo el tercer arco de corte (15,16). Por último, tanto el nuevo arco de corte (1,2) como el (12,11) deben reco-- rrerse al menos dos veces en cualquier solución al (MCPP), por lo tanto añadimos una copia de cada uno de ellos obteniendo el multigrafo mixto G^T que se muestra en la Figura 4.2

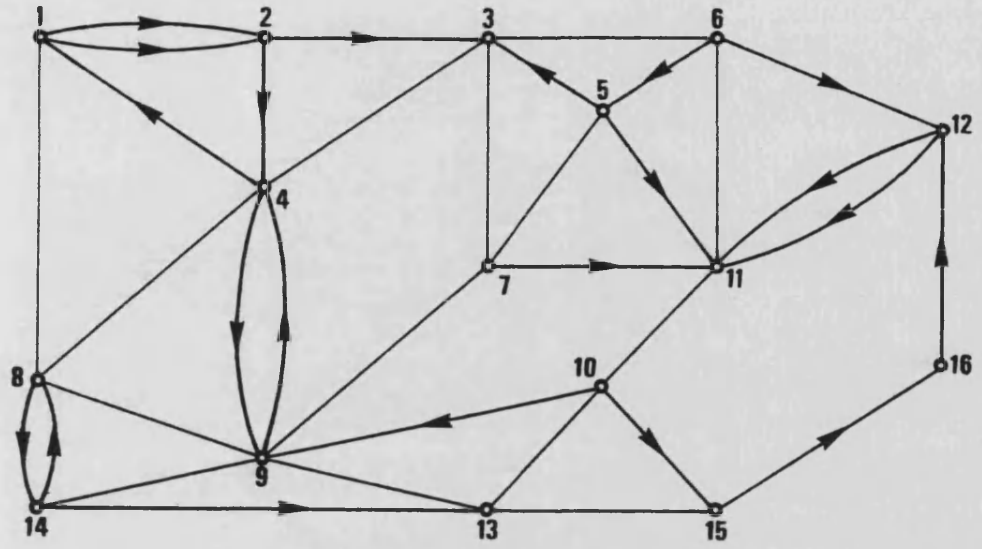


Figura 4.2: El grafo transformado G^T.

En la Etapa 2 asignamos unas direcciones aleatorias a cada una de las aristas de G^T, éstas direcciones se detallan en la Tabla siguiente respecto a los grafo G_d y G'.

Arista	Dirección en G_d	Arco art. en G'
{1,8}	1 → 8	(8,1)
{3,4}	3 → 4	(4,3)
{3,6}	3 → 6	(6,3)
{3,7}	3 → 7	(7,3)
{4,8}	4 → 8	(8,4)
{5,7}	5 → 7	(7,5)
{6,11}	6 → 11	(11,6)
{7,9}	7 → 9	(9,7)
{8,9}	8 → 9	(9,8)
{9,13}	9 → 13	(13,9)
{9,14}	9 → 14	(14,9)
{10,11}	10 → 11	(11,10)
{10,13}	10 → 13	(13,10)
{13,15}	13 → 15	(15,13)

El grafo G_d obtenido tras la asignación de las -- direcciones aleatorias es representado en la Figura 4.3.

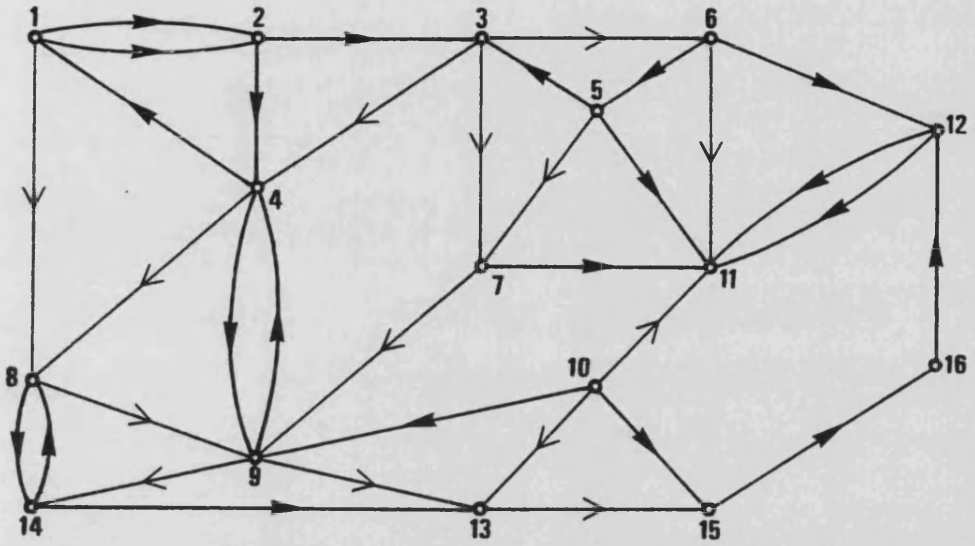


Figura 4.3: El grafo G_d.

En G_d determinamos los conjuntos de fuentes y sumideros teniendo en cuenta las asignaciones de dirección de las aristas.

Fuentes	D(i)	Sumideros	D(i)
8	1	1	-2
9	1	3	-1
11	6	5	-2
13	2	6	-2
15	1	10	-4

En la Etapa 4 se define el grafo G' que se representa en la Figura 4.4, resolviendo ahora el problema de flujo de coste mínimo con los valores de D(i) calculados anteriormente, cuya solución es:

$\hat{f}^g(8,1)=2$, $\hat{f}^g(4,3)=0$, $\hat{f}^g(6,3)=1$, $\hat{f}^g(7,3)=0$, $\hat{f}^g(8,4)=0$, $\hat{f}^g(7,5)=1$
$\hat{f}^g(11,6)=2$, $\hat{f}^g(9,7)=1$, $\hat{f}^g(9,8)=1$, $\hat{f}^g(13,9)=1$, $\hat{f}^g(14,9)=0$,
$\hat{f}^g(13,10)=2$, $\hat{f}^g(11,10)=2$, $\hat{f}^g(15,13)=1$, $f(6,5)=1$, $f(11,6)=2$,

Siendo los restantes flujos por arcos no artificiales cero, -- con lo que el valor total del flujo óptimo ha sido de 12 (re-- cordemos que el coste asociado a cada uno de los arcos artificiales es cero).

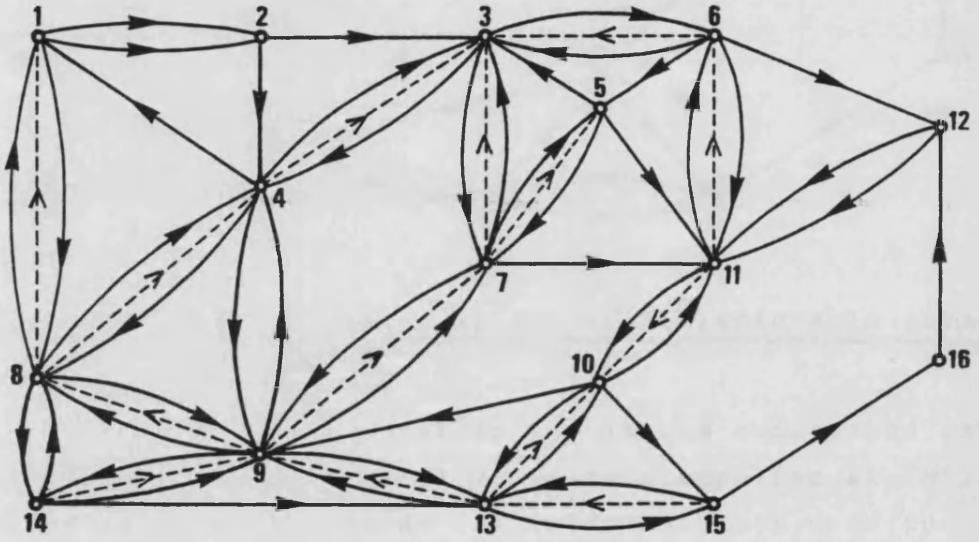


Figura 4.4: Grafo G' { Arcos artificiales ----→---- ; Arcos no artificiales ——→——

Finalmente de acuerdo con la Etapa 6 del algoritmo, construimos el grafo euleriano asignado de G, $\tilde{G}$ en función de los valores de los flujos que han circulado por cada arco - del grafo G'. El grafo $\tilde{G}$ es representado en la Figura 4.5.

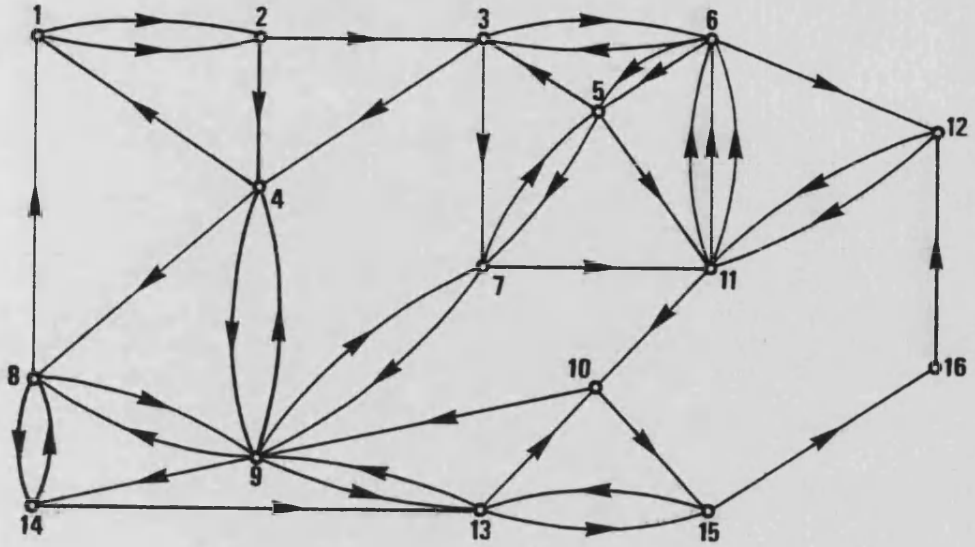


Figura 4.5: Grafo euleriano $\widetilde{G}$, correspondiente a la cota superior inicial.

El coste total de las líneas existentes en G, nos proporciona el valor inicial de la cota superior al (MCPP), cuyo valor es en este caso de 214 unidades. Esta cota puede ser mejorada sustituyendo:

1º El arco (9,13) de coste 10, por el camino más corto de coste 6 entre los vértices 9 y 13 (en este orden), dado por la sucesión de arcos (9,8),(8,14),(14,13).

2º El camino formado por los arcos (9,7), (7,5), de coste 17, por el camino más corto de coste 12 entre los vértices 9 y 5 (en este orden), formado por los arcos (9,8),(8,1),(1,2),(2,3),(3,6),(6,5).

3º La arista {9,8}, es ahora recorrida tres veces en la dirección 9 → 8, y una vez en dirección 8 → 9, por lo que pueden ser eliminadas dos copias (una en cada sentido), de los arcos que sustituyen a esta arista en el grafo G. Análogamente, la arista {3,6} que es recorrida en dirección 3 → 6 dos veces, y en dirección 6 → 3 una vez, proporciona una última simplificación eliminando dos de sus copias (una en cada sentido).

Por lo tanto, el coste de la cota superior mejorada según hemos visto en los puntos 1º, 2º y 3º, es de 199, - y su correspondiente grafo euleriano asignado se representa en la Figura 4.6

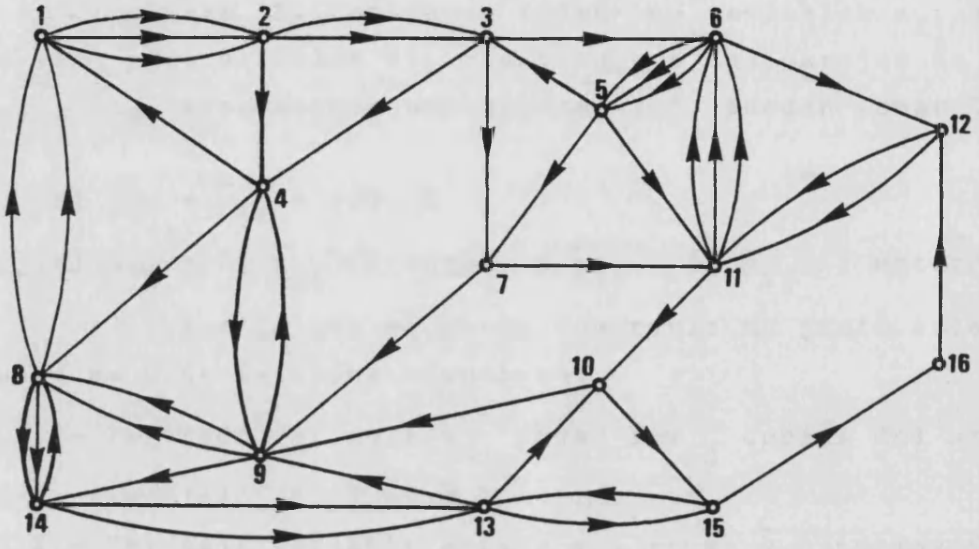


Figura 4.6: Grafo euleriano asignado de G, correspondiente a - la cota superior mejorada.

4.3. SOLUCION POSIBLE A PARTIR DE LA RELAJACION LINEAL.

El problema $(\overline{P})$, consiste en eliminar las restricciones de integridad de las variables que aparecen en la formulación de (P). Como ya se dijo en la Subsección 2.2.3, Kappauf y Koehler [40] demuestran que si $(\overline{x},\overline{y})$ es la solución óptima del problema $(\overline{P})$, entonces todas las variables x_{ij} son enteras (incluido el valor 0), mientras que las parejas de variables y_{ij}, y_{ji} asociadas a una arista $\{i,j\}$ pueden tomar los valores:

(a) $\overline{y}_{ij} = \overline{y}_{ji} = 1/2$, ó

(b) $\overline{y}_{ij} = 0$, $\overline{y}_{ji} \geq 1$ entero ó $\overline{y}_{ji} = 0$, $\overline{y}_{ij} \geq 1$ entero.

Por lo que se puede construir un grafo euleriano asignado de G de la forma siguiente:

1.- Por cada variable x_{ij} poner $1+\overline{x}_{ij}$ copias del arco asociado (i,j) en $\overline{G}$.

2.- Por cada variable entera y_{ij} poner $\overline{y}_{ij}$ copias de arcos en $\overline{G}$ (en representación de la arista $\{i,j\}$).

3.- Por cada pareja de variables y_{ij}, y_{ji} tales que $\overline{y}_{ij} = \overline{y}_{ji} = 1/2$, poner una copia del arco (i,j) y otra del arco (j,i) (en representación de la arista $\{i,j\}$).

Este procedimiento aplicado al grafo de la Figura 4.1 produjo el grafo euleriano asignado de G de la Figura 4.7.

La solución obtenida en este caso por la resolución de $(\overline{P})$, tiene un coste total de 195 unidades lo que representa una disminución de 4 unidades con respecto a la solución obtenida por nuestro heurístico, no obstante, como se puede apreciar en la Tabla 4.2, ésto no ha ocurrido en la mayoría de los problemas testados cuyas características se ofrecerán con más detalle en la Sección 6.

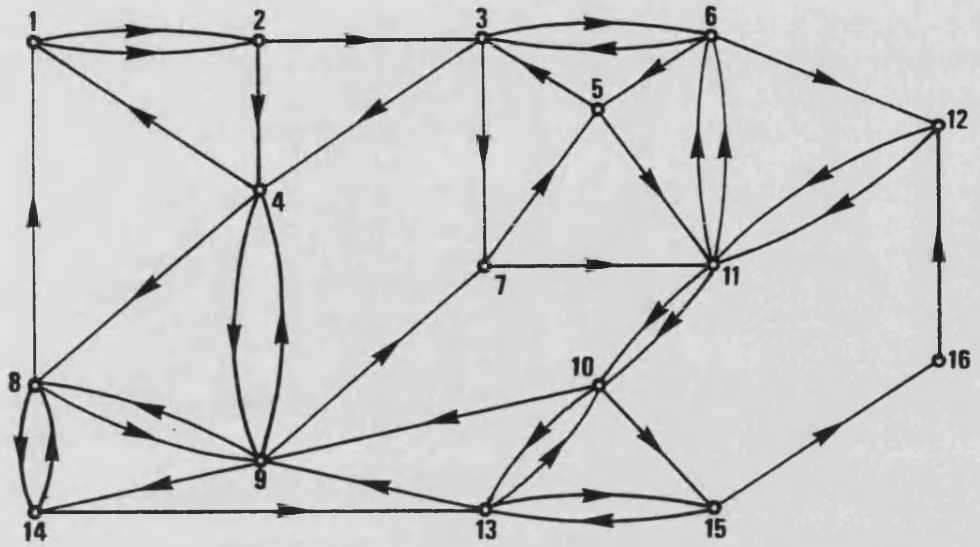


Figura 4.7: Grafo $\overline{G}$ obtenido a partir de la resolución de $(\overline{P})$.

Comentario 4.1

De los 35 problemas presentados en la Tabla 4.2, cuyas características más importantes se detallarán en la Sección 6, hay tres (29,30 y 34) en los que no figura el valor óptimo pues ello requería más del tiempo límite de 500 segundos de C.P.U, que nos hemos fijado, para resolver el (MCPP) en cada uno de estos ejemplos,con un ordenador UNIVAC 1100/60.

De los 32 problemas que se han resuelto óptimamente, el heurístico ha proporcionado el valor óptimo en doce de ellos, frente a los seis óptimos proporcionados por la solución posible obtenida a partir de la resolución de $(\overline{P})$. Por otra parte, el promedio de alejamiento, en tanto por cien, respecto del valor óptimo , ha sido, para el heurístico del 2.3%, mientras que para la relajación lineal del 3.4% (el valor óptimo de cada problema se encuentra en la columna 5 de la Tabla - 5.6), lo que significa que el porcentaje medio de cercanía al valor óptimo ha sido del 1.1% a favor del heurístico.

El tiempo medio de resolución por problema para el heurístico,es aproximadamente ocho veces menor que para re

solución de la relajación lineal,usando el paquete de programas FMPS de la firma UNIVAC, 0.28 segundos para el heurístico frente a 2.45 segundos para el FMPS.

Finalmente, el heurístico, proporciona información sobre las aristas y arcos de corte, con sus posibles duplicaciones, en el grafo original, obteniendo,en su caso, un grafo transformado en el que las direcciones de algunas aristas originales ya han sido fijadas (véase Tabla 6.1).

Resumiendo diremos que las cualidades de cercanía al valor óptimo, rapidez de computación, e información sobre las direcciones que toman las aristas de corte en el grafo original, a favor del heurístico, hacen que el algoritmo heurístico sea preferible para obtener soluciones posibles al --- (MCPP) que la relajación lineal. Por todo ello, hemos elegido como único modo de obtención de la cota superior inicial para el algoritmo de Branch and Bound que presentaremos en la Sección 6, el valor producido por la solución del heurístico.

1	2	3	4	5	6	7
1	83	0%	0.03	83	0%	1.17
2	54	0%	0.05	54	0%	0.98
3	36	0%	0.03	36	0%	1.20
4	31	0%	0.08	31	0%	1.41
5	140	0%	0.09	146	4.3%	1.45
6	194	7.2%	0.16	194	7.2%	1.58
7	55	0%	0.05	62	12.7%	1.27
8	122	0%	0.08	124	1.6%	1.42
9	191	0%	0.15	191	0%	1.84
10	95	6.7%	0.07	100	12.3%	1.09
11	122	0%	0.10	123	0.8%	1.30
12	199	4.7%	0.15	195	2.6%	1.59
13	184	0%	0.14	188	2.2%	1.90
14	657	4.3%	0.32	674	7.0%	2.51
15	887	0.7%	0.51	888	0.8%	3.24
16	257	1.6%	0.20	259	2.4%	2.13
17	67	1.5%	0.25	66	0%	2.50
18	394	2.1%	0.25	399	3.4%	2.46
19	235	11.4%	0.15	234	10.9%	1.64
20	123	3.4%	0.08	129	8.4%	1.49
21	826	1.5%	0.51	826	1.5%	3.82
22	610	1.5%	0.33	610	1.5%	2.73
23	811	0%	0.38	811	1.5%	2.75
24	793	0.2%	0.39	808	2.1%	3.10
25	972	0.8%	0.41	974	1.0%	3.51
26	726	3.1%	0.41	745	5.8%	3.36
27	1062	2.1%	0.57	1062	2.1%	3.50
28	1170	0.7%	0.59	1170	0.7%	4.37
29	936	---	0.56	960	---	3.84
30	946	---	0.63	946	---	4.15
31	275	7.0%	0.16	267	3.9%	1.87
32	338	9.7%	0.26	327	6.2%	2.19
33	317	4.6%	0.23	317	4.6%	2.18
34	1198	---	0.87	1202	---	4.80
35	1202	0%	0.73	1207	0.4%	4.62

Tabla 4.2: Comparación heurístico-Solución posible a partir de la relajación lineal.

1. Número del problema.
2. Cota superior proporcionada por el heurístico.
3. % de alejamiento del óptimo utilizando el heurístico.
4. Tiempo de C.P.U en segundos para obtener la solución del heurístico.
5. Valor solución posible obtenida por la resolución de $(\overline{P})$.
6. % de alejamiento del óptimo utilizando $(\overline{P})$.
7. Tiempo de C.P.U en segundos para resolver $(\overline{P})$, con el FMPS.

S E C C I O N 5

RELAJACIONES

En esta Sección, nos proponemos el estudio de los problemas relajados que proporcionan cotas inferiores al valor óptimo del problema (P), así como su posible utilización en el algoritmo de tipo Branch and Bound que se presentará en la Sección 6. Como bien es sabido, el éxito de un algoritmo de este tipo, estriba en gran parte en la calidad de las cotas inferiores utilizadas. Sin embargo, no siempre el procedimiento que proporciona las cotas inferiores más próximas al valor óptimo de un problema, es el más atractivo desde el punto de vista operativo. Usualmente una buena cota inferior puede perder su potencia si consume excesivo tiempo computacional, ya que este hecho, la eliminaría como cota inferior útil. La dificultad estriba entonces, en obtener cotas inferiores rápidas de calcular, pero que además sean efectivas. La utilización de cotas inferiores derivadas de la relajación lagrangiana, expuesta someramente en la Sección 1, ha supuesto una alternativa de importancia vital para la resolución de muchos problemas de Optimización Combinatorial, como ya se ha demostrado en las aplicaciones expuestas en [20] , y en particular, en la resolución del problema del agente viajero (TSP); como caso ejemplar, ver [6] y [9] . Tres, son las relajaciones que han sido estudiadas para obtener buenas cotas inferiores de las que solamente una ha resultado conveniente,imbuida en el algoritmo exacto para resolver el (MCPP), que se expondrá en la Sección siguiente.

Estudiaremos por separado:

1.- La relajación lineal (RLin).
2.- La relajación lagrangiana, respecto a las restricciones de simetría (RL1).
3.- La relajación lagrangiana, respecto a las restricciones de obligatoriedad (RL2).

5.1 LA RELAJACION LINEAL

Esta relajación consiste, como sabemos, en la eliminación del conjunto de restricciones de integridad sobre las variables que aparecen en la formulación de (P), con lo cual -- obtenemos el problema de minimización lineal ($\overline{P}$), con idéntica función objetivo que (P), pero sujeta únicamente al conjunto de restricciones (1), (2) y (3). Este problema relajado ha sido estudiado en [40], y se ha dado un resumen de las propiedades -- más interesantes que presenta el poliedro del conjunto de sus - soluciones posibles, en la Sección 2 de esta memoria.

A partir de la solución del problema ($\overline{P}$), se pueden obtener simultáneamente una cota superior y una cota inferior al valor óptimo del (MCPP), lo cual para ciertas aplicaciones prácticas puede interesar, a modo de test, si la optimalidad no es un requisito indispensable, puesto que si la diferencia entre estos dos valores es relativamente pequeña, puede que no compense el esfuerzo computacional requerido para obtener la solución óptima. Por otra parte, hemos resuelto el problema ($\overline{P}$) en 35 problemas test utilizando el algoritmo primal-dual del -- simplex inmerso en el paquete de programa FMPS que está considerado como uno de los más rápidos en la actualidad, pudiendo observar en la Tabla 4.2, que el tiempo de C.P.U. consumido en un ordenador UNIVAC Serie 1100/60 para un problema con 65 restricciones y 135 variables (problema nº 21) ha sido de 3.8 segundos, que puede parecer "poco tiempo", si solamente lo utilizásemos - un número reducido de veces, pero que sin embargo resulta excesivo si se pretende utilizar la relajación lineal para producir cotas inferiores en el árbol de ramificación de un algoritmo de Branch and Bound. Además, la cota inferior producida por la relajación lineal, y el método del subgradiente aplicado a la relajación lagrangiana respecto a las restricciones de obligatoriedad, que será expuesta en la Subsección 5.3, ha diferido en menos del 1% en promedio una de la otra, utilizando un máximo de - 20 iteraciones con un tiempo de computación mucho menor.

Notación 5.1

Dado el conjunto de aristas L del grafo G, representaremos por L_d al conjunto de todos los pares ordenados -- (i,j), (j,i) asociados a cada arista $\{i,j\} \in L$. Esto es:

$$L_d = \{(i,j), (j,i): \exists \{i,j\} \in L\}$$

en L_d están representadas todas las parejas de direcciones posibles de cada una de las aristas de G.

A continuación ofrecemos la formulación del problema lineal dual de $(\overline{P})$ que llamaremos $(D\overline{P})$, con objeto de relacionar las variables duales con los multiplicadores lagrangianos que aparecerán en las dos relajaciones lagrangianas de las Subsecciones 5.2 y 5.3.

5.1.1 El problema dual de $(\overline{P})$: $(D\overline{P})$.

Dualizando las restricciones (1) y (2) del problema lineal $(\overline{P})$, asociando las variables duales ν_i con i=1,...,n a cada una de las restricciones de simetría referentes a los -- vértices i del grafo G, y una variable γ_1 con 1=1,...,p a cada una de las restricciones de obligatoriedad referentes a las p - aristas del grafo G, obtenemos el siguiente problema lineal de maximización:

$$(D\overline{P}) \qquad \text{Max} \left\{ \sum_{i=1}^{n} D(i)\nu_i + \sum_{1=1}^{p} \gamma_1 + \sum_{A} c_{ij} \right\} \qquad (0_D)$$

sujeto a:

$$\nu_i - \nu_j \leq c_{ij} \qquad \forall (i,j) \in A \qquad (1_D)$$

$$\nu_i - \nu_j + \gamma_1 \leq c_{ij} \qquad \forall (i,j) \in L_d \qquad (2_D)$$

$$\left.\begin{array}{ll} \nu_i \text{ no restringido} & \forall i \\ \gamma_1 \geq 0 & \forall 1 \end{array}\right\} \qquad (3_D)$$

donde podemos observar que existe una restricción por cada arco en el grafo G, pero dos restricciones por cada arista en G. Las variables ν_i no están restringidas en signo, puesto que sus restricciones primales asociadas eran igualdades, mientras que las

variables γ_l son requeridas a ser no negativas.

Resolver el problema ($D\overline{P}$) es, desde el punto de vista computacional, tan costoso como resolver el problema primal ($\overline{P}$). Nosotros únicamente utilizaremos el problema dual, para la obtención de propiedades de los problemas lagrangianos que serán definidos más adelante.

Notación 5.2

Representaremos por (ν^*,γ^*) a la solución óptima del problema dual ($D\overline{P}$).

Proposición 5.1

Si el vector $(\overline{\nu}, \overline{\gamma})$ es una solución posible del problema ($D\overline{P}$), entonces para cada arista $l=\{i,j\}$ tenemos que $\overline{\gamma}_l \leq c_{ij}$ y $\left|\overline{\nu}_i - \overline{\nu}_j\right| \leq c_{ij} - \overline{\gamma}_l$

Demostración

Consideremos la pareja de restricciones asociadas a la arista $l=\{i,j\}$ del conjunto de restricciones (2_D) del problema lineal ($D\overline{P}$). Puesto que $(\overline{\nu},\overline{\gamma})$ es posible, tenemos:

$$\left.\begin{array}{l} \overline{\nu}_i - \overline{\nu}_j + \overline{\gamma}_l \leq c_{ij} \\ \overline{\nu}_j - \overline{\nu}_i + \overline{\gamma}_l \leq c_{ji} \end{array}\right\} \quad \forall l=\{i,j\} \in L \qquad (5.1)$$

Ahora bien, puesto que el coste de recorrer la arista $\{i,j\}$ es el mismo independientemente del sentido que se elija, tenemos que $c_{ij} = c_{ji}$. Sumando las desigualdades en la expresión (5.1) obtenemos que $\overline{\gamma}_l \leq c_{ij}$, y también $\left|\overline{\nu}_i - \overline{\nu}_j\right| \leq c_{ij} - \overline{\gamma}_l$. ●

5.2 LA RELAJACION LAGRANGIANA DE LAS RESTRICCIONES DE SIMETRIA

Como ya vimos en la Sección 1, la relajación lagrangiana de un conjunto de restricciones de un problema de optimización, consiste en eliminarlas del conjunto total de restricciones, incorporándolas a la función objetivo por medio de un multiplicador lagrangiano. En nuestro caso, consideremos las restricciones de simetría (1), que como ya vimos en la Sección 3, son equivalentes a las restricciones (1'), cuya expresión recordamos que era:

$$\sum_j x_{ij} + \sum_j y_{ij} - \sum_j x_{ji} - \sum_j y_{ji} = D(i) \qquad \forall i \qquad (1')$$

siendo $D(i)=d_e^G(i) - d_s^G(i)$.

Relajando lagrangianamente las restricciones (1'), asociando un multiplicador u_i a la restricción i-ésima de (1'), para cada una de éstas, añadiendo además al conjunto de restricciones (2),(3) y (4), no relajadas de (P), el conjunto de restricciones de paridad (5), que como se demostró en la Proposición 3.10, son redundantes con las restricciones (1'), incluyendo también las restricciones de integridad sobre las variables w_i, cuya no negatividad era redundante con las restricciones (2),(3) y (4), obtenemos la formulación del problema lagrangiano relajado, que llamaremos PR(x,y,u):

$$(PR(x,y,u)) \qquad \text{Min} \left\{ \mathcal{L}_1(x,y,u) = \sum_A (c_{ij}+u_j-u_i)x_{ij} + \sum_{L_d} (c_{ij}+u_j-u_i)y_{ij} + \sum_i D(i)u_i + \sum_A c_{ij} \right\}$$

sujeto a: (2),(3),(4),(5) y siendo w_i enteras $\forall i$, mientras que los multiplicadores u_i no están restringidos.

5.2.1 La resolución de PR(x,y,u)

Dado un conjunto de multiplicadores lagrangianos u_i cualesquiera, debemos resolver el problema de minimización en x e y: PR(x,y,u). En primer lugar, veremos la Proposición siguiente, en la que se demuestra que para que el problema relajado PR(x,y,u) esté acotado inferiormente, el vector de multiplicadores u, no debe transformar ningún coste en negativo.

Proposición 5.2

Dado un vector u de multiplicadores, el problema PR(x,y,u) está acotado inferiormente sii:

$$\tilde{c}_{ij} = c_{ij} + u_j - u_i \geq 0 \qquad \forall (i,j) \in A \quad y \quad \forall (i,j) \in L_d.$$

Demostración

Supongamos que existe un coste modificado $\tilde{c}_{ht} < 0$, y que $(\bar{x},\bar{y})$ es una solución cualquiera de PR(x,y,u), sea M un número entero positivo tan grande como queramos, tenemos las dos posibilidades siguientes:

(a) Si $(h,t) \in A$, entonces haciendo $\hat{x}_{ht} = \bar{x}_{ht} + 2M$, se siguen cumpliendo las restricciones de paridad, con un incremento de la función objetivo, dado por:

$$\Delta f_1(x,y,u) = f_1(\hat{x},y,u) - f_1(\bar{x},\bar{y},u) = 2M\, \tilde{c}_{ht} < 0$$

por lo que PR(x,y,u) no está acotado inferiormente.

(b) Si $(h,t) \in L_d$, entonces haciendo $\hat{y}_{ht} = \bar{y}_{ht} + M$, $\hat{y}_{th} = \bar{y}_{th} + M$ también se siguen cumpliendo las restricciones de paridad, con un incremento estrictamente negativo que, al igual que el caso (a), es de $2M\tilde{c}_{ht}$ (i.e: dependiente de M), teniendo la misma conclusión que en (a).

La condición suficiente es obvia. ●

En virtud de la Proposición 5.2, tenemos que para obtener cotas inferiores del valor óptimo del problema (P) que no sean irrelevantes, es necesario seleccionar los multiplicadores u, de forma que los costes modificados que figuran en la función objetivo del problema PR(x,y,u) sean todos no negativos. La siguiente Proposición nos da la posibilidad de obtener unos

multiplicadores u que satisfacen ésta condición.

Proposición 5.3

Toda solución posible del problema dual $(\overline{DP})$, proporciona un vector de multiplicadores lagrangianos al problema PR(x,y,u), usando las variables duales asociadas a las restricciones de simetría, de forma que $\tilde{c}_{ij} \geq 0 \quad \forall (i,j) \in A \cup L_d$.

Demostración

Supongamos que el vector $(\overline{v},\overline{\gamma})$ es una solución posible del problema dual $(\overline{DP})$ (siempre existe puesto que el problema primal $(\overline{P})$ tiene solución óptima finita), en virtud de las restricciones duales (1_D) y (2_D), tenemos que:

$$\left.\begin{array}{ll} c_{ij} + \overline{v}_j - \overline{v}_i \geq 0 & \forall (i,j) \in A \\ c_{ij} + \overline{v}_j - \overline{v}_i \geq \overline{\gamma}_1 \geq 0 & \forall (i,j) \in L_d \end{array}\right\} \qquad (5.2)$$

Ahora bien, considerando que los costes modificados se han definido como $\tilde{c}_{ij} = c_{ij} + u_j - u_i \quad \forall (i,j) \in A \cup L_d$, haciendo $u_i = \overline{v}_i \quad \forall i$, obtenemos el resultado deseado. ●

Proposición 5.4

Dado un vector de multiplicadores $\overline{u}$ tal que $\tilde{c}_{ij} \geq 0$ $\forall (i,j) \in A \cup L_d$, y dada una solución óptima $(\overline{x},\overline{y})$ del problema $PR(x,y,\overline{u})$, tenemos:

(a) Si $\overline{u}_i < \overline{u}_j$ para algún $(i,j) \in L_d$, entonces $\overline{y}_{ji} = 0$.

(b) Si $\overline{u}_i = \overline{u}_j$ para algún $(i,j) \in L_d$, entonces $\overline{y}_{ij} = 0$ ó $\overline{y}_{ji} = 0$, sin pérdida de optimalidad.

Demostración (por reducción al absurdo)

(a) Supongamos que $\overline{y}_{ji} \geq 1$, tenemos además por hipótesis que $\tilde{c}_{ij} < \tilde{c}_{ji}$ y por consiguiente haciendo $y_{ij} = \overline{y}_{ij} + \overline{y}_{ji}$ e $y_{ji} = 0$, obtenemos una nueva solución posible del problema $PR(x,y,\overline{u})$, pero con un coste de la función objetivo $f_1(\overline{x},\overline{y},\overline{u}) + (\tilde{c}_{ij} - \tilde{c}_{ji})\,\overline{y}_{ji} < f_1(\overline{x},\overline{y},\overline{u})$, lo cual es absurdo.

(b) Supongamos que $i<j$ (sin pérdida de generalidad), y que simultáneamente $\bar{y}_{ij} \geq 1$ e $\bar{y}_{ji} \geq 1$. En este caso se sabe por hipótesis que $\tilde{c}_{ij} = \tilde{c}_{ji}$, por lo que haciendo $y_{ij}=\bar{y}_{ij} + \bar{y}_{ji}$, e $y_{ji}=0$, obtenemos una solución posible del mismo coste que la producida por $(\bar{x},\bar{y})$, esto es, sin pérdida de optimalidad. ●

Comentario 5.1

La Proposición 5.4 permite, antes de resolver el problema PR(x,y,u) dado un multiplicador u de antemano, asignar el valor 0 exactamente a p variables y_{ij}, una por cada pareja de variables asociadas a cada arista del grafo G, sin modificar el valor óptimo del problema PR(x,y,u). Por otra parte, es obvio, que el multiplicador nulo (i.e: con todas sus coordenadas iguales a cero), es una posible elección para el problema relajado, de forma que éste esté acotado inferiormente, puesto que los costes modificados,en este caso, coinciden con los costes originales.

En lo sucesivo, adoptaremos la regla siguiente para caracterizar unívocamente a las p variables y_{ij} que se les da el valor 0 antes de resolver el problema PR(x,y,u):
" Si entre dos vértices i,j , tales que $i<j$ (suponemos los vértices ordenados 1,2,...,n), existe la arista {i,j} y tenemos que $u_i = u_j$, entonces asignaremos el valor 0 a la variable y_{ji}"

De acuerdo con la regla anterior consideremos la siguiente Definición.

Definición 5.1

Dado un vector de multiplicadores u del problema PR(x,y,u), sea el conjunto de pares ordenados:

$$L_d(u) = \{(i,j)\in L_d: u_j < u_i \text{ ó si } u_j = u_i \text{ entonces } i < j\} .$$

En el conjunto $L_d(u)$ hay representadas exactamente p (número de aristas en G) direcciones, dependiendo de la relación existente entre las parejas de multiplicadores asociados al par de vértices terminales de cada arista de G.

Resolver el problema PR(x,y,u), dado un vector de multiplicadores u, es pues equivalente a resolver el problema PR(x,y',u) siguiente:

$$(PR(x,y',u)) \quad \underset{x,y'}{\text{Min}} \left\{ f_1(x,y',u) = \sum_A \tilde{c}_{ij} x_{ij} + \sum_{L_d(u)} \tilde{c}_{ij} y'_{ij} + K(u) \right\}$$

sujeto a:

$$\sum_j x_{ij} + \sum_j x_{ji} + \sum_j y'_{ij} - 2w_i = b_i \qquad \forall i \qquad (5')$$

$$\left.\begin{array}{ll} x_{ij},\ y'_{ij} \geq 0, \text{ enteras} & \forall (i,j) \in A \cup L_d(u) \\ w_i \text{ enteras} & \forall i = 1,\ldots,n \end{array}\right\} \qquad (6')$$

En donde el término K(u) de la función objetivo viene dado por la expresión:

$$K(u) = \sum_{L_d(u)} \tilde{c}_{ij} + \sum_i D(i) u_i + \sum_A c_{ij} \qquad (5.3)$$

es independiente de los valores de las variables x_{ij}, y'_{ij}, siendo:

$$y'_{ij} = y_{ij} + y_{ji} - 1 \qquad \forall (i,j) \in L_d(u) \qquad (5.4)$$

es decir, y'_{ij} únicamente existe cuando existe el par ordenado $(i,j) \in L_d(u)$, y finalmente de acuerdo con la Proposición 5.4 tenemos que:

$$\left.\begin{array}{ll} v(PR(x,y,u)) = v(PR(x,y',u)) & \forall u \text{ tal que } \tilde{c}_{ij} \geq 0 \\ & \forall (i,j) \in A \cup L_d(u) \end{array}\right\} \qquad (5.5)$$

Para resolver el problema PR(x,y',u) dado un vector de multiplicadores u, que proporcione unos costes modificados no negativos, vamos a separar el término K(u) de la función objetivo de PR(x,y',u) obteniendo el problema de minimización:

$$(P_1) \qquad \text{Min} \left\{ \sum_A \tilde{c}_{ij} x_{ij} + \sum_{L_d(u)} \tilde{c}_{ij} y'_{ij} \right\}$$

sujeto a: (5') y (6').

El conjunto de restricciones (5'), indica que la paridad de cada vértice del grafo G debe ser par, en donde hemos demostrado en la Proposición 5.4, que únicamente se necesita una variable y_{ij} de las dos que tiene asociadas cada arista $\{i,j\}$ del grafo G, esta variable es por la definición de $L_d(u)$ la que tiene un coste reducido $\tilde{c}_{ij}$ menor, o en caso de que ambas tengan el mismo coste reducido asociado, hemos utilizado el convenio de escoger aquella variable y_{ij} tal que $i<j$. El problema (P_1) consiste pues, en minimizar el número extra de arcos (expresado por x_{ij}), y de aristas (expresado por y'_{ij}), que son necesarias para convertir el grafo G en par. Es necesario observar, que las direcciones de los arcos son irrelevantes para conseguir este objetivo y que, por lo tanto, el problema (P_1) representa la formulación del (CPP) sobre el grafo no dirigido $G'=(N,L')$, con costes asociados $\tilde{c}_{ij}$ y donde L' es un conjunto formado por todas las aristas $l=\{i,j\}$ del grafo G, con coste modificado $\tilde{c}_l=\min\{\tilde{c}_{ij},\tilde{c}_{ji}\}$, y con las aristas obtenidas a partir de los arcos de G, ignorando su dirección, con su correspondiente coste modificado. En total, existen en L' m+p aristas, y por lo dicho $v(PR(x,y',u))=v(P_1)+K(u)$.

Resumiendo, ofrecemos a continuación el procedimiento general para obtener cotas inferiores válidas utilizando la relajación lagrangiana de las restricciones de simetría que abreviadamente representaremos por (RL_1).

<u>Procedimiento general para obtener cotas inferiores a v(P) utilizando (RL_1)</u>

<u>Etapa 1:</u> Seleccionar un vector de multiplicadores $\hat{u}$, de forma que la matriz de costes $[\hat{c}_{ij}]$ tenga sus elementos no negativos.

<u>Etapa 2:</u> Calcular los costes de los caminos más cortos entre todos los vértices de grado impar en G'.

<u>Etapa 3:</u> Calcular un 1-acoplamiento perfecto de mínimo coste M*, en el grafo completo formado por los vértices impares en G', utilizando los costes de

los caminos más cortos entre ellos calculados - en la Etapa 2. Sea c(M*), el coste de dicho acoplamiento.

Etapa 4: Calcular $v(PR(x,y',\hat{u})) = c(M^*) + K(\hat{u})$. Stop.

Para resolver la Etapa 1, como ya se ha señalado antes el vector de multiplicadores $\hat{u} = 0$, es una posible -- elección, o bien, si es conocida alguna solución del problema dual $(D\overline{P})$, podemos conseguir un vector de multiplicadores $\hat{u}$ -- (tal como se demostró en la Proposición 5.3), que puede proporcionar mejores cotas inferiores; sin embargo, no hay garantía de que ésto ocurra a causa de que al modificar los costes, las distancias de los caminos más cortos calculados en la Etapa 2, también cambian, es por ésto, que si intentamos aumentar en lo posible el valor de K(u) resolviendo el problema lineal:

$$(Q) \qquad \max_{u} \; K(u)$$

sujeto a:

$$u_i - u_j \leq c_{ij} \qquad \forall (i,j) \in A \cup L_d$$

no podemos asegurar que el valor de la cota inferior vaya a -- ser superior al obtenido con otros multiplicadores u. Por otra parte los intentos de obtener buenos multiplicadores u resol-- viendo problemas lineales como (Q), sin una garantía de incremento positivo de la cota inferior proporcionada por (RL_1), repercutirían en un coste computacional excesivo que haría poner en duda la eficiencia del algoritmo. Afortunadamente, la se--- lección u = 0, no es excesivamente mala, y hemos diseñado un - algoritmo heurístico, que permite el cálculo rápido de unos -- "buenos" multiplicadores que en la práctica han dado buenos resultados. El heurístico está basado en las siguientes ideas:

1º. Maximizar en lo posible K(u), teniendo en cuenta el - término $\sum_i D(i)u_i$. Esto se hace con una regla heurística sencilla: se asignan multiplicadores u_i no negativos a aquellos vértices i tales que $D(i) > 0$, y multiplicadores u_i negativos o nulos a aquellos vértices

tales que $D(i) < 0$. Mientras que si $D(i) = 0$, el valor de u_i es irrelevante, pero se hace igual a cero con objeto de maximizar $\sum_{L_d(u)} \tilde{c}_{ij}$ respecto a las aristas incidentes con el vértice i. Por otra parte, si un vértice i es incidente con un número mayor de aristas que el valor absoluto de D(i) es conveniente, la mayoría de los casos, asignar el valor cero al multiplicador u_i correspondiente.

2º. Los costes modificados $\tilde{c}_{ij} = c_{ij} + u_j - u_i$ obtenidos tras el cómputo de los multiplicadores u_i $\forall i$, deben ser no negativos.

El algoritmo para el cálculo de los multiplicadores u_i es descrito a continuación tal y como ha sido utilizado en la práctica.

Heurístico para el cálculo de los multiplicadores u.

Paso 1: Dar una ordenación $i_1, i_2, \ldots, i_n$ a todos los vértices del grafo G de forma que:

$$\left|D(i_k)\right| \geq \left|D(i_{k+1})\right| \qquad \forall k = 1, \ldots, n-1.$$

Paso 2: Hacer $\tilde{c}_{ij} = c_{ij}$ $\forall (i,j) \in A \cup L_d$. Calcular $d_a^G(i_k)$.

Paso 3: Desde k = 1 hasta k = n, hacer:

(a) Si $\left|D(i_k)\right| \leq d_a(i_k)$, hacer $u_{i_k} = 0$ y tomar el siguiente k.

(b) Si $\left|D(i_k)\right| > d_a(i_k)$, entonces

(b1) si $D(i_k) > 0$, hacer:

$$\left.\begin{array}{l} u_{i_k} = \min_j \tilde{c}_{i_k j} \\ \tilde{c}_{i_k j} \leftarrow \tilde{c}_{i_k j} - u_{i_k} \\ \tilde{c}_{j i_k} \leftarrow \tilde{c}_{j i_k} + u_{i_k} \\ \text{tomar el siguiente k.} \end{array}\right\} \forall j \text{ incidente con } i_k$$

(b2) si $D(i_k) < 0$, hacer:

$$\left.\begin{array}{l} u_{i_k} = -\min c_{ji_k} \\ \tilde{c}_{i_k j} \leftarrow \tilde{c}_{i_k j} - u_{i_k} \\ \tilde{c}_{ji_k} \leftarrow \tilde{c}_{ji_k} - u_{i_k} \\ \text{tomar el siguiente } k \end{array}\right\} \forall j \text{ incidente con } i_k$$

Al final de esta Subsección ofreceremos un ejemplo de aplicación de este algoritmo heurístico al grafo de la Figura 4.2. Mientras tanto, seguimos comentando las restantes etapas del procedimiento general de obtención de la cota inferior a partir de (RL_1).

La Etapa 2, requiere el cálculo de todos los caminos más cortos en G', con los costes modificados $\tilde{c}_{ij}$ entre todos los vértices de grado impar de G. Esta Etapa, ha sido realizada con la aplicación iterativa del algoritmo de Dijkstra, que, como se comentó en la Sección 1, es de tipo polinómico.

La Etapa 3, requiere el cálculo de un 1-acoplamiento de coste mínimo, para lo que se ha utilizado el algoritmo propuesto por Edmonds en [17].

Por último en la Etapa 4, se calcula el valor de la cota inferior sumando el coste del matching M* obtenido en la Etapa 3 y el término K(u) obtenido a partir de los multiplicadores u por el algoritmo heurístico anteriormente descrito. Representaremos por LBMAT al valor de la cota inferior obtenida a partir de (RL_1).

Ejemplo para la obtención de LBMAT.

Consideremos de nuevo el grafo de la Figura 4.2, sobre el que vamos a realizar las etapas del procedimiento general de obtención de la cota inferior al valor óptimo del problema (P) producida por (RL_1).

El procedimiento general aplicado a este grafo es descrito seguidamente por etapas.

Etapa 1: Cálculo de los multiplicadores u.

Para cada vértice i calculamos su correspondiente D(i), al igual que los grados de los vértices respecto de las aristas de G que representaremos por $d_a(i)$, lo cual se muestra en la Tabla 5.1.

Vértice i	D(i)	$d_a(i)$
1	-1	1
2	0	0
3	2	3
4	0	2
5	-1	1
6	-2	2
7	-1	3
8	0	3
9	1	4
10	-2	2
11	4	2
12	0	0
13	1	3
14	-1	1

Tabla 5.1

Podemos observar que únicamente hay un vértice i que satisface la desigualdad $|D(i)| > d_a(i)$, por lo que el algoritmo heurístico para calcular los multiplicadores asigna el valor cero a todos los asociados a los vértices del grafo, excepto al vértice 11, para el que tenemos que $D(11) = 4 > 0$, y el multiplicador u_{11} se calcula hallando el mínimo coste de los arcos que salen del vértice 11 junto con los de las aristas incidentes con él, esto es:

$$u_{11} = \min_j c_{11j} = \min \{4,8\} = 4$$

ahora, procedemos a calcular los costes modificados de los arcos y aristas en el grafo G, donde en este caso solamente son modificados los costes de los arcos y aristas incidentes con el vértice 11.

Respecto a los arcos que entran en el vértice 11 obtenemos los costes modificados:

$$\tilde{c}(5,11) = c(5,11) + u_{11} - u_5 = 3+4-0 = 7$$

$$\tilde{c}(7,11) = c(7,11) + u_{11} - u_7 = 1+4-0 = 5$$

$$\tilde{c}(12,11) = c(12,11) + u_{11} - u_{12} = 5+4-0 = 9$$

de esta forma hemos incrementado en cuatro unidades a cada uno de los costes de los arcos que entran en el vértice 11.

En lo concerniente a los costes modificados de las aristas incidentes con el vértice 11, obtenemos las parejas de costes siguientes:

para la arista $\{6,11\} = \begin{cases} \tilde{c}(6,11) = c(6,11) + u_{11} - u_6 = 4+4-0=8 \\ \tilde{c}(11,6) = c(6,11) - u_{11} + u_6 = 4-4+0=0 \end{cases}$

para la arista $\{10,11\} = \begin{cases} \tilde{c}(10,11) = c(10,11) + u_{11} - u_{10} = 8+4-0=12 \\ \tilde{c}(11,10) = c(10,11) - u_{11} + u_{10} = 8-4+0=4 \end{cases}$

todos los demas costes permanecen invariables, dando origen al grafo G' no dirigido que se muestra en la Figura 5.1

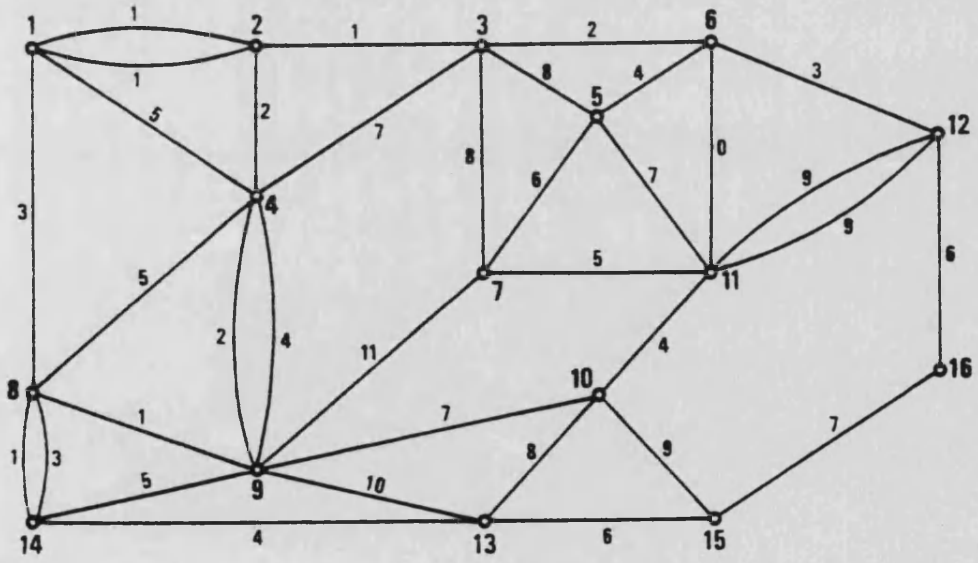


Figura 5.1: Grafo G' con los costes modificados.

Etapa 2: Cálculo de los caminos más cortos entre los vértices impares de G'.

En G' tenemos cuatro vértices impares 3,8,9,15, entre los que calculamos los caminos más cortos por medio del algoritmo de Dijsktra, y puesto que G' es un grafo no dirigido el c.m.c(i,j) = c.m.c(j,i) para todo (i,j), por lo que la matriz de distancias es simétrica

	3	8	9	15
3	-	5	5	15
8	5	-	1	11
9	5	1	-	15
15	15	11	15	-

Etapa 3: Cálculo de 1-acoplamiento de mínimo coste.

Aplicando el algoritmo de Edmonds, obtenemos el 1-acoplamiento mínimo M* que contiene los caminos más cortos que unen a los vértices 3 con el 9, y 8 con el 15:
P(3,9) = (3,2),(2,4),(4,9) de coste 5.
P(8,15)= (8,14),(14,13),(13,15) de coste 11.

Podemos observar que caminos posibles en el grafo G' pueden no serlo en el grafo original G, puesto que las direcciones de los arcos en el grafo original son ignoradas en el grafo G', por lo que realizando las oportunas duplicaciones en los arcos y aristas que forman los caminos P(3,9) y P(8,15), obtenemos un grafo par.

El valor del acoplamiento es en este caso:
c(M*) = c.m.c(3,9) + c.m.c(8,15) = 5+11 = 16.

Etapa 4: Cálculo de LBMAT.

LBMAT = v(PR(x,y,u)) = v(PR(x,y',u)) = c(M*) + K(u) donde

$$K(u) = \sum_{L_d(u)} \hat{c}_{ij} + \sum_{i} D(i)u_i + \sum_{A} c_{ij}$$

Por la definición de $L_d(u)$ tenemos que :

$\sum_{L_d(u)} \hat{c}_{ij} = 76$, y por otro lado,

$\sum_i D(i)u_i = 16$

$\sum_A c_{ij} = 82$, lo que da un valor total para K(u)=174.

Por último, el valor final para la cota inferior obtenida por la relajación de las restricciones de simetría, y que representaremos por LBMAT, es:

$$LBMAT = c(M^*) + K(u) = 16 + 174 = 190.$$

que para este problema, supuso el valor óptimo del problema (P), encontrado en el nudo 34 del árbol de ramificación del algoritmo de Branch and Bound que será explicado en la Sección 6. Al final de la Sección 5, ofrecemos un cuadro comparativo entre las cotas inferiores producidas por las tres relajaciones que hemos testado sobre 35 problemas, así como su alejamiento del valor óptimo de cada problema.

5.3 LA RELAJACION LAGRANGIANA DE LAS RESTRICCIONES DE OBLIGATORIEDAD

Esta tercera relajación, ha sido la utilizada para la obtención de cotas inferiores en los nudos internos del árbol de Branch and Bound correspondiente al algoritmo exacto de resolución del (MCPP) que se explicará en la Sección 6, las causas de esta elección son, entre otras, la mayor facilidad para encontrar soluciones posibles respecto a la relajación de las restricciones de simetría, y la mayor rapidez computacional frente a la relajación lineal.

5.3.1 Formulación del problema relajado PR(x,y,λ)

Según vimos en la Sección 1, podemos relajar lagrangianamente las restricciones de obligatoriedad (2) del problema (P), asociando un escalar λ_{ij} a cada una de las restricciones $y_{ij} + y_{ji} \geq 1$, siendo y_{ij} e y_{ji} la pareja de variables asociadas a la arista {i,j} del grafo mixto original G, obteniendo el problema de minimización PR(x,y,λ) que se formula a continuación:

$$(PR(x,y,\lambda)) \quad \text{Min} \Big\{ \mathcal{L}_2(x,y,\lambda) = \sum_A c_{ij}x_{ij} + \sum_L c_{ij}(y_{ij}+y_{ji}) - \sum_L \lambda_{ij}(y_{ij}+y_{ji}-1) + \sum_A c_{ij} \Big\}$$

sujeto a: (1'),(3) y (4).

Considerando que las restricciones relajadas son desigüaldades del tipo ($\geq$), los multiplicadores λ_{ij} deben ser no negativos para cualquier arista {i,j}, de esta forma obtenemos, para cada vector λ, una cota inferior al valor óptimo del problema (P) representado por $v(PR(x,y,\lambda))$.

Agrupando convenientemente los términos de la función lagrangiana $\mathcal{L}_2(x,y,\lambda)$, obtenemos la expresión:

$$\mathcal{L}_2(x,y,\lambda) = \sum_A c_{ij}x_{ij} + \sum_L (c_{ij}-\lambda_{ij})(y_{ij}+y_{ji}) + \sum_L \lambda_{ij} + \sum_A c_{ij} \quad (5.6)$$

donde para un vector λ dado, el tercer y cuarto términos de -- (5.6), son constantes.

Comparando $\mathcal{L}_2(x,y,\lambda)$ con la función objetivo del problema (P), observamos cierto parecido entre ellas, eliminando los términos constantes en ambas, pues simplemente difieren en los costes asociados a los pares de variables asociadas a las aristas de G. Los nuevos costes transformados, que representaremos por $\tilde{c}_{ij} = c_{ij}-\lambda_{ij}$, son menores o iguales que los originales, pudiendo, en principio, tomar valores negativos. Los multiplicadores λ_{ij} representan la penalización que se le asigna a cada arista $\{i,j\}$, en este caso dicha penalización favorece el que alguna de las variables asociadas a ella, sea -- distinta de cero, en sustitución de las restricciones de obligatoriedad relajadas. Finalizando, diremos que hemos sustituido las restricciones (1) por sus equivalentes (1'), para obtener una formulación semejante a la del problema de flujo de -- coste mínimo, que se comentó en la Sección 1.

5.3.2 La resolución de PR(x,y,λ)

Dado un vector de multiplicadores $\lambda \geq 0$, podemos - obtener la solución óptima de PR(x,y,λ), resolviendo un problema de flujo de mínimo coste, definido por los términos independientes de las restricciones (1'), asociadas a cada vértice de G, que indican si el vértice i es una fuente (si $D(i) > 0$), o - es un sumidero (si $D(i) < 0$), o un vértice intermedio (si $D(i) = 0$). De esta forma, cada variable x_{ij}, representa el flujo que ha circulado por el arco (i,j) incurriendo en un coste c_{ij} por cada unidad de flujo, mientras que las variables y_{ij}, y_{ji}, representan el número de unidades de flujo transportadas a través de la arista $\{i,j\}$, en una y otra dirección respectivamente, incurriendo en un coste modificado $\tilde{c}_{ij}$ por unidad. El grafo G^f sobre el que se realiza el flujo, está compuesto por todos los los arcos de G con sus costes originales, junto con dos arcos, en direcciones opuestas, en sustitución de cada una de las aristas de G, con costes modificados $\tilde{c}_{ij}$, que valen lo mismo independientemente de la dirección, por último se supone que la

capacidad de cada arco de G^f es infinita.

La solución de $PR(x,y,\lambda)$, proporciona un conjunto de valores enteros $\bar{x}_{ij}$, $\bar{y}_{ij}$, que representan el número de copias del arco (i,j), o de apariciones de la arista {i,j} en dirección de i a j, que se deben añadir al grafo inducido por los arcos de G, de forma que el grafo transformado G_t, sea simétrico. Por otra parte, la solución del problema $PR(x,y,\lambda)$, proporciona una solución al problema (P) cuando en G_t hay al menos un arco (i,j) ó (j,i), en representación de cada una de las aristas de G, puesto que en este caso G_t sería un grafo euleriano asignado de G. Por el contrario, si existen aristas en G por las que no ha circulado flujo (i.e: por sus arcos correspondientes en G_t), existirán restricciones relajadas que no se satisfacen, lo cual suscita la idea de penalizar más a las aristas por las que no ha circulado flujo, aumentando el valor del multiplicador asociado, facilitando el que un nuevo flujo con los costes modificados, utilice todas las aristas de G.

Dada una solución de $PR(x,y,\hat{\lambda})$, con $\hat{\lambda}$ dado, que proporcione una solución de (P), sabemos que es ε-óptima, siendo el valor de $\varepsilon = \sum_L \hat{\lambda}_{ij}(y_{ij}+y_{ji}-1)$, lo cual debe interpretarse como que una penalización excesiva a los costes de las aristas de G, puede proporcionar soluciones posibles pero que se alejan del valor óptimo a medida que van aumentando los valores de los multiplicadores λ_{ij}. En consecuencia, estrategias de selección de multiplicadores λ's, que penalicen excesivamente a los costes de las aristas, producirán normalmente "buenas" cotas inferiores, pero "malas" soluciones posibles al problema (P), este razonamiento sugiere una elección de multiplicadores con valores "intermedios", con objeto de alcanzar simultáneamente ambos objetivos.

Proposición 5.5

El problema $PR(x,y,\lambda)$, tiene la propiedad de la-Integridad.

Demostración

En efecto, los coeficientes de las restricciones (1'), corresponden a la matriz de incidencia del grafo G^f que-es totalmente unimodular (ver Garfinkel(1972),p.73), y por lo-tanto todas las soluciones posibles básicas de la relajación - lineal de $PR(x,y,\lambda)$, son enteras, lo cual implica que:

$$v(PR(x,y,\lambda)) = v(\overline{PR(x,y,\lambda)})$$

como queríamos demostrar.●

Comentario 5.2

En virtud de la Proposición anterior, las res---tricciones de integridad (4) de $PR(x,y,\lambda)$, pueden ser eliminadas sin alterar su valor óptimo, por lo que el algoritmo del simplex podría ser utilizado para su resolución, no obstante, cualquiera de los algoritmos existentes para resolver proble--mas de flujos de coste mínimo, como el presentado en Ford y Fulkerson (1962), o el algoritmo "out of kilter" son de hecho más eficientes (ver Bazaraa y Jarvis (1977)).

Proposición 5.6

(a) La mejor cota inferior producida por el problema $PR(x,y,\lambda)$, variando λ, es igual a la producida por la relaja--ción lineal, es decir: $\max_{\lambda} v(PR(x,y,\lambda)) = v(\overline{P})$

(b) sea λ^*, un vector de multiplicadores óptimo, esto es: $v(PR(x,y,\lambda^*))=v(\overline{P})$. Entonces, λ^* coincide con el vector de soluciones duales óptimas parciales γ^*.

Demostración

Ambos resultados son consecuencia del Teorema 1.7 comentado en la Introducción.●

Comentario 5.4

La Proposición 5.6, nos dice que la mejor cota -

inferior que podemos hallar usando la relajación de las restricciones de obligatoriedad (RL_2), variando los multiplicadores λ, nunca puede superar el valor obtenido por la relajación lineal (tratada en la Subsección 5.1). Este hecho podría llevar a pensar que (RL_2) no es una relajación aconsejable puesto que está dominada por otra relajación, sin embargo, la mayor rapidez computacional para el cálculo de valores cercanos a $v(\overline{P})$, utilizando el método del subgradiente con un número razonable de iteraciones, le da validez inmersa en un algoritmo de Branch and --- Bound.

Queremos recalcar también, que la sucesión de λ's generados por el método del subgradiente, pueden ser restringida a que cada componente de sus elementos, no supere al coste - c_{ij} de la arista asociada, puesto que las soluciones posibles - del problema dual ($D\overline{P}$), y en particular la óptima, cumple que para toda arista $l=\{i,j\}$, se tiene que $0 \leqslant \gamma_l^* \leqslant c_{ij}$ (ver Proposición 5.1), y en consecuencia también $0 \leqslant \lambda_{ij} \leqslant c_{ij}$, sin pérdida de optimalidad. Este resultado se expone en la siguiente Proposición.

Proposición 5.7

$$\text{máximo } v(PR(x,y,\lambda)) = v(\overline{P}).$$
$$0 \leqq \lambda \leqq c$$

Demostración

Es evidente a raiz del Comentario 5.4. ●

Antes de la utilización del método del subgradiente, queremos presentar la forma de elegir unos "buenos" multiplicadores iniciales, a partir de la solución óptima del problema $PR(x,y,\lambda)$ con $\lambda=0$.

5.4.3 Elección de "buenos" multiplicadores iniciales para el método del subgradiente.

Sea (x^o,y^o), la solución óptima de $PR(x,y,0)$ (i.e: con $\lambda=0$). Construimos los siguientes conjuntos:

$$L_0 = |\{i,j\} \varepsilon L: y_{ij}^o + y_{ji}^o = 0|$$

$$L_1 = |\{i,j\} \varepsilon L:\ y^o_{ij} + y^o_{ji} = 1|$$

$$L_2 = |\{i,j\} \varepsilon L:\ y^o_{ij} + y^o_{ji} \geq 2|$$

En L_0, se encuentran las aristas por donde no ha circulado ninguna unidad del flujo solución del problema relajado PR(x,y,0). En L_1, las aristas por las que exactamente ha circulado una unidad de flujo. En L_2, las aristas por las que han circulado al menos dos unidades de flujo (en cualquier dirección). Evidentemente, (x^o,y^o) será una solución posible, y en este caso también óptima por ser $\lambda=0$, si y solo si $L_0=\phi$, en caso contrario ($L_0 \neq \phi$), asignaremos los valores de los "buenos" multiplicadores iniciales de la forma siguiente:

$$\lambda^1_{ij} = c_{ij} \qquad \forall\{i,j\} \varepsilon L_0$$

$$\lambda^1_{ij} = [c_{ij}/2] \qquad \forall\{i,j\} \varepsilon L_1$$

$$\lambda^1_{ij} = 0 \qquad \forall\{i,j\} \varepsilon L_2$$

Esta asignación, está de acuerdo con lo dicho en la Subsección 5.4.2, acerca de los dos objetivos que debían satisfacer los multiplicadores λ's, puesto que se penalizan al máximo los costes de las aristas no utilizadas por el flujo, reduciendo sus costes al valor cero, mientras que las aristas por donde ha circulado una unidad de flujo se les asigna una penalización intermedia $[c_{ij}/2]$ donde $[\cdot]$ representa la parte entera; por último al resto de las aristas no se les penaliza.

Supongamos que resolvemos el problema $PR(x,y,\lambda^1)$, siendo (x^1,y^1) la solución óptima de él. Definimos de forma paralela los conjuntos de aristas:

$$L^1_0 = |\{i,j\} \varepsilon L:\ y^1_{ij} + y^1_{ji} = 0|$$

$$L^1_1 = |\{i,j\} \varepsilon L:\ y^1_{ij} + y^1_{ji} = 1|$$

$$L^1_2 = |\{i,j\} \varepsilon L:\ y^1_{ij} + y^1_{ji} \geq 2|$$

Vamos a demostrar, que la cota inferior producida por el valor óptimo de $PR(x,y,\lambda^1)$, supera o iguala a la cota inferior trivial al valor óptimo del problema (P), obtenida por la suma de todos los costes del grafo G que representaremos por c(G).

Proposición 5.8

$$v(PR(x,y,\lambda^1) \geq c(G).$$

Demostración

En efecto, sea (x^1,y^1), la solución óptima del problema $PR(x,y,\lambda^1)$, tenemos que:

$$v(PR(x,y,\lambda^1) = \mathcal{L}_2(x^1,y^1,\lambda^1) = \sum_A c_{ij}x^1_{ij} + \sum_L (c_{ij}-\lambda^1_{ij})(y^1_{ij}+y^1_{ji}) +$$

$$+ \sum_L \lambda^1_{ij} + \sum_A c_{ij} = \text{(por la definición de } L^1_0, L^1_1 \text{ y } L^1_2)$$

$$= \sum_A c_{ij}x^1_{ij} + \sum_{L^1_1} (c_{ij} - [c_{ij}/2]) + \sum_{L^1_2} c_{ij}(y_{ij}+y_{ji}) +$$

$$+ \sum_{L^1_0} c_{ij} + \sum_{L^1_1} [c_{ij}/2] + \sum_A c_{ij} =$$

$$= \sum_L c_{ij} + \sum_A c_{ij} + \sum_A c_{ij}x^1_{ij} + \sum_{L^1_2} c_{ij}(y^1_{ij}+y^1_{ji}-1) =$$

$$= c(G) + \sum_A c_{ij}x^1_{ij} + \sum_{L^1_2} c_{ij}(y^1_{ij}+y^1_{ji}-1) \geq c(G). \bullet$$

Comentario 5.5

Hemos establecido una cota inferior al valor de $PR(x,y,\lambda^1)$, además la cota está ajustada (i.e: no se puede aumentar), puesto que para un grafo G que sea euleriano, tendríamos que v(P)=c(G). Notar que con $\lambda=0$, esta cota inferior no tiene porqué ser cierta, lo que justifica que, en general, λ^1 es mejor selección de multiplicadores que $\lambda=0$, este hecho también se refleja con los valores obtenidos de las cotas inferiores con $\lambda=0$ y $\lambda=\lambda^1$, respectivamente, sobre el total de los problemas testados. Ofreceremos al final de esta Subsección un análisis comparativo de las tres relajaciones.

A continuación ofrecemos un ejemplo de obtención de los valores óptimos de los valores de las cotas inferiores proporcionados por el problema PR(x,y,λ), con λ=0 y λ=λ^1, respectivamente. Estos valores serán representados por LBFLU1 y LBFLU2 para simplificar las referencias a ellos.

Ejemplo de obtención de las cotas inferiores producidas por (RL_2)

Consideremos de nuevo el grafo de la Figura 4.2, sobre el que vamos a calcular los valores LBFLU1 y LBFLU2.

Cálculo de LBFLU1

En este caso tenemos que $\lambda_{ij}=0$ $\{i,j\}\varepsilon L$, y que - por lo tanto, los costes modificados de las aristas del grafo ejemplo G son idénticos a los originales. Para resolver el problema de flujo de coste mínimo incumbente a la solución de PR(x,y,0), calculamos en primer lugar los términos D(i) para cada vértice i del grafo G, obteniendo los conjuntos de fuentes y sumideros con sus ofertas y demandas respectivas, reflejadas en la Tabla 5.2

Fuentes	D(i)	Sumideros	D(i)
3	2	1	-1
9	1	5	-1
11	4	6	-2
13	1	7	-1
		10	-2
		14	-1

Tabla 5.2

Resolviendo el problema de flujo de coste mínimo sobre el grafo G^f, representado en la Figura 5.2, en el que cada arista del grafo G ha sido sustituida por dos arcos dirigidos en sentidos opuestos, obtenemos los flujos óptimos distintos de cero que han circulado por los arcos de G^f reflejados en la Tabla 5.3.

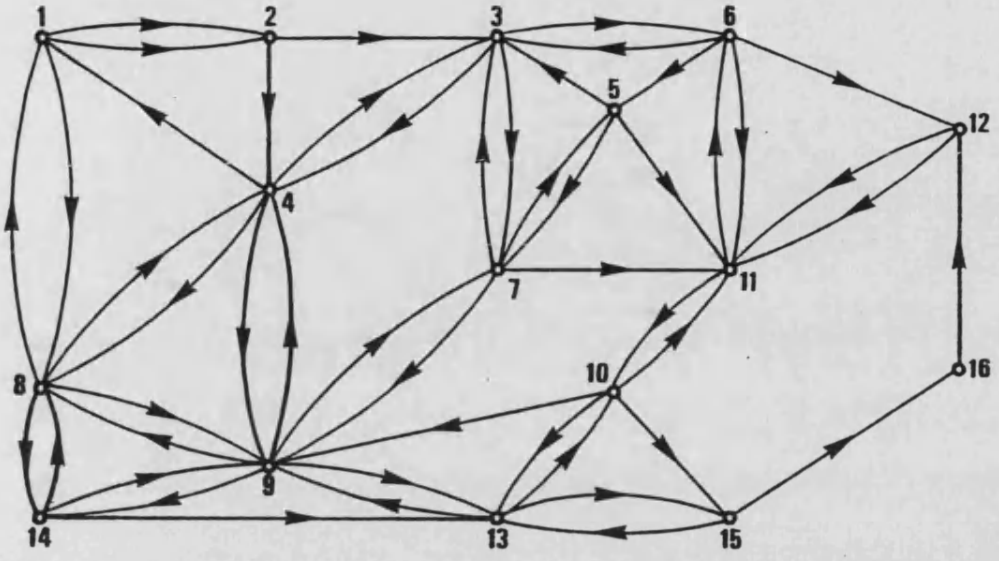


Figura 5.2: Grafo G^f sobre el que se realiza el flujo.

Arco	Flujo	Coste
(3 , 4)	1	7
(3 , 7)	1	8
(4 , 1)	1	5
(6 , 5)	1	4
(8 ,14)	1	1
(9 , 8)	1	1
(11, 6)	3	12
(11,10)	1	8
(13,10)	1	8
T O T A L....		54

Tabla 5.3: Flujo para el cálculo de LBFLU1.

El coste total del flujo circulado ha sido de 54 y puesto que la suma de los costes de los arcos del grafo G es de 82, obtenemos el valor de la cota inferior producida con $\lambda=0$.

LBFLU1 = 54 + 82 = 136

El valor de LBFLU1 corresponde al coste del grafo simétrico, producido por la duplicación de arcos y apariciones de aristas en uno y otro sentido, obtenidos por la solución óptima del flujo realizado. Este grafo se muestra en la Figura 5.3.

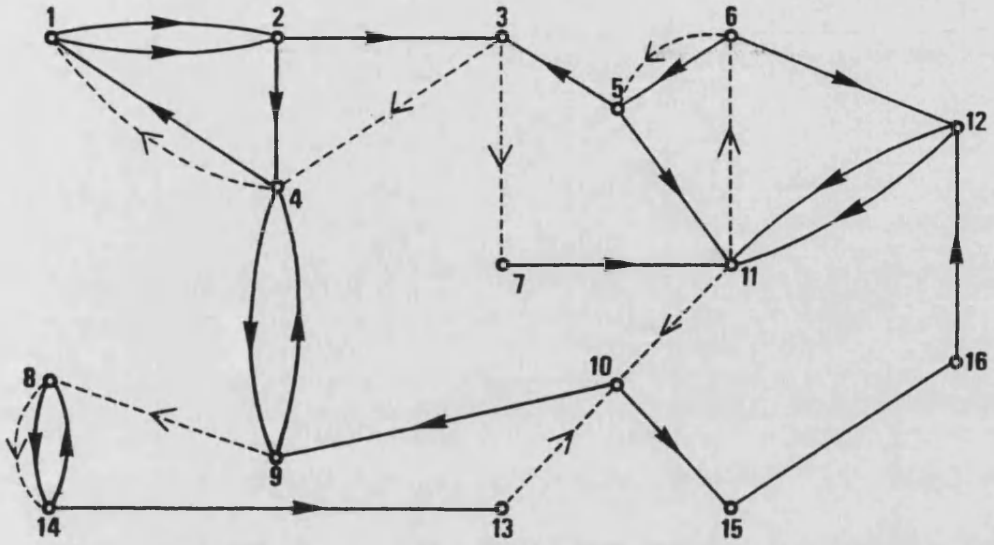


Figura 5.3: Grafo simétrico producido por el flujo
Arcos originales de G ——→
Arcos producidos por el flujo ----→----

Comparando la Figura 4.2 con la Figura 5.2, podemos observar que existen varias aristas del grafo G por donde no ha circulado flujo, en consecuencia no se produce una solución óptima del problema (P) y ni tan siquiera se obtiene una cota que supere el coste del grafo G que en nuestro ejemplo es de 160. Seguidamente pasamos a calcular el valor de LBFLU2.

Cálculo de LBFLU2

Utilizando la solución del flujo óptimo obtenido en el cálculo de LBFLU1, construimos los conjuntos de aristas L_0, L_1 y L_2, que como ya vimos, representan los conjuntos de aristas por donde no ha circulado flujo, o ha pasado exactamente una unidad, o han pasado al menos dos, respectivamente. En nuestro ejemplo tenemos que:

$$L_0 = |\{1,8\},\{3,6\},\{4,8\},\{5,7\},\{7,9\},\{9,13\},\{9,14\},\{13,15\}|$$

$$L_1 = |\{3,4\},\{3,7\},\{8,9\},\{10,11\},\{10,13\}|$$

$$L_2 = |\{6,11\}|$$

Obteniendo los multiplicadores λ^1_{ij} siguientes:

$$\lambda^1_{1,8} = 3\ ,\ \lambda^1_{3,6} = 2,\ \lambda^1_{4,8} = 5,\ \lambda^1_{5,7} = 6,\ \lambda^1_{7,9} = 11,$$

$$\lambda^1_{9,13} = 10,\ \lambda^1_{9,14} = 5,\ \lambda^1_{13,15} = 6,\ \lambda^1_{3,4} = 3,\ \lambda^1_{3,7} = 4,$$

$$\lambda^1_{8,9} = 0,\ \lambda^1_{10,11} = 4,\ \lambda^1_{10,13} = 4,\ \lambda^1_{6,11} = 0.$$

obteniendo el grafo con costes modificados de la Figura 5.4

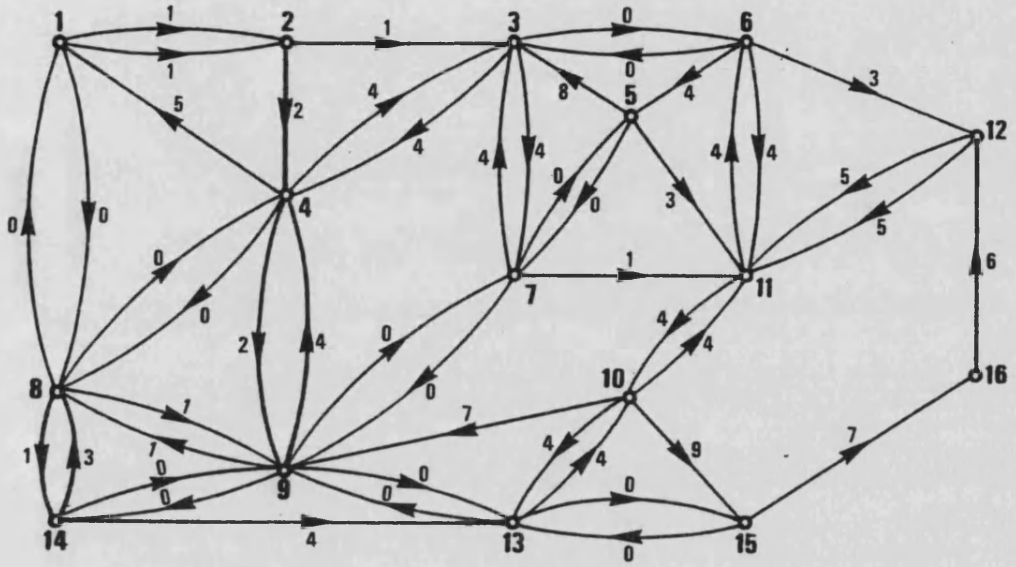


Figura 5.4: Grafo G^f con costes modificados por λ^1.

sobre el que realizamos un nuevo flujo, con el mismo conjunto de fuentes y sumideros, sin alterar sus ofertas y demandas respectivas. El flujo óptimo obtenido, se presenta en la Tabla 5.4.

Arco	Flujo	Coste
(3 , 4)	1	4
(3 , 6)	1	0
(4 , 8)	1	0
(6 , 5)	1	4
(8 , 1)	1	0
(9 , 7)	1	0
(9 ,14)	1	0
(11, 6)	2	8
(11,10)	2	8
(13, 9)	1	0
T O T A L.....		24

Tabla 5.4: Flujo para el cálculo de LBFLU2.

El coste total del flujo anterior es de 24, siendo por lo tanto 30 unidades más barato que el correspondiente para el cálculo de LBFLU1, sin embargo, obtenemos una cota inferior cuyo valor es:

$$LBFLU2 = \sum_{A} c_{ij} + v(Flujo) + \sum_{L} \lambda^{1}_{ij} =$$
$$= 82 + 24 + 63 = 169.$$

Observar que el flujo obtenido para el cálculo de LBFLU2, al igual que el obtenido para el cálculo de LBFLU1, no proporciona una solución posible al problema (P), por no utilizar todas las aristas del grafo original G.

A continuación, damos una breve descripción del método del subgradiente que proporcionó, tras 50 iteraciones, el mejor valor LBSUB = 177.7, en este ejemplo.

Notación 5.3

MAXIT representa el número de iteraciones máximo que se realizan en el método del subgradiente.

(x^{ν}, y^{ν}) representa la solución óptima del problema $PR(x,y,\lambda^{\nu})$.

t_{ν} es el "paso del subgradiente".

MAXCONT es el número máximo de iteraciones manteniendo el último t_{ν}, sin haber mejorado la cota inferior.

v(H) representa el valor de la solución posible obtenida por el heurístico descrito en la Subsección 4.1, para obtener la cota superior inicial al valor de (P).

LBSUB indica el mejor valor hallado durante todo el proceso del método del subgradiente.

5.4.3 Utilización del método del subgradiente aplicado a PR(x,y,λ).

Describimos a continuación, el método del subgradiente utilizado, para obtener cotas inferiores al problema (P), utilizando la relajación (RL_2):

Paso 1: (Inicialización)

Hacer $\nu = 1$. Seleccionar t_1: $0 < t_1 \leqslant 2$. Inicializar: MAXCONT de forma que $1 < MAXCONT \leqslant MAXIT$, LBSUB = 0, CONT = 1, $S^1 = \sum_L (y^1_{ij} + y^1_{ji} - 1)$. LBSUB = 0, CONT = 1.

Paso 2: (Cálculo de los nuevos multiplicadores)

(a) Si CONT = MAXCONT + 1, hacer $t_\nu = t_\nu/2$. Si no ir a (b).

(b) Si $S^\nu = 0$, (x^ν, y^ν) es una solución posible de (P) con valor $\mathcal{L}_2(x^\nu, y^\nu, \lambda^\nu)$. STOP. En caso contrario ir a (c).

(c) Calcular: $$\theta^{\nu+1} = \frac{t_\nu (v(H) - \mathcal{L}_2(x^\nu, y^\nu, \lambda^\nu))}{\sum_L (y^\nu_{ij} + y^\nu_{ji} - 1)^2}$$

hacer $\lambda^{\nu+1}_{ij} = \max\{0, \min\{c_{ij}, \lambda^\nu_{ij} - \theta^\nu S^\nu\}\}$ $\forall \{i,j\} \in L$.

Paso 3: (Cálculo de la nueva solución)

Hacer $\nu = \nu + 1$. Obtener la solución óptima (x^ν, y^ν) del problema $PR(x,y,\lambda^\nu)$.

Hacer LBSUB = $\max(LBSUB, \mathcal{L}_2(x^\nu, y^\nu, \lambda^\nu))$.

Si $\nu = MAXIT$, el valor de LBSUB es el mejor obtenido. STOP. En caso contrario:

(i) Si $LBSUB = \mathcal{L}_2(x^\nu, y^\nu, \lambda^\nu)$, hacer CONT = 1. Ir al Paso 2.

(ii) Si $LBSUB > \mathcal{L}_2(x^\nu, y^\nu, \lambda^\nu)$, hacer CONT = CONT + 1. Ir al Paso 2.

Las gráficas siguientes muestran la situación de las tres cotas inferiores, que han sido tratadas en esta Sección, entre sí, y respecto al valor óptimo del problema (P). Para todos estos problemas se tomó MAXIT=50, MAXCONT=5, $t_1 = 0.75$.

En el eje de abscisas se representa el número de iteración del método del subgradiente, donde cada 5 iteraciones se levanta una perpendicular,cuya altura refleja el mejor valor obtenido en ése grupo de iteraciones. Los dos primeros valores corresponden a las cotas LBFLU1 y LBFLU2.

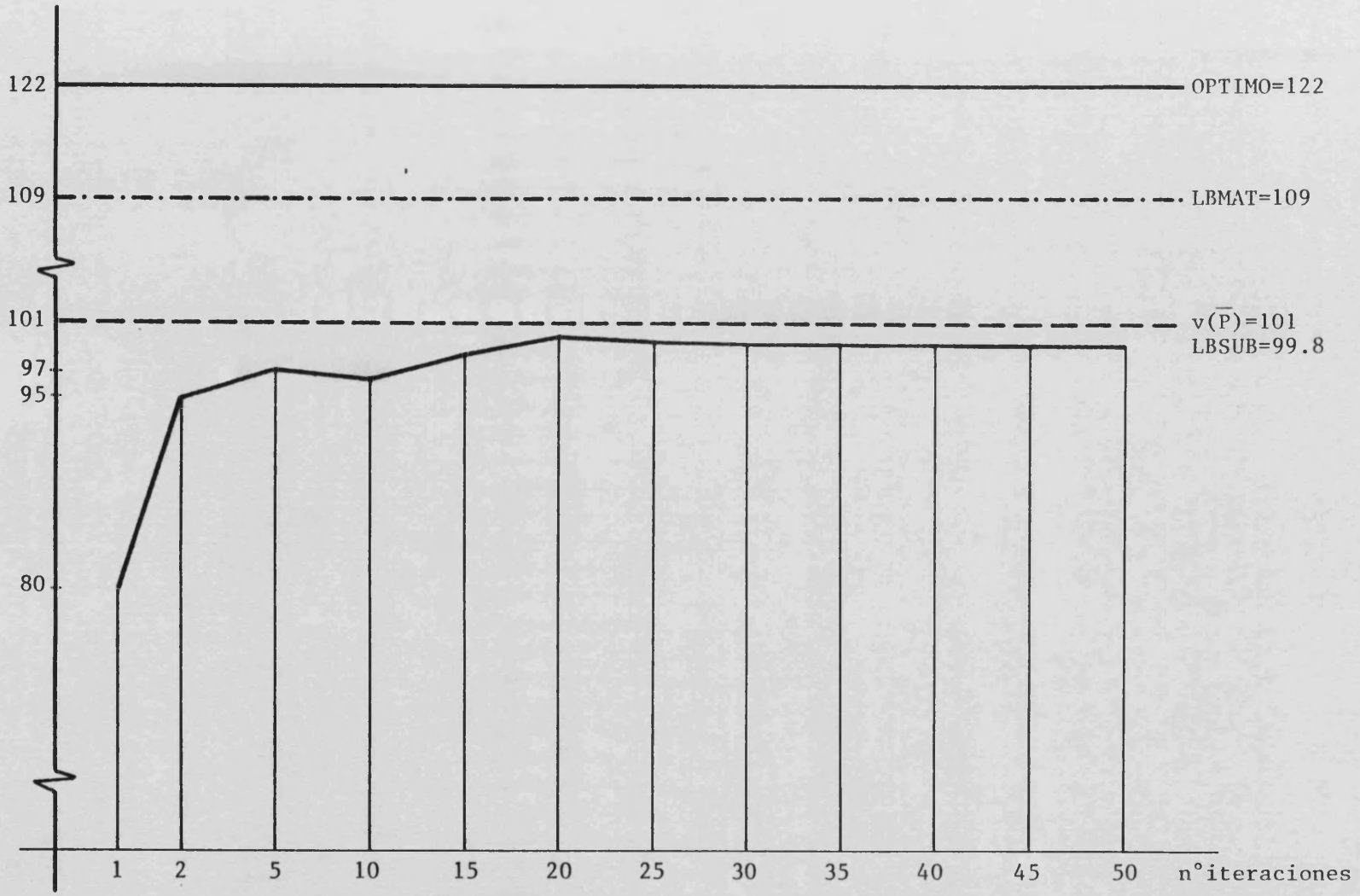


Figura 5.5: Problema 8

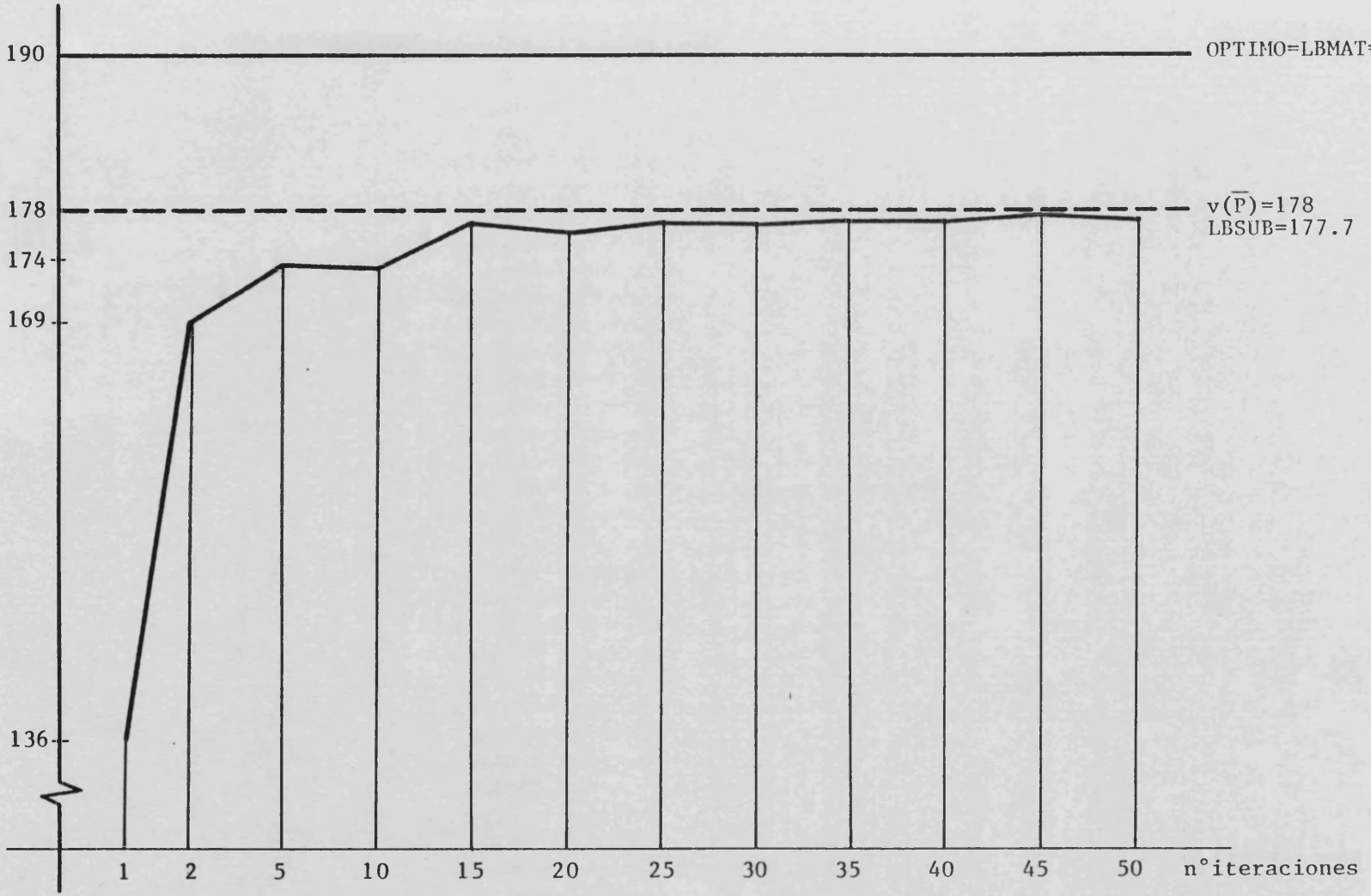


Figura 5.6: Problema 12

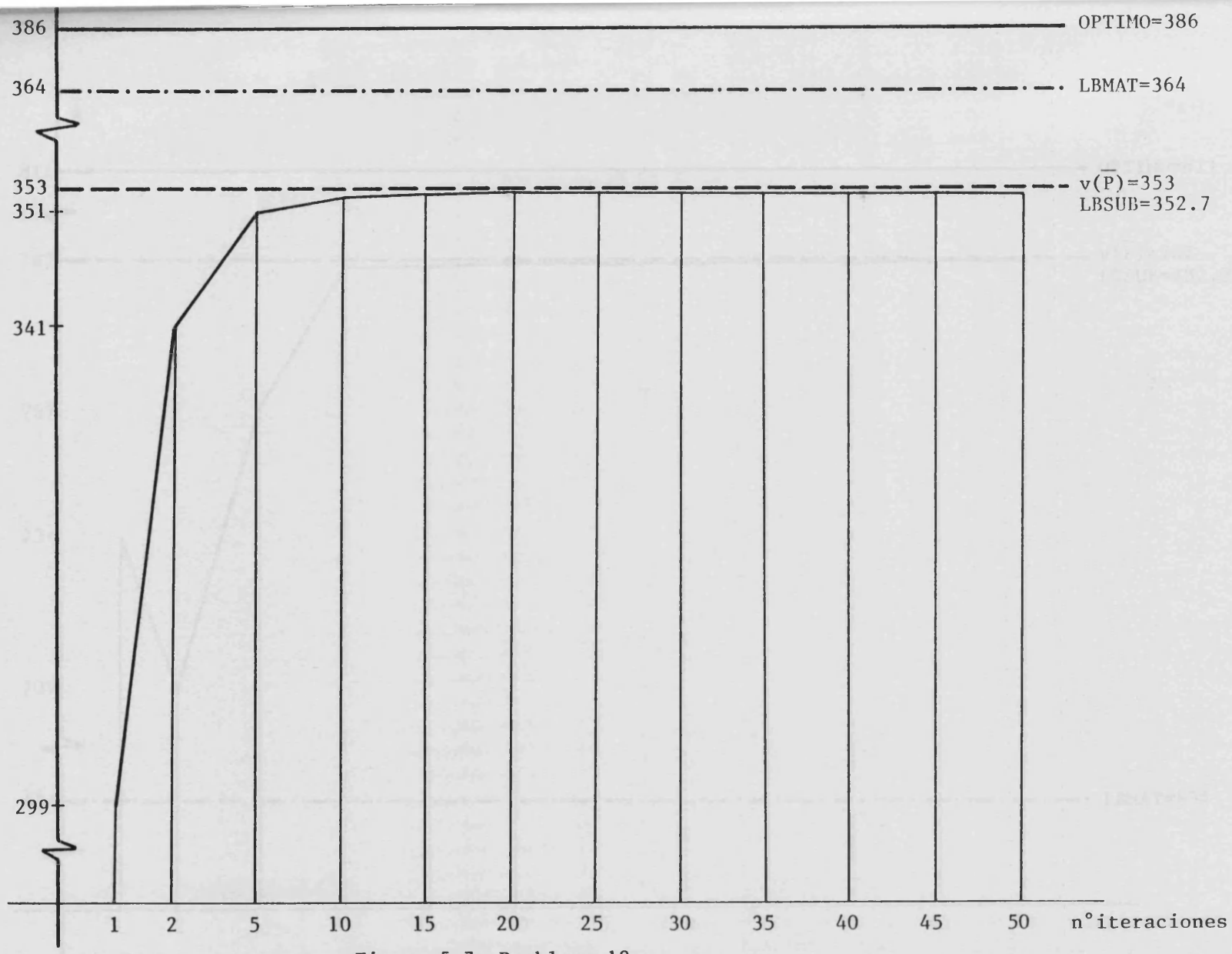


Figura 5.7: Problema 18

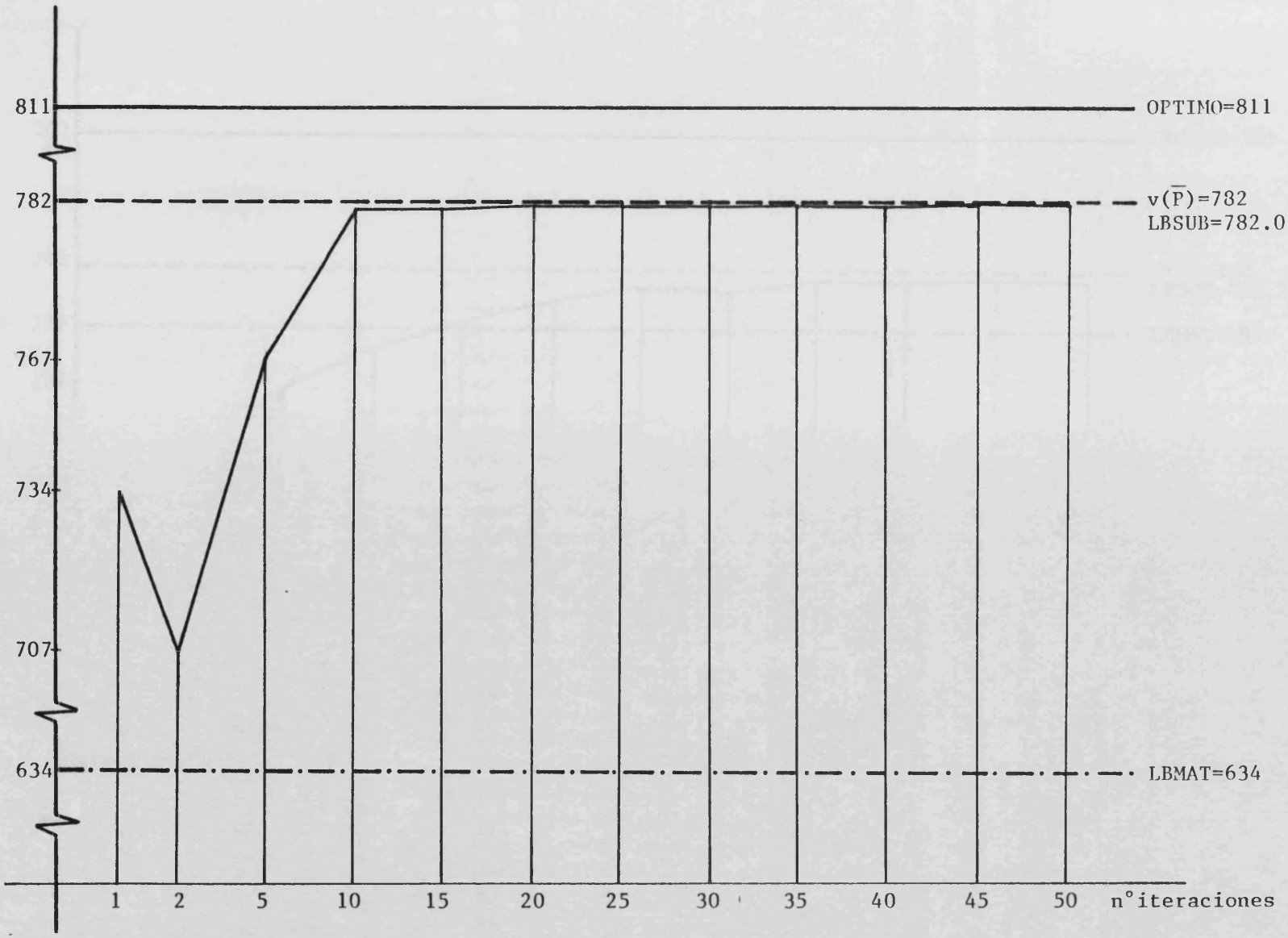


Figura 5.8: Problema 23

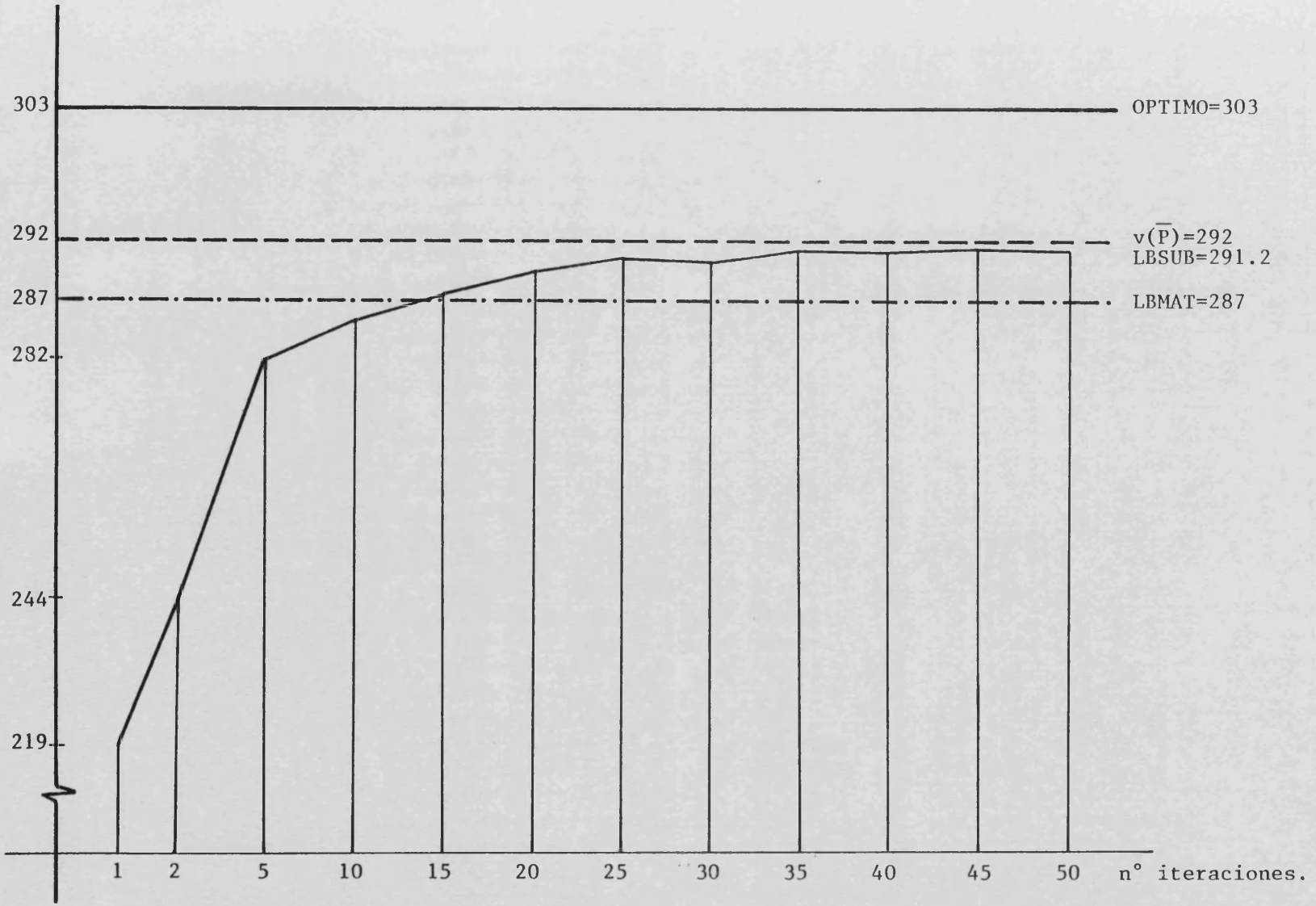


Figura 5.9: Problema 33.

Comentario 5.6

En las Figuras 5.5-5.9, podemos observar que el incremento de los valores obtenidos por el método del subgradiente, es mucho mayor en las primeras iteraciones, alcanzando valores muy cercanos a la cota inferior producida por la relajación lineal.

La Tabla 5.5 nos muestra el comportamiento del método del subgradiente sobre los treinta y cinco problemas testados, donde distinguimos los valores de LBFLU1 y LBFLU2, así como los mejores valores obtenidos tras veinte y cincuenta iteraciones del subgradiente, que representamos por LBSUB(20) y LBSUB(50), respectivamente.

Los incrementos medios porcentuales que se obtienen a partir de los resultados de la Tabla 5.5 son:

$$\overline{\Delta}(\text{LBFLU1} \rightarrow \text{LBFLU2}) = 8.6\%$$
$$\overline{\Delta}(\text{LBFLU2} \rightarrow \text{LBSUB(20)}) = 5.9\%$$
$$\overline{\Delta}(\text{LBSUB(20)} \rightarrow \text{LBSUB(50)}) = 0.18\%$$
$$\overline{\Delta}(\text{LBSUB(50)} \rightarrow v(\overline{P})) = 0.17\%$$

lo que significa que:

1°.- Los multiplicadores λ^1 proporcionan cotas inferiores que representan aproximadamente el 58%, en promedio, del incremento total durante todo el proceso del subgradiente, por lo que puede calificárseles de "buenos".

2°.- El incremento porcentual medio producido en las 20 primeras iteraciones es aproximadamente 32 veces mayor que el producido entre las iteraciones 21 y 50.

3°.- El incremento porcentual medio entre la iteración 51 y un número ilimitado de iteraciones del subgradiente, supone un acercamiento medio del 0.17% al valor máximo que podemos esperar, correspondiente al valor de la relajación lineal.

Los tres puntos anteriores, sugieren que un número de iteraciones máximo de 20, es más razonable que, por ejemplo,

50 ó más, para utilizar el método del subgradiente como productor de cotas inferiores imbuídas en el algoritmo de Branch and Bound que presentaremos en la siguiente Sección. Por otra parte, el tiempo medio de C.P.U para realizar 20 iteraciones del método del subgradiente, ha sido de 0.9 segundos , frente al de 2.45 segundos para resolver los problemas lineales con el FMPS (ver Comentario 4.1). A la ventaja de rapidez de cálculo, a -- cambio de un ligero detrimento de la optimalidad, hay que añadir la posibilidad de encontrar soluciones ε-óptimas para el-- (MCPP) durante el proceso iterativo del método del subgradiente; en los problemas 1 y 3, por ejemplo, se encontró la solución óptima del (MCPP), al encontrar la primera cota LBFLU1, - por lo que no se realizaron iteraciones del método del subgradiente, éstos dos problemas han sido excluídos del cálculo de los promedios anteriores, por esta razón.

En la Tabla 5.6 se presentan los valores obtenidos por las tres relajaciones en la colección de los problemas testados, junto con el C.P.U en segundos para la obtención delas mismas. En la columna 2.a, se encuentran los valores obtenidos a partir de (RLin), cuyo alejamiento medio respecto del valor óptimo de cada problema es del 6.4% aproximadamente, habiéndose obtenido el valor óptimo únicamente en dos ocasiones (problemas 1 y 3). Por otra parte, el alejamiento medio de las cotas producidas por (RL_1), ha sido ligeramente inferior al de (RLin), situándose en un 6.3% aproximadamente, esto sí, con un tiempo medio de resolución de 0.15 segundos de C.P.U, lo que representa una rapidez de cálculo 16 veces mayor que la Relajación Lineal. Además, este tiempo incluye el del cálculo de los "buenos" multiplicadores u, por medio del heurístico presentado en la Subsección 5.2, y así se pueden calificar puesto quese ha obtenido el valor óptimo en diez ocasiones, de un totalde 32 para los que éste es conocido. Por último, en las columnas 4.a y 4.b, se dan los mejores valores obtenidos de entre - LBFLU1 y LBFLU2 para cada problema, el alejamiento medio es, en este caso, aproximadamente del 11.5%, pero con un tiempo medio de resolución de 0.10 segundos de C.P.U, es decir aproxima

damente 24 veces más rápida de obtener que (RLin), y vez y media más rápida, en promedio, que (RL_1). Tomando como cota inferior la mejor de entre LBMAT,LBFLU1 y LBFLU2, para cada uno de los problemas, obtendríamos una cota inferior alejada, en valor medio, a tan sólo el 4.5% del valor óptimo, y con un tiempo medio de obtención de aproximadamente 0.25 segundos de C.P.U, es decir, aproximadamente diez veces más rápida que la relajación lineal. Este último hecho descarta practicamente a la relajación lineal como utilizable en un algoritmo de Branch and Bound para resolver el (MCPP), puesto que la combinación de dos relajaciones lagrangianas produce, en promedio, mejores cotas inferiores y además en un tiempo de computación menor. Esta última aserción no puede ser respaldada más que por la experiencia computacional, pues como se puede apreciar en la Tabla 5.6 existen problemas para los que el valor $v(\overline{P})$, ha superado al valor de LBMAT (ver problemas 34 y 35, por ejemplo) y recíprocamente (ver problemas 26 y 32, por ejemplo), pero como se verá en la Sección 6, el intento de obtener cotas inferiores aproximadas a los valores de la relajación lineal en los nudos internos del algoritmo de Branch and Bound para resolver el (MCPP), con la utilización del método del subgradiente, no produce más que un incremento excesivo del tiempo de C.P.U, que no compensa la mejoría de las cotas inferiores producidas.

Problema	LBFLU1	LBFLU2	LBSUB(20)	LBSUB(50)	$v(\overline{P})$
1	83	----	-------	-------	83
2	48	51	51.0	51.0	51
3	36	----	-------	-------	36
4	29	29	30.0	30.0	30
5	112	126	129.0	129.0	129
6	130	153	159.2	159.2	163
7	37	50	50.0	50.0	50
8	80	95	99.8	99.8	101
9	154	171	174.0	174.0	174
10	41	66	75.0	75.5	76
11	82	106	111.9	111.9	112
12	136	169	177.1	177.7	178
13	125	162	169.5	169.8	170
14	505	559	590.0	590.2	596
15	786	787	846.9	846.9	847
16	202	191	236.2	237.3	238
17	59	59	62.9	63.0	63
18	299	341	352.7	352.7	353
19	120	174	190.8	192.2	193
20	84	102	106.6	106.6	110
21	732	772	790.1	790.8	791
22	503	570	574.7	574.7	575
23	734	707	781.5	782.0	782
24	664	704	747.4	748.9	750
25	821	880	907.7	909.9	911
26	554	648	663.2	666.8	668
27	945	977	1015.8	1015.9	1016
28	1086	1132	1148.4	1151.2	1152
29	753	840	889.8	891.9	893
30	793	849	897.9	902.1	904
31	107	217	232.0	232.0	232
32	154	238	259.3	363.4	265
33	219	244	289.5	291.2	292
34	934	979	1086.4	1089.2	1090
35	1070	1076	1158.5	1160.8	1162

Tabla 5.5: Evolución del método del subgradiente.

1	2.a	2.b	3.a	3.b	4.a	4.b	5
1	83*	1.17	72	0.02	83*	0.003	83
2	51	0.98	53	0.004	51	0.01	54
3	36*	1.20	36*	0.01	36*	0.004	36
4	30	1.41	30	0.01	29	0.02	31
5	129	1.45	140*	0.02	126	0.02	140
6	163	1.58	174	0.05	153	0.04	181
7	50	1.27	55*	0.004	50	0.01	55
8	101	1.42	109	0.02	95	0.02	122
9	174	1.84	158	0.04	171	0.03	191
10	76	1.09	89*	0.02	66	0.01	89
11	112	1.30	122*	0.02	106	0.02	122
12	178	1.59	190*	0.02	169	0.04	190
13	170	1.90	184*	0.02	162	0.05	184
14	596	2.51	569	0.14	559	0.10	630
15	847	3.24	790	0.54	787	0.43	881
16	238	2.13	241	0.03	202	0.06	253
17	63	2.50	62	0.12	59	0.08	66
18	353	2.46	364	0.12	341	0.08	386
19	193	1.64	211*	0.02	174	0.03	211
20	110	1.49	107	0.01	102	0.02	119
21	791	3.82	728	0.30	772	0.22	814
22	575	2.73	559	0.16	570	0.11	601
23	782	2.75	634	0.16	739	0.09	811
24	750	3.10	677	0.20	704	0.14	791
25	911	3.51	890	0.18	880	0.17	964
26	668	3.36	688	0.17	648	0.13	704
27	1016	3.50	898	0.41	977	0.15	1040
28	1152	4.37	1056	0.34	1132	0.26	1162
29	893	3.84	795	0.41	840	0.25	---
30	904	4.15	798	0.41	849	0.31	---
31	232	1.87	257*	0.03	217	0.005	257
32	265	2.19	308*	0.05	238	0.06	308
33	292	2.18	287	0.06	244	0.05	303
34	1090	4.80	1022	0.54	979	0.29	---
35	1162	4.62	1050	0.45	1076	0.24	1202

Tabla 5.6: Comparación relajaciones.

1. Número de problema.
2.a Valor obtenido por (RLin).
2.b Tiempo de C.P.U en segundos para la obtención de (RLin).
3.a Valor obtenido por (RL_1).
3.b Tiempo de C.P.U en segundos para la obtención de (RL_1).
4.a Valor obtenido por (RL_2) con el máximo valor de LBFLU1 y LBFLU2.
4.b Tiempo de C.P.U en segundos para la obtención de LBFLU1 y LBFLU2.
5.Valor óptimo del problema.

S E C C I O N 6

ALGORITMO DE BRANCH AND BOUND

Los algoritmos de Branch and Bound han sido, durante las dos últimas décadas, los más popularmente utilizados para la resolución de problemas de Optimización Combinatorial, y problemas de Programación Lineal Entera. En 1963 Little y otros, ofrecen la primera experiencia computacional de un algoritmo de éste tipo, para resolver el problema del agente viajero (TSP), y siete años más tarde Held y Karp [34], introducen mejoras en el cálculo de las cotas inferiores utilizando la Relajación Lagrangiana, logrando excelentes resultados. Otros problemas conocidos, como los de scheduling, localización, set covering, y un largo etcétera, han sido tratados utilizando el método de Branch and Bound durante los últimos cinco años.

En esta Sección, presentamos la descripción de un algoritmo de Branch and Bound para resolver el problema del cartero chino en un grafo mixto (MCPP). Los aspectos fundamentales de los que consta éste, se concretan en:

1.- Separación en subproblemas candidatos.

2.- Estrategia de ramificación.

3.- Eliminación de subproblemas.

Por último, al final de la Sección presentamos un resumen de los resultados computacionales más interesantes correspondientes a una colección de problemas test, generados aleatoriamente.

6.1 SEPARACION EN SUBPROBLEMAS CANDIDATOS.

Siguiendo la metodología descrita por Geoffrion-[25], la estrategia de separación de todo algoritmo de Branch-and Bound, debe proporcionar una partición del conjunto de soluciones posibles del problema original a resolver, originando una colección de subproblemas que son llamados "subproblemas--candidatos", los cuales pueden, a continuación, ser eliminados al determinar su solución óptima, o si se verifica,que no pueden producir soluciones mejores al problema original,que una supuestamente conocida. Por el contrario, si un subproblema no se elimina, se particiona en dos o más subproblemas que son añadidos a la lista de subproblemas candidatos, y así sucesivamente.

La estrategia de separación origina un árbol, en el que partiendo del vértice o nudo raiz (asociado al problema original), se ramifica asociando un único nudo a cada subpro--blema candidato, identificando problemas candidatos eliminados con nudos "saturados", y aquéllos que no lo son, con nudos o vértices "vivos"; el árbol así construido se denomina "árbol de Branch and Bound" o de "ramificación".

Para resolver el (MCPP), hemos utilizado una estrategia de separación, imponiendo restricciones sobre las parejas de variables asociadas a cada una de las aristas del conjunto L.

Sea $F(P_k)$ el conjunto de soluciones posibles del problema en el nudo k, la separación se realiza con las variables y_{ij}, y_{ji}, siendo $\{i,j\}\varepsilon L(k)$, en la forma siguiente:

$$F(P_k) \cap [y_{ij} \geq 1] \quad \text{y} \quad F(P_k) \cap [y_{ji} = 0]$$

donde L(k) representa el conjunto de aristas que no han sido utilizadas,en alguna separación existente en el único camino desde el nudo raiz,al nudo k,en el árbol de Branch and Bound.

El nudo k, ha sido separado en dos nudos sucesores por medio de las restricciones $y_{ij} \geq 1$, e $y_{ij} = 0$. Esto sig-

nifica que, en la solución de uno de los dos problemas sucesores, la arista {i,j} debe recorrerse al menos una vez en sentido de i a j, mientras que en el otro, la citada arista no puede recorrerse en ése sentido convirtiéndose, por lo tanto, en el arco (j,i).

La estrategia de separación descrita en el párrafo anterior, está bien definida, puesto que, cualquier solución posible al (MCPP), debe cumplir una y sólo una de las dos restricciones que caracterizan a la separación.

La finitud del algoritmo, está garantizada, pues el número total de aristas en el grafo es finito, y no hay repeticiones en el estudio de un mismo problema candidato, a causa de que se ramifica por variables asociadas a las aristas del conjunto L(k), en cada nudo k. Es fácil observar, que para un grafo mixto original con p aristas, el número máximo de nudos en el árbol de ramificación es $2^{p+1} - 1$, por lo que el crecimiento del número de subproblemas candidatos, crece exponencialmente con el número de aristas.

Definición 6.1

En un nudo k del árbol de ramificación, llamaremos camino fundamental del nudo k (c.f.de k), al único camino que va desde el nudo raiz (nudo 0), al nudo k, en el árbol de Branch and Bound.

Notación 6.1

En un nudo k, cualquiera,del árbol de ramificación, representaremos por:

(P_k) al problema asociado.

L(k) al conjunto de aristas no utilizadas en alguna separación existente, en el c.f.de k.

$A_1(k) = \{(i,j): \exists [y_{ij} \geq 1]$ ó $\exists [y_{ji} = 0]$, en el c.f.de k $\}$

$A_2(k) = \{(j,i): \exists [y_{ij} \geq 1]$, en el c.f.de k $\}$

z*(k) es el valor de la mejor solución encontrada, al analizar el nudo k (solución incumbente).

(x^k, y^k) es la solución óptima de (P_k), cuando éste sea posible.

Proposición 6.1

Si el problema (P_k) es posible, entonces $y^k_{ji} \leq 1$, $\forall (j,i) \in A_2(k)$, sin pérdida de optimalidad.

Demostración

Es obvia, en virtud del Teorema 3.2. ●

La Proposición anterior, nos recuerda el hecho de que una arista no puede recorrerse en ambos sentidos, más de una vez en cada uno de ellos, en cualquier solución óptima del (MCPP), si el coste de ésta es estrictamente positivo, mientras que si su coste asociado es cero, puede suponerse que será recorrida una vez como máximo, sin pérdida de optimalidad. En consecuencia, en cada ramificación del tipo $y_{ij} \geq 1$, se puede suponer que su variable pareja cumple la restricción $0 \leq y_{ji} \leq 1$, sin riesgo de eliminar soluciones óptimas del problema original (P). El problema (P_k), por lo tanto, puede ser formulado de la forma siguiente:

$$(P_k) \quad \text{Min} \left\{ \sum_A c_{ij} x^k_{ij} + \sum_{A_1(k)} c_{ij} y^k_{ij} + \sum_{A_2(k)} c_{ij} y^k_{ij} + \sum_{L(k)} c_{ij} (y^k_{ij} + y^k_{ji}) + \sum_A c_{ij} \right\}$$

sujeto a:

$$\sum_j (1+x^k_{ij}) + \sum_j y^k_{ij} = \sum_j (1+x^k_{ji}) + \sum_j y^k_{ji} \qquad \forall i \qquad (1_k)$$

$$y^k_{ij} + y^k_{ji} \geq 1 \qquad \forall \{i,j\} \in L(k) \qquad (2_k)$$

$$\left.\begin{array}{ll} x^k_{ij}, y^k_{ij}, y^k_{ji} \geq 0 & \forall (i,j) \in A, \ \forall \{i,j\} \in L(k) \\ 1 \leq y^k_{ij} & \forall (i,j) \in A_1(k) \\ 0 \leq y^k_{ij} \leq 1 & \forall (i,j) \in A_2(k) \end{array}\right\} \qquad (3_k)$$

$$x^k_{ij}, y^k_{ij}, y^k_{ji} \text{ enteras} \qquad \forall (i,j) \in A \quad \forall \{i,j\} \in L \qquad (4_k)$$

6.2 ESTRATEGIA DE RAMIFICACION.

Durante el desarrollo del árbol de ramificación, se van eliminando los subproblemas candidatos (P_k), de acuerdo-con criterios que se comentarán en la Subsección siguiente.Cuando la lista de subproblemas es vacía, el proceso de Branch and Bound finaliza, proporcionándonos una solución óptima al (MCPP) asociada a alguno(s) de los problemas examinados.

La estrategia de ramificación es la regla utilizada para la selección de los problemas candidatos. En nuestro problema, la estrategia de ramificación debe responder a dos - preguntas, en cualquier nudo no saturado:

1.- ¿Por qué variable y_{ij} debo ramificar?.
2.- ¿Cuál de las dos posibles ramificaciones se elige en primer lugar?.

por otra parte, en cualquier vértice saturado, debe indicarnos cuál es el siguiente vértice vivo a explorar.

Aunque cada algoritmo de Branch and Bound debe utilizar una estrategia de ramificación específica, que le permita explotar al máximo las propiedades del problema a resolver, existen estrategias de ramificación, de tipo más general, aplicables a cualquier algoritmo de Branch and Bound, entre -- las que destacan la estrategia LLB (Least Lower Bound) y la LIFO (Last In First Out).

La estrategia LLB, se caracteriza por examinar-- el vértice vivo correspondiente a la menor cota inferior hallada hasta el momento, intentando encontrar el camino a "buenas" soluciones posibles (se supone que las cotas inferiores más -- bajas, se sitúan en nudos pertenecientes a caminos que conducen a soluciones casi óptimas). Esta estrategia, tiene el inconveniente de que, a menudo, examina consecutivamente nudos-- pertenecientes a caminos fundamentales con pocas separaciones-en común, lo que representa un problema de almacenamiento de - datos en los vértices vivos, con el fin de utilizar la información de sus nudos predecesores.

La estrategia LIFO investiga el nudo sucesor del que se está examinando, si éste no se satura. Esta estrategia produce, con frecuencia, un gran número de soluciones posibles de coste superior a la incumbente, pero supone un ahorro de memoria considerable respecto de la estrategia LLB.

La estrategia utilizada en nuestro trabajo para resolver el (MCPP) ha sido la LIFO, junto con la estrategia de selección de la variable a ramificar, que será explicada más adelante.

Proposición 6.2

Si en el c.f.de k existen p separaciones, entonces encontramos la solución óptima de (P_k), y por lo tanto el nudo k se satura.

Demostración

En efecto, en la situación de la hipótesis tenemos que el conjunto L(k) es vacío, por lo que las restricciones (2_k) del problema (P_k) no existen. En consecuencia, (P_k) es un problema de flujo de coste mínimo con capacidades enteras sobre algunas de sus variables y_{ij} (aquellas tales que $(i,j)\varepsilon A_2(k)$). Las restricciones (1_k), son equivalentes a las ecuaciones de Kirchoff o de conservación de flujo (ver Introducción), sobre el grafo G_k^f formado por todos los arcos originales de G, junto con los arcos $(i,j)\varepsilon A_1(k) \cup A_2(k)$. La resolución de este problema de flujo nos proporciona la solución de (P_k). ●

El problema de flujos resuelto para la obtención de la solución de (P_k) en la Proposición anterior, puede no ser posible, a causa de la no existencia de algunas de las dos direcciones posibles de cada arista del grafo original G, o bien porque la oferta total no puede ser transportada hacia los sumideros, a causa de las restricciones de capacidad sobre los arcos de $A_2(k)$. En cualquier caso, se determina si el problema (P_k) es posible o no, al resolver éste problema de flujos, y en caso afirmativo, los flujos circulados por los arcos de G_k^f proporcionan la solución óptima de (P_k) que además es posible para el (MCPP).

Comentario 6.1

La estrategia LIFO permite examinar rápidamente, los nudos de los niveles más bajos del árbol de ramificación, y por lo tanto que se obtengan un buen número de soluciones posibles del (MCPP), una por cada nudo k tal que $L(k)=\phi$.

La Figura 6.1 representa parte de la rama izquierda de nuestro árbol de ramificación, utilizando la estrategia LIFO, sobre un problema con p aristas. De los dos sucesores posibles de cada vértice vivo, se estudia en primer lugar el correspondiente a la separación $y_{ij} \geq 1$, prosiguiendo la ramificación hasta obtener la saturación de un nudo (subrayado en la Figura 6.1), donde la operación de "vuelta atrás" o "backtracking" ramifica por el nudo del nivel más inferior de la rama que está siendo explorada. Queremos hacer hincapié en la importancia de la aplicación de las Proposiciones 3.3 a 3.9 , en cada backtracking, pues la conversión de una arista {i,j} en el arco (j,i) puede producir un problema imposible, si la arista {i,j} fuera de corte en sentido de i a j, teniendo en cuenta las conversiones de aristas en arcos en las ramificaciones anteriores del camino fundamental correspondiente, o bien, si es posible se puede reducir el número de aristas candidatas a ser ramificadas posteriormente, con la consiguiente reducción del conjunto de nudos a explorar.

Los nudos p y p+1, de la Figura 6.1, han sido saturados al encontrar las soluciones óptimas de los problemas asociados (con $L(p)=L(p+1)=\phi$), mientras que el nudo p+2 ha sido saturado por otras razones que se explicarán en la Subsección 6.3, y se subraya con dos rayas.

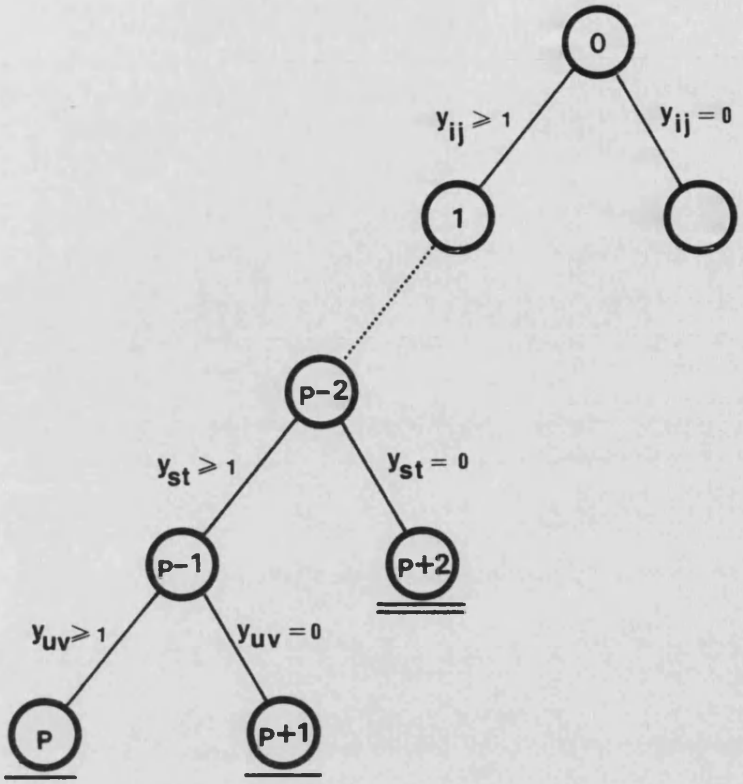


Figura 6.1: Estrategia LIFO.

6.2.1 Selección de la variable de ramificación.

La estrategia de separación en cualquier nudo k, ya hemos visto que se realiza sobre una cualquiera de las aristas del conjunto L(k), y como muestra la Figura 6.1 se analiza primero el subproblema correspondiente al incorporar la restricción $y_{ij} \geq 1$, al conjunto de soluciones posibles del problema P_k.

La elección del sentido de la arista que debe ser examinado en primer lugar, se ha realizado en función de la proximidad entre las direcciones fijadas por la cota superior y la solución óptima del problema. De los 35 problemas test analizados, en 34 de ellos las direcciones de las aristas fijadas en un solo sentido por la cota superior, han sido recorridas al menos en ese sentido también por la solución óptima, mientras que el único problema que no respetaba esta regla (problema 6) era por únicamente una sola de sus aristas. Con objeto de hallar la solución óptima lo más rápidamente posible, se da preferencia a este primer conjunto de aristas y a continuación se ramifica

por las aristas que aparecieron en ambas direcciones en la cota superior (i.e: aquéllas por las que circuló un flujo de una unidad por su arco artificial correspondiente), en este segundo -- grupo de aristas se les dá prioridad en la ramificación a las - que fueron simplificadas por la mejora de la cota superior co-- mentada en la Sección 4. El éxito de esta regla, se refleja en la prontitud en que se encuentra el óptimo del (MCPP) en casi - todos los problemas testados con relación al número de nudos -- examinados (ver Tabla 6.2).

Finalmente, esta estrategia ha sido mejorada con la ordenación de las aristas en orden decreciente respecto a su coste en cada uno de los grupos de aristas.

6.3 ELIMINACION DE SUBPROBLEMAS

La eliminación de subproblemas candidatos en nuestro algoritmo de Branch and Bound, se realiza por una de las tres condiciones siguientes:

a) Al resolver el subproblema asociado a un nudo k, se obtiene una solución óptima a dicho subproblema, y por lo tanto una solución posible al (MCPP).

b) El subproblema (P_k) es imposible.

c) La cota inferior obtenida en el nudo k supera o iguala al valor de la solución incumbente z*(k).

El caso a ocurre cuando al utilizar la estrategia de ramificación LIFO junto con la de elección de la variable a ramificar, se verifican las condiciones de la Proposición 6.2, o bien si al resolver el problema relajado de (P_k) se satisface la condición (c) del Teorema 1.6, que no es más que la traslación del Teorema de la Holgura Complementaria de la Teoría de la Dualidad adaptada a la Relajación Lagrangiana.

El caso b se produce cuando el problema de flujo definido por las restricciones (1_k) y (3_k) del problema (P_k) es imposible, y esto sucede cuando en el nudo anterior en el camino fundamental del nudo k se ramifica por la separación $y_{ij} = 0$, siendo la arista {i,j} de corte en sentido de i a j respecto del grafo G_k cuyos arcos son los de A, junto con las aristas de L(k), y los arcos $(i,j) \in A_1(k)$ e $(i,j) \in A_2(k)$, estos últimos con capacidad máxima de una unidad, o bien cuando se ramifica por la separación $y_{ij} \geq 1$ (en este caso no se producen nuevas aristas de corte), y el flujo óptimo necesita que circulen como mínimo dos unidades de flujo por el arco (j,i), lo que equivale a la no optimalidad del sentido $i \rightarrow j$ de la arista {i,j} en el problema (P_k) y sus descendientes.

El caso c hace referencia a la importancia de obtener los valores más altos para la cota inferior obtenida en

cada nudo k, de forma que se obtengan valores cercanos al valor óptimo del problema (P_k), con el mínimo esfuerzo computacional posible.

La cota inferior más apropiada para el proceso de saturación de nudos es, sin duda, la obtenida por la relajación - lagrangiana de las restricciones de obligatoriedad, por los motivos siguientes:

1º. Proporciona buenas cotas inferiores, sobre todo cuando se están analizando nudos en los niveles más bajos -- del árbol de ramificación.

2º. Facilita los criterios de imposibilidad de (P_k), puesto que para obtenerla se realiza al menos un flujo sujeto al conjunto de restricciones (1_k) y (3_k).

3º. La separaciones del tipo $y_{ij} \geq 1$ no varían la paridad de los vértices del grafo asociado al nudo predecesor del k, cambiando ésta únicamente, en el caso en que a raiz de una separación del tipo $y_{ij} = 0$, se produzcan nuevas aristas de corte y/o duplicaciones de ciertos arcos, tal como fue discutido en la Sección 2, por lo que la relajación lagrangiana de las restricciones de simetría es poco relevante en la mayor parte de los - nudos explorados.

4º. La restricciones de capacidad generadas por las separaciones $y_{ij} \geq 1$, no pueden ser integradas en el con-- junto de restricciones del problema relajado PR(x,y',u), con el fin de mejorar su valor óptimo.

Unicamente hemos utilizado la relajación (RL_1), - en la obtención de la cota inferior en el nudo raiz, puesto que en este caso es cuando hay más aristas en el grafo asociado que en cualquier nudo interno del árbol, pues la asignación de -- direcciones a las aristas existentes mediante las separaciones utilizadas convierte a éstas en arcos, transformando paulatina-- mente el grafo mixto G original en una sucesión de grafos --

derivados de él,que en algunos casos finaliza en un grafo totalmente dirigido para el que se encuentra la solución óptima, según se indicó en la Proposición 6.2.

Las Proposiciones siguientes permiten el ahorro de cálculo de la cota inferior en algunos nudos del árbol de ramificación.

NOTACION 6.2

En un nudo k cualquiera del árbol de Branch and-Bound, representaremos por:

$PR(x^k,y^k,\lambda)$ al problema relajado lagrangianamente de (P_k) respecto de las restricciones (2_k).

$\mathcal{L}_2(x^k,y^k,\lambda)$ a la función objetivo de $PR(x^k,y^k,\lambda)$, con λ dado.

$LBFLU1(k) = v(PR(x^k,y^k,0))$.

$(\bar{x}^k,\bar{y}^k)$ la solución óptima de $PR(x^k,y^k,0)$.

$$L_0(k) = \{ \{i,j\} \in L(k) : y^k_{ij} + y^k_{ji} = 0 \}$$

$$L_1(k) = \{ \{i,j\} \in L(k) : y^k_{ij} + y^k_{ji} = 1 \}$$

$$L_2(k) = \{ \{i,j\} \in L(k) : y^k_{ij} + y^k_{ji} \geq 2 \}$$

Nótese que los conjuntos $L_0(k)$, $L_1(k)$ y $L_2(k)$ forman una partición del conjunto $L(k)$ en cualquier nudo.

Proposición 6.3

Si en un nudo k se ramifica por la arista $\{s,t\}$ perteneciente al conjunto $L_1(k) \cup L_2(k)$, entonces en al menos uno de sus dos nudos sucesores la cota LBFLU1 no aumenta.

Demostración

En efecto, sean k_1 y k_2 los dos nudos sucesores del nudo k, correspondientes a las separaciones $y_{st} \geq 1$, e $y_{ts}=0$, respectivamente. Por hipótesis sabemos que: $\bar{y}^k_{st} + \bar{y}^k_{ts} \geq 1$, pudiendo ocurrir:

a) $\bar{y}^k_{st} \geq 1$, en cuyo caso, el flujo óptimo $(\bar{x}^k,\bar{y}^k)$ obtenido en el nudo k, es también óptimo en el nudo k_1, por lo que $LBFLU1(k_1) = LBFLU1(k)$.

b) $\bar{y}^k_{st} = 0$, en cuyo caso, el flujo calculado en el nudo k,

es también óptimo en el nudo k_2, por lo que $LBFLU1(k_2) = LBFLU1(k)$. ●

Proposición 6.4

Sea k_1 el nudo sucesor del nudo k, correspondiente a la separación $y_{st} \geq 1$, entonces si $\{s,t\} \in L_0(k)$, se tiene que $LBFLU1(k_1) \leq LBFLU1(k) + 2c_{st}$.

Demostración

Obvia si consideramos que (x^{k1}, y^{k1}) definida por:

$$x_{ij}^{k1} = \bar{x}_{ij}^{k} \qquad \forall (i,j) \in A$$

$$y_{ij}^{k1} = \bar{y}_{ij}^{k} \qquad \forall (i,j) \in A_1(k) \cup A_2(k)$$

$$\left.\begin{array}{l} y_{ij}^{k1} = \bar{y}_{ij}^{k} \\ y_{ji}^{k1} = \bar{y}_{ji}^{k} \end{array}\right\} \qquad \forall \{i,j\} \in L(k) - \{s,t\}$$

$$y_{st}^{k1} = \bar{y}_{ts}^{k1} = 1$$

es una solución posible del problema $PR(x^k, y^k, 0)$, cuyo coste es $LBFLU1(k) + 2c_{st}$. ●

Proposición 6.5

Sea $(\bar{x}^k, \bar{y}^k)$ la solución óptima de $PR(x^k, y^k, 0)$. Sea λ un vector de multiplicadores lagrangianos cualquiera. Entonces:

$$v(PR(x^k, y^k, \lambda)) \leq LBFLU1(k) + \sum_{L_0(k)} \lambda_{ij} + \sum_{L_2(k)} \lambda_{ij}(1 - \bar{y}_{ij}^{k} - \bar{y}_{ji}^{k})$$

Demostración

Puesto que $(\bar{x}^k, \bar{y}^k)$ es una solución posible del problema $PR(x^k, y^k, \lambda)$, tenemos que:

$$v(PR(x^k, y^k, \lambda)) \leq \mathcal{L}_2(\bar{x}^k, \bar{y}^k, \lambda)$$

Además sabemos que:

$$\mathcal{L}_2(\bar{x}^k,\bar{y}^k,\lambda) = \sum_A c_{ij}\bar{x}^k_{ij} + \sum_{A_1(k)} c_{ij}\bar{y}^k_{ij} + \sum_{A_2(k)} c_{ij}\bar{y}^k_{ij} +$$

$$+ \sum_{L(k)} c_{ij}(\bar{y}^k_{ij} + \bar{y}^k_{ji}) + \sum_{L(k)} \lambda_{ij}(1-\bar{y}^k_{ij}-\bar{y}^k_{ji}) + \sum_A c_{ij} =$$

(por la definición de LBFLU1(k))

$$= \text{LBFLU1}(k) + \sum_{L(k)} \lambda_{ij}(1-\bar{y}^k_{ij}-\bar{y}^k_{ji}) =$$

(por definición de $L_0(k)$, $L_1(k)$ y $L_2(k)$)

$$= \text{LBFLU1}(k) + \sum_{L_0(k)} \lambda_{ij} + \sum_{L_2(k)} \lambda_{ij}(1-\bar{y}^k_{ij}-\bar{y}^k_{ji}). \bullet$$

Corolario 6.5.1

El valor óptimo del problema relajado $PR(x^k,y^k,\lambda)$ con un λ cualquiera, no es mayor que el valor de la cota LBFLU1(k) más el coste de las aristas de $L_0(k)$.

Demostración

Es una consecuencia de que $0 \leqslant \lambda_{ij} \leqslant c_{ij}$ $\forall\{i,j\}\in L(k)$ según vimos en la Proposición 5.7, y además considerando que los productos $\lambda_{ij}(1-\bar{y}^k_{ij}-\bar{y}^k_{ji}) \leqslant 0$ $\forall\{i,j\}\in L_2(k)$. ●

Comentario 6.2

El Corolario 6.5.1 nos proporciona un criterio para no calcular la cota inferior en un nudo k del árbol de Branch and Bound, si ésta no puede alcanzar el valor de la solución incumbente z*(k), esto es si:

$$\text{LBFLU1}(k) + \sum_{L_0(k)} c_{ij} < z^*(k)$$

no se calculará la cota inferior en el nudo k, en cuyo caso se pasa al estudio de los nudos sucesores k_1 y k_2, donde conocida la solución óptima del problema $PR(x^k,y^k,0)$ se puede obtener el valor de LBFLU1(k_1) y de LBFLU1(k_2), en el caso de que se ramifique por alguna de las aristas del conjunto $L_1(k) \cup L_2(k)$, sin realizar el flujo correspondiente, según se ha visto en la Proposición 6.3; por otra parte, si la ramificación es por una de las aristas del conjunto $L_0(k)$, se puede calcular el incremento máximo del valor de la cota LBFLU1(k_1), que si está alejado del va-

lor de z*(k), nos dá un criterio para no persistir en el intento de saturar el nudo k_1 utilizando el método del subgradiente, continuando la ramificación. Por último si LBFLU1(k) + $\sum_{L_0(k)} c_{ij}$ si supera al valor de la solución incumbente, la Proposición 6.5 nos facilita un criterio para seleccionar vectores λ's de forma que:

$$\sum_{L_0(k)} \lambda_{ij} + \sum_{L_2(k)} \lambda_{ij}(1-\bar{y}^k_{ij}-\bar{y}^k_{ji}) \geq z^*(k) - \text{LBFLU1}(k) \qquad (6.1)$$

donde no hay que olvidar que los multiplicadores

$$\lambda^1_{ij} = c_{ij} \qquad \forall\{i,j\}\in L_0(k)$$

$$\lambda^1_{ij} = [c_{ij}/2] \qquad \forall\{i,j\}\in L_1(k)$$

$$\lambda^1_{ij} = 0 \qquad \forall\{i,j\}\in L_2(k)$$

que dan origen a la cota LBFLU2(k) = $v(PR(x^k,y^k,\lambda^1))$, satisfacen en este caso la desigualdad (6.1).

Aplicación del Algoritmo de Branch and Bound a la resolución del (MCPP), en un problema concreto. Ejemplo.

Con esta última etapa, concluimos el estudio y resolución del problema número 12, que hemos venido utilizando como ejemplo durante toda la memoria. El grafo original de este problema se encuentra representado en la Figura 4.1, y el árbol de ramificación completo se halla en la Figura 6.2. A la derecha de cada nudo, aparece el valor de la cota inferior correspondiente al mismo, cuando ésta ha sido calculada. El par ordenado de números entre paréntesis que aparece junto a (o encima) cada rama del árbol, indica la separación que se ha realizado. Así, por ejemplo, el par (7,9) indica que la arista {7,9} debe recorrerse al menos una vez en sentido $7 \to 9$, en la solución del (MCPP), mientras que $(\overline{7,9})$, impide que dicha arista se recorra en dicho sentido, y por lo tanto se transformará,en los nudos-sucesores, en el arco (9,7). Los nudos saturados, aparecen subrayados una o dos veces, dependiendo de si se ha encontrado la solución óptima del problema (P_k) asociado, que por supuesto es posible de (P), cuyo valor se encuentra debajo del nudo, o si se ha encontrado que la cota inferior supera o iguala al valor de la solución incumbente. Las estrellas, asociadas a algunas de las cotas inferiores, indican que han sido saturadas por la cota LBFLU2, el resto han sido saturados por LBFLU1, no habiéndose recurrido a la optimización subgradiente en ninguno de los nudos internos del árbol.

Para obtener el orden de la ramificación, recordemos que en cada nudo k se puede ramificar por cualquiera de las aristas de L(k), y esto es aplicable también al nudo 0, en el que se calcula la cota superior mejorada, y donde se determinan las aristas y arcos de corte con sus repeticiones, quedando únicamente las aristas del conjunto

$$L(0) = \{\{3,7\},\{13,10\},\{11,10\},\{9,14\},\{11,6\},\{8,1\},\{4,8\},\{3,4\},\{7,9\},\{5,7\},\{3,6\},\{9,8\},\{15,13\}\}$$

que ya hemos ordenado de acuerdo con la estrategia de ramifica

ción, ésto es,las ocho primeras aristas de L(0) se recorrían únicamente en uno de sus dos sentidos en la cota superior inicial (ver Figura 4.5), y las cinco aristas siguientes que se recorrían en ambos sentidos en la cota superior inicial, fueron simplificadas para mejorar la cota superior, recorriéndose en uno solo de sus sentidos en la cota superior mejorada (ver Figura 4.6), por último la única arista que se recorría en ambos sentidos en la cota superior mejorada, es la {15,13} que forma el último grupo de aristas. En cada uno de los grupos, las aristas se ordenan de acuerdo con su coste asociado, en forma decreciente (esto obliga a incorporar las aristas de mayor coste,lo más rápidamente posible,en los problemas de flujo que se resuelven en cada nudo). En el nudo 14 se llega a la primera "cola" del árbol (i.e: $L(14) = \phi$), y el flujo de mínimo coste calculado en él proporciona una solución posible (óptima de (P_{14})), cuyo coste es de 199, no mejorando por lo tanto la mejor solución encontrada en el nudo raiz,que también tuvo ése valor. La estrategia LIFO nos lleva al examen del vértice vivo 13 correspondiente al nivel más inferior (último de los que han entrado en el árbol), ramificando por $\overline{(15,13)}$, y proporcionando otra solución posible que esta vez tiene un coste de 217. La ramificación prosigue, finalizando en el nudo 34 donde se encuentra una solución posible de coste 190 que iguala a la cota inferior obtenida en el nudo raiz, y por lo tanto es óptima. Podemos ahora obtener dicha solución siguiendo las ramificaciones que conducen desde el nudo 0 al nudo 34 (i.e: por el c.f.de 34), sustituyendo cada arista del grafo transformado en el nudo 0 (ver Figura 4.2), -- por un arco dirigido en el sentido de la ramificación y añadiendo las copias de los arcos que indique el flujo óptimo circulado en la resolución del problema (P_{34}) dado por:

$f^*(6,3) = 1$, $f^*(8,9) = 1$, $f^*(10,15) = 1$, $f^*(11,6) = 1$,

$f^*(11,10) = 1$.

obteniendo, la solución óptima al (MCPP), representada por el grafo euleriano asignado de G de la Figura 6.3.

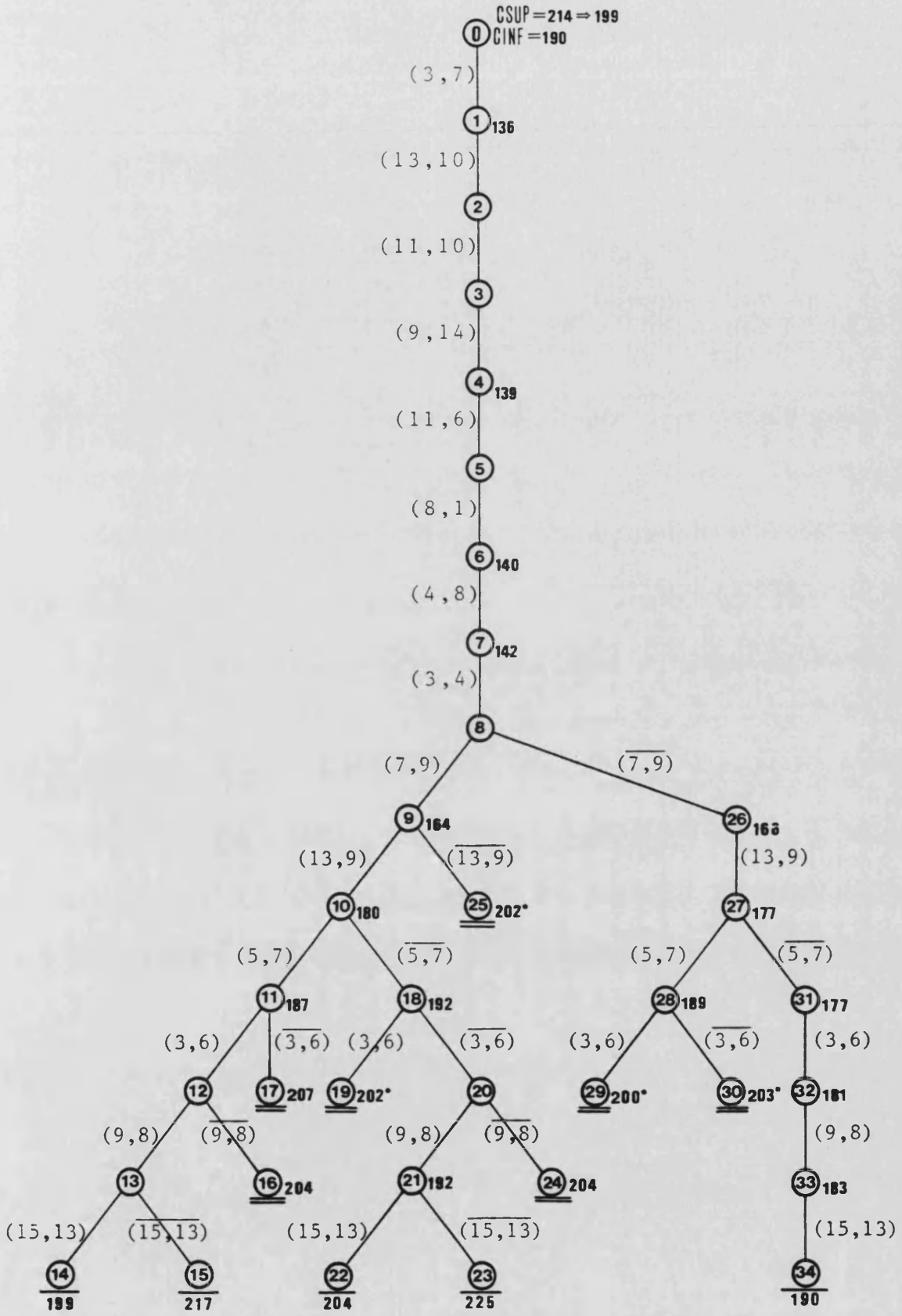


Figura 6.2: Arbol de Branch and Bound del problema 12.

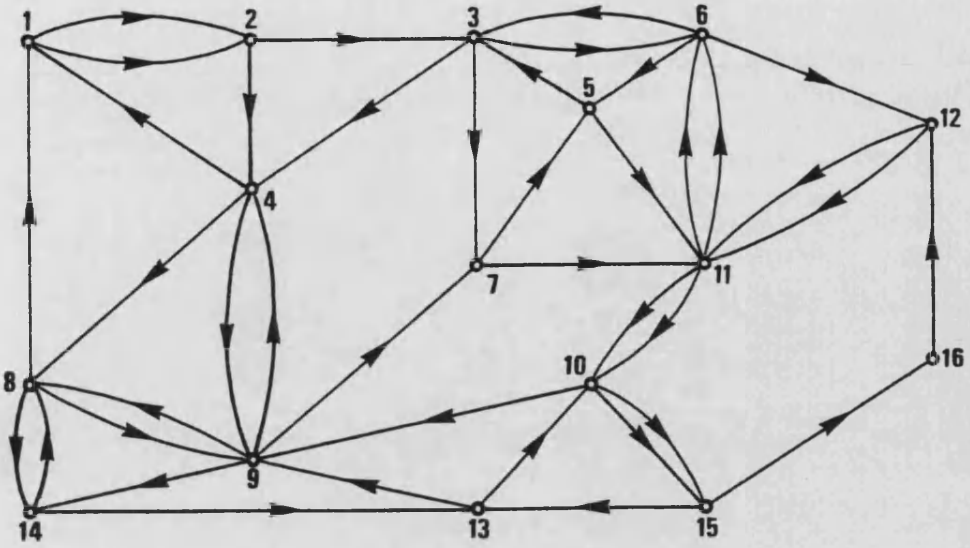


Figura 6.3: La solución óptima al (MCPP) del problema 12.

6.4 RESULTADOS COMPUTACIONALES.

Durante el desarrollo de las Secciones precedentes, ya han sido presentados algunos resultados computacionales parciales referentes a la colección de problemas testados, referentes a la cota superior, relajaciones, método del subgradiente,etc....En la tabla 6.1 ofrecemos las características más importantes de los 35 grafos mixtos, fuertemente conexos, e impares, generados aleatoriamente, con costes asociados a sus arcos y aristas, obtenidos al azar entre los valores 1 y 25. Las columnas 2,3,4,5, nos dan el número de vértices totales, número de vértices impares, número de arcos y número de aristas en el grafo original, mientras que las columnas 6 y 7 nos dan respectivamente el número de arcos y de aristas en el grafo transformado en el nudo 0, por la aplicación de las Proposiciones 3.3 a 3.9 en el grafo original (Etapa 1 de la Cota Superior), la columna 7,por lo tanto, nos dá la profundidad máxima de las ramas del árbol de Branch and Bound, para cada uno de los problemas.

Para la utilización del método del subgradiente en el algoritmo de Branch and Bound, se han probado dos niveles de utilización, en el primer nivel, se acudía a la optimización subgradiente cuando un nudo k no quedaba saturado ni por LBFLU1(k), ni por LBFLU2(k), pero se tenía que:

$$\max (LBFLU1(k),LBFLU2(k))\varepsilon[0.95z^*(k),z^*(k)) \qquad (6.2)$$

esto es, si la cota inferior mejor de entre LBFLU1(k) y LBFLU2 (k), quedaba a menos de 5 centésimas del valor de la mejor solución encontrada hasta ése momento, con objeto de que la optimización subgradiente saturara gran parte de estos nudos.

En la Tabla 6.2, figura el número de nudos examinados por el algoritmo de Branch and Bound (columna 2), el número de nudos saturados por la cota inferior (excluídos los nudos imposibles), en la columna 4, donde puede apreciarse que este número es aproximadamente la mitad de los nudos explorados, en las columnas 5 y 6 se dan, respectivamente, el número de ve-

ces que se acude a la optimización subgradiente y de éstas el número de veces que el nudo en cuestión quedaba saturado. Con esta utilización de la optimización subgradiente, se pudieron resolver resolver en un tiempo máximo de 500 segundos de C.P.U en un computador UNIVAC 1100/60, todos los problemas, excepto el 29,30,34 y 35. Los tiempos obtenidos para los 31 problemas restantes se hallan en la columna 7. Puesto que la optimización subgradiente requería un máximo de 20 iteraciones y cada una de ellas requiere la resolución de un problema de flujo de mínimo coste, se probó un segundo nivel de utilización del método del subgradiente sobre aquellos nudos que no eran saturados ni por LBFLU1(k),ni por LBFLU2(k), acudiendo a él únicamente en los nudos , donde el máximo de estas cotas inferiores quedaba a tan sólo 5 milésimas de la solución incumbente, esto es si:

$$\max (LBFLU1(k),LBFLU2(k)) \in [0.995z^*(k),z^*(k)) \qquad (6.3)$$

Con este segundo nivel, cabía esperar un aumento en el número de nudos examinados, pero una disminución en el número de llamadas a la optimización subgradiente, así como un aumento relativo del número de nudos saturados por dicha optimización. Los resultados obtenidos con este segundo nivel se encuentran en la Tabla 6.3, sobre los problemas que acudían al subgradiente más de una vez (ésta corresponde al nudo 0), y por lo tanto sobre los únicos que cabía esperar diferencias. Con este segundo nivel se aprecia un claro aumento del número total de nudos explorados, que en total supone un 35% más con el segundo nivel que con el primero, pero con un tiempo de computación de aproximadamente el 30% menor a favor del segundo nivel. Además, se pudo resolver el problema 35 con esta segunda estrategia, dentro del tiempo límite (este problema se ha excluído del cálculo de los porcentajes anteriores). El método del subgradiente no se muestra efectivo si se utiliza sistemáticamente como herramienta de saturación, a menos que se disminuya su uso a niveles casi nulos (2do. nivel), y desgraciadamente no existen métodos que permitan regular esta utilización "a priori", pues el comportamiento del método del subgradiente es diferente en

cada problema, en particular, como ya se pudo observar al final de la Sección 5.

1	2	3	4	5	6	7
1	8	4	7	4	11	0
2	9	2	7	5	12	2
3	7	2	3	6	10	0
4	13	8	14	7	23	3
5	13	6	13	8	20	7
6	18	12	16	16	18	15
7	10	4	8	6	12	4
8	10	10	7	10	9	9
9	19	8	15	15	28	10
10	8	6	5	10	5	10
11	13	8	12	13	14	11
12	16	8	16	17	21	14
13	16	6	16	17	21	14
14	27	16	43	20	44	20
15	43	28	74	20	81	20
16	17	10	19	20	20	20
17	25	16	25	20	35	17
18	25	16	25	20	35	17
19	15	8	9	20	19	16
20	10	6	9	10	9	10
21	38	20	61	27	68	27
22	30	14	46	23	49	22
23	33	16	48	22	58	19
24	28	18	47	25	47	25
25	32	16	55	26	62	23
26	30	14	47	28	50	28
27	41	26	52	29	68	25
28	48	16	78	32	94	28
29	38	22	52	35	54	34
30	44	20	62	38	73	35
31	15	8	10	21	10	21
32	21	6	12	26	13	25
33	20	10	15	24	26	20
34	50	24	77	40	90	38
35	50	24	66	39	88	35

Tabla 6.1: Características más importantes de los grafos test.

1. Número de problema.
2. Número de vértices.
3. Número de vértices impares.
4. Número de arcos en el grafo original.
5. Número de aristas en el grafo original.
6. Número de arcos en el grafo transformado del nudo raiz.
7. Número de aristas en el grafo transformado del nudo raiz.

1	2	3	4	5	6	7
1	1	0	1	0	0	0.03
2	7	0	0	1	0	0.09
3	1	0	1	0	0	0.03
4	7	0	2	1	0	0.16
5	1	0	1	1	0	0.14
6	101	54	47	18	2	5.8
7	1	0	1	1	0	0.07
8	63	0	23	1	0	1.1
9	45	0	21	1	0	1.3
10	21	21	2	1	0	0.4
11	1	0	1	1	0	0.7
12	34	34	7	1	0	1.8
13	1	0	1	1	0	1.4
14	509	31	231	96	19	76.7
15	255	50	122	16	9	53.5
16	175	37	80	38	29	11.9
17	55	17	24	5	3	5.6
18	117	22	47	7	4	9.7
19	16	16	0	1	0	0.9
20	29	17	11	2	1	0.6
21	299	48	146	56	12	118.0
22	349	25	171	61	9	46.5
23	183	0	88	36	23	33.0
24	1543	26	754	454	128	498.1
25	1119	23	552	184	74	283.3
26	595	93	288	101	23	114.0
27	295	25	142	80	25	133.0
28	331	57	158	111	12	226.5
31	23	23	0	1	0	0.6
32	93	93	29	6	0	5.7
33	143	39	66	27	16	10.8

Tabla 6.2: Resultados computacionales con entrada al subgradiente al nivel 0.95.

1. Número del problema.
2. Total de nudos estudiados.
3. Nudo en el que se encuentra el óptimo.
4. Total de nudos saturados.
5. Número de veces que se acude al subgradiente.
6. Número de nudos saturados por el subgradiente.
7. Tiempo de C.P.U en segundos.

1	2	3	4	5	6	7
6	107	55	50	1	0	3.8
14	589	31	271	19	8	53.4
15	311	50	150	13	8	59.5
16	345	37	165	12	11	14.7
17	77	17	35	1	0	6.5
18	135	22	56	2	1	9.4
20	33	17	13	1	0	0.7
21	401	48	197	16	8	83.3
22	355	25	174	20	9	84.7
23	321	0	157	10	9	35.3
24	2437	26	1201	97	67	288.0
25	1543	23	764	71	65	210.0
26	663	93	322	30	18	74.0
27	533	25	261	25	18	98.1
28	345	57	165	31	11	107.4
32	94	94	29	5	0	5.7
33	227	39	108	4	3	12.0
35	1499	0	731	56	38	498.6

Tabla 6.3: Resultados computacionales con entrada al subgradiente al nivel 0.995 (no figuran los problemas que no ofrecen variación respecto del nivel 0.95).

1. Número del problema.
2. Total de nudos estudiados.
3. Nudo en el que se encuentra el óptimo.
4. Total de nudos saturados.
5. Número de veces que se acude al subgradiente.
6. Número de nudos saturados por el subgradiente.
7. Tiempo de C.P.U en segundos.

6.3 COMENTARIOS Y VALORACION

El problema del Cartero Mixto (MCPP) es un problema de Optimización Combinatorial, que, como se ha señalado, pertenece a la clase NP - (problemas no acotados polinomialmente en tiempo de resolución). Este hecho nos dá una idea de la dificultad que entraña su resolución, puesto que, salvo que las clases NP y P sean, en realidad, una sola, no existe ningún algoritmo polinomial que pueda resolver el (MCPP).

Consideremos la importancia de los algoritmos acotados polinomialmente. Una función polinómica crece con mucha menor rapidez que una - función exponencial, y una función exponencial crece mucho menos rápidamente que una función factorial. Para ilustrar esta aserción, consideremos el siguiente ejemplo de Lawler (1976):

"Supongamos que un algoritmo que resuelva el problema del arc-covering requiera $100n^3$ operaciones, y otro requiera 2^n operaciones, - donde n es el número de vértices a recubrir. El algoritmo exponencial es - más eficiente para grafos con no más de 17 vértices. Para grafos más grandes, sin embargo, el algoritmo polinomial tiene un crecimiento mejor, con una relación de crecimiento exponencial en tiempo de resolución respecto al otro. Un problema con 50 vértices, puede ser computacionalmente tratable -- por medio del algoritmo polinomial, pero es probablemente intratable por el algoritmo exponencial."

Para resolver el (MCPP) hemos construído un algoritmo cuyo tiempo de resolución crece exponencialmente con el número de aristas del - grafo, sin embargo, las ventajas que éste supone respecto a un algoritmo de enumeración son indiscutibles. Pensemos en que un grafo mixto con p aristas admite 2^p orientaciones distintas de ellas, y que para encontrar la solución hay que resolver 2^p flujos de mínimo coste. Suponiendo que un ordenador potente resolviera un problema de flujos en 0.001 segundos (esta cifra también depende del número de arcos en el grafo), la solución al (MCPP) supondría 1048.6 segundos para un problema con 20 aristas, 9.3 horas para un problema con 25 aristas y mas de un año y un mes, para resolver problemas con 35 aristas. Nuestro algoritmo ha resuelto problemas de hasta 39 aristas con un tiempo inferior a los 500 segundos con un ordenador UNIVAC 1100/60 que es más lento que el supuesto del párrafo anterior.

Por otra parte, el tiempo de resolución puede disminuir os-

tensiblemente, si nos restringimos a soluciones ε-óptimas del (MCPP), a diferencia de los algoritmos heurísticos que existían, que tan sólo garantizaban soluciones con coste no superior a 5/3 veces el valor de la solución óptima.

La utilización del (MCPP) para resolver problemas de la vida cotidiana, como son el de recogida de basura, limpieza de calles, etc.. en las grandes ciudades, hacen de él un problema de interés no sólo científico, sino también social, debido al gran coste público que suponen estas tareas.

En la práctica computacional, se ha demostrado que la Relajación Lagrangiana es mucho más eficiente que la clásica Relajación Lineal, para obtener cotas inferiores en los nudos del árbol de Branch and Bound, así como que la solución heurística que proporciona la cota superior inicial,resulta más rápida de obtener que la obtenida a partir de la Relajación Lineal e incluso con valores más cercanos al óptimo. Además la optimización subgradiente se muestra mucho más operativa si se utiliza con cuidado inmersa en el algoritmo de Branch and Bound.

Por último, esperamos que este primer algoritmo exacto para resolver el (MCPP), dé lugar a posteriores investigaciones (al igual -- que ocurrió con el TSP), que permitan abordar problemas de mayor tamaño a los utilizados en este trabajo.

REFERENCIAS Y BIBLIOGRAFIA

[1] BALAS,E. (1980)

Cutting planes from conditional bounds: a new approach to Set Covering.

Math. Programming Study 12 (1980) pp. 279-312

[2] BALAS,E. and N.CHRISTOFIDES (1981)

A restricted Lagrangean approach to the Travelling Salesman Problem.

Math. Programming 21 (1981) pp. 19-46

[3] BALINSKI,M.L. Ed. (1974)

Approaches to Integer Programming.

Math. Programming Study 2 (1974)

BAZARAA,M.S. and JARVIS,J.

Linear Programming and Network Flows.

John Wiley and Sons, New York (1977)

[4] BELLMAN,R. and K.L.COOKE (1969)

The Königsberg bridges problem generalized.

Il. of Math.Anal.and Appl. 25 (1969) 1

[5] BELTRAMI,E.J. and L.D.BODIN (1974)

Networks and vehicle routing for municipal waste collection.

Networks 4 (1974) pp. 65-94

BERGE,C. (1962)

The Theory of Graphs and its Applications

John Wiley and Sons, New York (1962)

BONDY,J. and MURTY,U.S.R. (1976)

Graph Theory with Applications.

Mac Millan Press, London (1976)

BUSACKER,R.G. and T.L.SAATY (1965)

Finite Graphs and Networks

MacGraw Hill, New York (1965)

[6] CHRISTOFIDES,N. (1972)

Bounds for the Travelling Salesman Problem.

Operations Research 20 (1972) pp. 1044-1055

[7] CHRISTOFIDES,N. (1973)

The optimal traversal of a graph.

Omega 1 (1973) pp. 719-732

CHRISTOFIDES,N. (1975)

Graph Theory. An Algoritmic Approach.

Academic Press, New York (1975)

[8] CHRISTOFIDES,N. (1976)

Worst case analysis of a new heuristic for the TSP.

Tech. Report, Grad. School Industrial Admin., Carnegie Mellon University.

[9] CHRISTOFIDES,N. (1979)

The Travelling Salesman Problem

Combinatorial Optimization (N.Christofides et al. Eds.)

John Wiley (1979)

[10] CHRISTOFIDES,N.,A.MINGOZZI and P.TOTH (1979)

The Vehicle Routing Problem

Combinatorial Optimization (N.Christofides et al. Eds.)

John Wiley (1979)

[11] CHRISTOFIDES,N.,A.MINGOZZI and P.Toth (1981)

Exact Algorithms for the Vehicle Routing Problem, based on SpanningTree and Shortest Path Relaxations.

Math. Programming 20 (1981) pp. 255-282

[12] CORBERAN,A. (1982)

Circuitos Eulerianos Optimos en Grafos no dirigidos:RPP

Tesis Doctoral, Universidad de Valencia (1982)

[13] CORNUEJOLS,G. and G.L.NEMHAUSSER (1978)

Tight Bounds for Christofides' Travelling Salesman Heuristic.

Math. Programming 14 (1978) pp. 116-121

[14] DEMIANOV,V.F. (1966)

Algorithms for some Minimax Problems.

J.Comp.Syst.Sci. 2 (1966) pp.342-380

[15] DREYFUS,S.E. and LAW,A. (1978)

The Art and Theory of Dinamic Programming.

Academic Press, New York (1978)

[16] EDMONDS,J. (1965)

The Chinese Postman Problem

Operations Research 13 Suppl.1 (1965) p.373

[17] EDMONDS,J. (1965)

Maximum Matching and a Polyedrn with (0,1) Vertices.

Journal.Res.Nat.Bur.standards. Sec.B (1965) pp.125-130

[18] EDMONDS,J. and JOHNSON,E. (1973)

Matching, Euler Tours and the Chinese Postman.

Math. Programming 5 (1973) pp.88-124

[19] ELAM,J.,GLOVER,F. and KLINGMAN,D. (1979)

A Strongly Convergent Primal Simplex Algorithm for generalized Networks.

Mathematical Operations Research. Vol.4 (1979) pp. 39-159

[20] FISHER,M. (1980)

Worst Case Analysis of Heuristic Algorithms.

Management Science 26 (1980) pp. 1-17

[21] FISHER,M. (1981)

Lagrangian Relaxation Methods for solving Integer Programming.

Management Science 27 (1981) pp. 1-18

FORD,L.R. and D.R.FULKERSON (1962)

Flows in Networks.

Princeton University Press (1962)

[22] FREDERICKSON,G.N. (1979)

Aproximation Algorithms for some Postman Problems.

Journal A.C.M. 26 (1979) pp. 538-554

[23] GAREY, M.R. and D.S.JOHNSON (1979)

Computers and Intractability: A guide to the Theory of NP-completeness.

Freeman and Co., San Francisco (1979)

GARFINKEL,R. and G.NEMHAUSER (1972)

Integer Programming.

John Wiley, New York (1972)

[24] GARFINKEL,R. (1979)

Branch and Bound Methods for Integer Programming (in Combinatorial Optimization.)

N.Christofides et al. Eds. John Wiley (1979)

[25] GEOFFRION,A. (1971)

Duality in Non-Linear Programming: A simplified applications-oriented development.

SIAM Review. Vol. 13. nº 1. pp. 1-37

GEOFFRION,A. and R.E.MARSTEN (1972)

Integer Programming Algorithms: A Framework and State-of-the-Art Survey.

Management Science, 18, (1972) pp.465-491

[26] GEOFFRION,A. (1974)

Lagrangean Relaxation for Integer Programming.

Math. Programming Study 2 (1974) pp. 82-114

[27] GLOVER,F.,HULTZ,J.,KLINGMAN,D. and STULTZ,J. (1978)

Generalized Networks: A Fundamental Computer-Based Planning Tool.

Management Science. Vol.24 (1978) pp. 1209-1220.

[28] GOFFIN,J.L. (1977)

On the convergence rates of Subgradient Optimization Methods.

Math. Programming 13 (1977) pp. 329-347

[29] GOLDEN,B. (1976)

Shortest Path Algorithms: a comparison.

Operations Research 24 (1976) pp. 1164-1168

[30] GOLDEN,B.L. and R.T.WONG (1981)

Capacitated Arc Routing Problems.

Networks 11 (1981) pp.305-315

[31] HAMMER,P.L.,E.L.JOHNSON,B.H.KORTE and G.L.NEMHAUSER (1977)

Studies in Integer Programming

North Holland (1977)

[32] HAMMER,P.L., E.L.JOHNSON and B.H.KORTE (1977)

Discrete Optimization I

North Holland (1977)

[33] HAMMER,P.L., E.L.JOHNSON and B.H.KORTE (1977)

Discrete Optimization II

North Holland (1977)

HARARY,F. (1969)

Graph Theory.

Addison-Wesley, Readind,Mass. (1969)

[34] HELD,M. and R.M.KARP (1970)

The Travelling Salesman Problem and Minimum Spaning Trees.

Operations Research 18 (1970) pp. 1138-1182

[35] HELD,M. and R.M.KARP (1971)

The Travelling Salesman Problem and Minimum Spaning Trees:

Part II.

Math.Programming 1 (1971) pp. 6-26

[36] HELD,M., P.WOLFE and H.P.CROWDER (1974)

Validation of Subgradient Optimization.

Math.Programming 6 (1974) pp. 62-88

[37] HU,T.C. (1969)

Integer Programming and Network Flows

Addison Wesley, Reading, Mass. (1969)

[38] JEROSLOW,R. (1979)

The Theory of Cutting Planes in Combinatorial Optimization.

N.Christofides et al. Eds. John Wiley (1979)

[39] JEWELL,W. (1962)
Optimal Flow Through Netwoks with Gains.
Operations Reseach. Vol.10 (1962) pp.476-499

[40] KAPPAUF,Ch.H. and G.J.KOEHLER (1979)
The Mixed Postam Problem.
Discrete Applied Mathematics 1 (1979) pp. 89-103

[41] KARP,R.M. (1975)
On the Computational Complexity of Combinatorial Problems.
Networks 5 (1975) pp.45-68

[42] KLEE,V. (1980)
Combinatorial Optimization: What is the State of the Art.
Math. of Operations Research 5 (1980) pp. 1-26

[43] KWAN,M.K. (1962)
Graphic Programming using even or odd points.
Chinese.Mathematics 1 (1962) p.273

[44] LAND,A.H. and A.DOIG (1960)
An Automatic Method for solving Discrete Programming.
Econometrica 28 (1960) pp.497-520

[45] LAND,A.H. and S.POWELL (1973)
Fortran Codes for Mathematical Programming.
John Wiley and Sons, New York (1973)

[46] LAND,A.H. and S.POWELL (1979)
Computer Codes for problems in Integer Programming,(in Discrete Optimization II)
P.L.Hammer et al. Eds.
North Holland (1979)

LAWLER,E. (1976)

Combinatorial Optimization: Networks and Matroids.

Hold, Rinehart & Winston (1976)

[47] LENSTRA,J.K. and RINOOY KAN,A.H.G. (1976)

On Genral Routing Problems.

Networks 6 (1976) pp. 593-597

[48] LENSTRA,J.K. and RINOOY KAN,A.H.G. (1981)

Complexity of Vehicle Routing and Scheduling Problems.

Networks 11 (1981) pp. 221-227

[49] MALE,J.W. and LIEBMAN,J.C. (1978)

Districting and Routing for Solid Waste Collection.

Journal of Environmental Engineering Div. (1978)

[50] MAURRAS,J. (1972)

Optimization of the Flow Through Networks with Gains.

Math.Programming, Vol.3 (1972) pp.135-144

[51] MILIOTIS,P. (1976)

Integer Programming Approach to the Travelling Salesman Problem.

Math.Programming 10, (1976) pp.367-378

[52] MILIOTIS,P. (1978)

Using Cutting Planes to solve the Symmetric Travelling Salesman Problem.

Math.Programming 15 (1978) pp. 177-188

MINIEKA,E. (1978)

Optimization Algorithms for Networks and Graphs.

Marcel Dekker, New York (1978)

[53] MINIEKA,E. (1979)

The Chinese Postman Problem for Mixed Networks.

Management Science 25 (1979) pp. 643-648

MINOUX,M. et GONDRAN,M.

Graphes et Algorithmes.

Ed. Eyrolles, Paris (1979)

[54] MOTA,E. (1982)

Circuitos Eulerianos Optimos en Grafos Dirigidos: El (DRPP).

Tesis Doctoral (1982),Universidad de Valencia.

MURTY,K.G. (1976)

Linear and Combinatorial Programming.

John Wiley, New York (1976)

[55] ORLOFF,C.S. (1974)

A Fundamental Problem in Vehicle Routing.

Networks 4 (1974) pp.35-64

[56] PADBERG,M.W. and HONG,S. (1980)

On the Symmetric Travelling Salesman Problem: A Computational Study.

Math.Programming Study 12 (1980) pp. 78-107

[57] PAPADIMITRIOU,C. (1976)

On the Complexity of Edge Traversing.

J.Assoc.Comp.Mach. 23 (1976) pp. 544-554

[58] POLJAC,B. (1967)

A General Method for solving Extremum Problems.

Soviet Math. Dokl. 8 (1967) pp. 593-597

[59] SANDI,C. (1979)

Subgradient Optimization (in Combinatorial Optimization).

N.Christofides et al. Eds. John Wiley (1979)

[60] SHAPIRO,J.F. (1971)

Generalized Lagrange Multipliers in Integer Programming.

Operations Research 19 (1971) pp. 1070-1075.

[61] SHAPIRO,J.F. (1979)

A Survey of Lagrangean Techniques for Discrete Optimization.

Annals of Discrte Math. 5 (1979) pp. 113-138

[62] SHAPIRO,J.F. (1979)

Mathematical Programming. Structures and Algorithms.

John Wiley, New York (1979)

[63] STERN,H. and DROR,M. (1979)

Routing Electric Meter Readers.

Comput. Oper. Res. 6 (1979) pp.209-223