

51  
2e)



UNIVERSIDAD NACIONAL AUTONOMA  
DE MEXICO

FACULTAD DE CIENCIAS

PROBLEMAS DE ACOPLAMIENTO  
EN GRAFICAS

**T E S I S**

QUE PARA OBTENER EL TITULO DE

**A C T U A R I O**

P R E S E N T A :

VICTOR RAFAEL PEREZ PEREZ

CD. UNIVERSITARIA, D. F.

1993

TESIS CON  
FALLA DE ORIGEN



## **UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso**

### **DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

# TESIS CON FALLA DE ORIGEN

## INDICE

INTRODUCCION	1
<b>Capítulo I</b>	
CONCEPTOS Y PROBLEMAS DE LA TEORIA DEL ACOPLAMIENTO	5
1.1 Conceptos y terminología básica de Teoría de Gráficas	5
1.2 Problemas que dan origen a la Teoría del acoplamiento	16
<b>Capítulo II</b>	
TEORIA DEL ACOPLAMIENTO EN GRAFICAS BIPARTITAS	22
2.1 Antecedentes	22
2.2 Equivalencias entre problemas de acoplamiento en gráficas bipartitas y problemas de optimización	23
2.3 Algunas propiedades teóricas de los acoplamientos en gráficas bipartitas	28
2.4 Algoritmos para la obtención del acoplamiento máximo en gráficas bipartitas	38
2.5 Aplicaciones del acoplamiento máximo	49
2.5.1 Sistemas de representantes distintos	49
2.5.2 Descomposición en cadenas	53
2.6 Acoplamiento Max-Min	59
2.7 El método Húngaro para el problema del acoplamiento de peso máximo	63
2.8 Acoplamiento Gale-Shapley	72

## Capítulo III

TEORIA DEL ACOPLAMIENTO EN GRAFICAS NO BIPARTITAS	75
3.1 Antecedentes	76
3.2 Arboles y flores	79
3.3 Algoritmo de solución al problema del acoplamiento máximo en gráficas no bipartitas	88
3.3.1 Construcción y etiquetación de nodos en flores	89
3.3.2 Algoritmo	93
3.3.3 Dualidad	98
3.4 Problema del acoplamiento de peso máximo para gráficas no bipartitas	102
3.4.1 Formulación del problema como problema de Programación Lineal	102
3.4.2 Dualidad	104
3.4.3 Algoritmo	107
3.4.4 Algunos aspectos necesarios para implementar un programa computacional que resuelva eficientemente el problema	113
3.5 Solución al problema del Cartero Chino	118

## Capítulo IV

COMENTARIOS REFERENTES AL PAQUETE COMPUTACIONAL	124
4.1 Elección de la estructura de datos para el programa	124
4.2 Estructura del programa	125
4.2.1 Definición de constantes y variables	129
4.2.1.1 Principales constantes y variables utilizadas para resolver los problemas de acoplamiento en gráficas bipartitas	129
4.2.1.2 Principales constantes y variables utilizadas para resolver los problemas de acoplamiento en gráficas generales	135
4.2.1.3 Principales constantes y variables utilizadas para resolver los problemas del acoplamiento de Gale-Shapley	139
4.2.2 Subrutinas	142
4.3 Manual del usuario	148
CONCLUSIONES	157
BIBLIOGRAFIA	158

## INTRODUCCION

Los problemas de acoplamiento forman parte de la Teoría de Gráficas, que se aplica en diversos campos de estudio: Ingeniería, Planeación, Administración, Química, Ciencias de la Computación, entre otras.

Las gráficas son una manera natural de representar un problema abstracto; así se pueden dibujar puntos que representen lugares, cosas o personas y utilizando líneas que los unan es posible representar alguna relación entre ellos.

Los problemas de acoplamiento surgen de diversas situaciones:

Supongamos que en un departamento de una universidad, se cuenta con  $n$  profesores que pueden impartir  $n$  cursos diferentes, se cuenta con evaluaciones sobre el rendimiento de asignar cada profesor a cada curso. ¿Cómo deben asignarse los cursos a los profesores para maximizar el rendimiento total a obtener?

Este problema en la teoría del acoplamiento se llama problema del acoplamiento de peso máximo; cuando el problema es pequeño, comparando todas las posibles soluciones podemos obtener la mejor. Sin embargo al aumentar el tamaño del problema, el número de soluciones posibles crece rápidamente, lo que haría imposible tratarlo por este método, por lo que es necesario utilizar técnicas de optimización para obtener la solución.

Los problemas de acoplamiento que pueden ser modelados en gráficas bipartitas tienen una estrecha relación con problemas de Asignación, Redes, Sistemas de Representantes Distintos y Conjuntos Parcialmente Ordenados, entre otros. Un resultado fundamental en la teoría del acoplamiento en gráficas bipartitas es el teorema de dualidad debido a König y que nos dice: la cardinalidad del acoplamiento máximo es igual a la cardinalidad de la cubierta de nodos mínima.

El primero en escribir sobre teoría de gráficas fue Euler en su famoso problema de los siete puentes de Königsberg (1736), dando origen a esta rama de las Matemáticas.

Los problemas sobre teoría del acoplamiento comenzaron a aparecer durante la segunda guerra mundial y con la publicación de los dos primeros libros de texto en teoría de gráficas después del libro de König (1936) debidos a Berge (1958) y Ore (1962) se da un crecimiento increíble en el estudio de la teoría de gráficas y en particular, de acoplamientos.

La importancia que toma la teoría del acoplamiento surge de preguntas tan sencillas como la siguiente ¿Cómo podemos encontrar un acoplamiento máximo, que no deje ningún nodo sin acoplar ?

Los antecedentes para encontrar acoplamientos máximos en gráficas bipartitas se encuentran en los trabajos de König y Egerváry en los años 30s. Kuhn (1955) y Hall (1956) presentaron el primer algoritmo formal para encontrar acoplamientos máximos en una bigráfica.

Parece ser que Kuhn utilizó por primera vez la frase "Método Húngaro" para algoritmos de este tipo.

En este período Ford y Fulkerson publicaron los primeros artículos de la teoría de flujo en redes (1956,1957,1961) que inmediatamente se utilizaron como una nueva herramienta en aplicaciones combinatorias de todos tipos. Para la teoría del acoplamiento estos fueron de gran importancia porque pueden ser usados para demostrar la mayoría de los resultados en acoplamientos sobre gráficas bipartitas.

Los acoplamientos en gráficas no bipartitas no fueron resueltos sino hasta 1965 con un algoritmo eficiente debido a Edmonds para encontrar acoplamientos máximos. Edmonds también propuso el término *tiempo polinomial* como medida de efectividad de un algoritmo, y que se ha desarrollado teóricamente en la Ciencia de la Computación.

El concepto de dualidad introducido por Von Neumann (1947) y el método simplex debido a Dantzig (1951), que resuelve eficientemente la mayoría de los problemas de Programación Lineal, fueron utilizados por Kuhn (1955,1956) para encontrar la solución al problema del acoplamiento de peso máximo en bigráficas. Para obtener un teorema combinatorio minímax de dualidad es necesario encontrar condiciones de integralidad en la solución óptima; en gráficas bipartitas se obtienen

fácilmente -unimodularidad total- y fueron demostradas por Hoffman y Kruskal (1956).

La extensión en el uso de la programación lineal al caso de gráficas no bipartitas requirió la introducción de nuevos conceptos como el de casco convexo de vectores de incidencia de acoplamientos por medio de desigualdades lineales. Tal conjunto de desigualdades lineales fue descubierto por Edmonds (1965). Este resultado permite obtener varios teoremas minimax en la teoría del acoplamiento como casos especiales de la teoría de dualidad de la programación lineal. Esta forma de tratar el problema dió origen a una nueva rama de la combinatoria: la combinatoria poliédrica.

El objetivo de este trabajo es analizar la parte teórica y algorítmica de la teoría del acoplamiento y que las soluciones propuestas se alcancen en tiempo polinomial. Se trata la implementación de un paquete computacional que resuelve eficientemente los siguientes problemas: acoplamiento máximo, acoplamiento de peso máximo en gráficas bipartitas y acoplamiento de Gale-Shapley, utilizando estructuras de datos adecuadas para lograr la eficiencia.

Conviene señalar que en la solución de los problemas se utilizan conceptos de programación lineal y entera, flujo en redes y teoría de gráficas dado que al combinarlas producen mejores algoritmos que los que se obtendrían si solamente utilizáramos alguna de ellas.

También se decidió implementar algunas soluciones en modo gráfico para la utilización del paquete con fines didácticos.

El presente trabajo se desarrolla de la siguiente manera :

En el primer capítulo veremos la terminología y conceptos básicos que se utilizan en el trabajo, así como algunos ejemplos que dieron origen a los diversos problemas de la teoría del acoplamiento.

En el segundo capítulo estudiamos la teoría del acoplamiento en gráficas bipartitas, lo relacionamos con la teoría de flujo en redes y especializamos el algoritmo de Flujo Máximo-Cortadura Mínima para obtener una solución eficiente al par de problemas duales acoplamiento máximo-cubierta de nodos mínima.

Se presenta el teorema minimax de König y las equivalencias con el teorema de Hall en Sistemas de Representantes Distintos y el teorema de Dilworth en Conjuntos Parcialmente Ordenados.

Para el problema del acoplamiento max-min se utiliza la programación lineal para obtener un algoritmo eficiente; para el problema del acoplamiento de peso máximo, presentamos el algoritmo conocido como Método Húngaro. Por último presentamos un tipo especial de acoplamiento que involucra preferencias llamado Gale-Shapley.

En el tercer capítulo presentamos el problema del acoplamiento máximo y el de peso máximo en gráficas no bipartitas.

Para el problema del acoplamiento máximo se definen los conceptos necesarios generalizando los utilizados en gráficas bipartitas. Un resultado dual al del acoplamiento de cardinalidad máxima, nos ayudará también en la solución del problema del acoplamiento de peso máximo, que se plantea como problema de programación lineal y en base a las condiciones de holgura complementaria y al método primal-dual de programación lineal se desarrolla un algoritmo de solución.

Por último se presenta la utilización de un problema de la teoría del acoplamiento -el problema del acoplamiento perfecto de peso mínimo- como subrutina en el algoritmo de solución al problema del cartero chino.

Los comentarios de la estructura e implementación del paquete computacional, así como el manual de usuario se presentan en el cuarto capítulo.

Por último presentamos las conclusiones de este trabajo y la bibliografía.

## CAPITULO 1

En este capítulo veremos los conceptos de la teoría de gráficas que utilizaremos en el presente trabajo. También definiremos los problemas básicos de la teoría del acoplamiento, así como los ejemplos de los cuales surgen. Comenzaremos con la definición de varios conceptos.

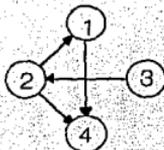
### 1.1. CONCEPTOS Y TERMINOLOGIA BASICA DE TEORIA DE GRAFICAS.

Una gráfica  $G$  consta de dos conjuntos,  $N(G)$  que es un conjunto finito no vacío de elementos llamados nodos o vértices y  $A(G)$  un conjunto finito formado por parejas de elementos de  $N$  llamados líneas. Se denota  $G = (N, A)$ .

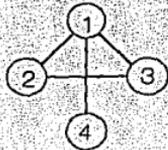
Notemos que están permitidas líneas múltiples entre dos nodos. No se permiten líneas de la forma  $(i, i)$  a menos que se especifique.

Si los elementos de  $A$  son parejas ordenadas entonces se les asocia una dirección y reciben el nombre de arcos; se dice que la gráfica  $G$  es dirigida, orientada o una digráfica. Si no tienen dirección se llaman aristas y  $G$  es no dirigida. Una multigráfica es una gráfica con aristas o arcos repetidos. Si una gráfica no tiene aristas o arcos repetidos se le llamará simple. Cuando en este trabajo hablemos de gráficas nos referiremos a gráficas simples.

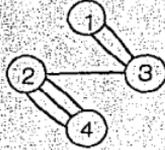
Si a los arcos o nodos de una gráfica  $G=(N,A)$  le asociamos un real o varios, mediante una función  $W : A \rightarrow R$  o  $V : N \rightarrow R$  o  $W : N \rightarrow R^k$  o  $W : (N,A) \rightarrow R^2$ , etc, decimos que  $G$  es una gráfica ponderada o una red y la denotamos  $G = (N, A, W)$  ó  $R = (N, A, W)$ .



Gráfica dirigida



Gráfica no dirigida



Multigráfica

Sea  $G=(N,A)$  una gráfica no dirigida.

Si la arista  $a \in A$  es de la forma  $(i,j)$ , decimos que  $a$  es incidente a  $i$  y  $j$ , así  $i$  y  $j$  son nodos adyacentes o vecinos. Dos aristas que tienen un nodo en común también son llamadas adyacentes. Definimos la función  $\Gamma : N \rightarrow p(N)$  ( $p(N)$  es el conjunto potencia de  $N$ ), como  $\Gamma(i) = \{\text{vecinos de } i\}$  para  $i \in N$ .

El número de aristas en  $G$  incidentes a un nodo  $i$  es llamado el grado de  $i$  y es denotado  $g(i)$ . Notemos que el grado de  $i \in N$  es la cardinalidad de  $\Gamma(i)$ , lo que no ocurre cuando se trata de multigráficas.

Si en una gráfica todos los grados son iguales a  $k$  decimos que la gráfica es  $k$ -regular.

Una secuencia de aristas en la cual el vértice final de uno es el vértice inicial del que le sigue en la secuencia es llamada una cadena. Si todas las aristas en una cadena son distintas entonces es una cadena simple y si además los nodos son diferentes entonces es una trayectoria. Estos conceptos se extienden también a digráficas.

La longitud de una cadena es el número de aristas en ella. Una cadena o cadena simple en la que el primer y último nodo son el mismo será cerrada.

Si una gráfica contiene una cadena simple y cerrada que incluye todas las aristas de  $G$ , la cadena es llamada un circuito Euleriano de  $G$  y una gráfica conteniendo un circuito Euleriano será llamada Euleriana.

Un ciclo es una trayectoria de longitud al menos dos, junto con una arista que une el primer y el último nodo; la longitud de un ciclo es el número de nodos en él. Un ciclo que contiene a todos los nodos de  $G$  recibe el nombre de ciclo Hamiltoniano.

Una gráfica que no contenga ciclos es llamada acíclica.

Para representar una cadena podemos usar la secuencia de aristas que la forman, la secuencia de vértices de sus aristas o la secuencia alternada de vértices y aristas. E igualmente podremos hacerlo para el caso de trayectorias y caminos (que se definirán más adelante para digráficas); también los representaremos, cuando no se preste a confusión, de la manera siguiente  $P : i_0 \rightarrow i_q$ , especificando si  $P$  es una trayectoria, cadena o camino que une al nodo  $i_0$  con el nodo  $i_q$ .

En la figura 1.2 a) observamos una gráfica no dirigida, en donde la secuencia  $1, a_1, 2, a_5, 6$  es una cadena simple de longitud dos. La secuencia  $1, a_1, 2, a_4, 5, a_6, 3, a_2, 2, a_5, 6, a_7, 3, a_3, 4, a_{10}, 6, a_9, 5, a_8, 1$  es un ciclo Euleriano (pero no un ciclo Hamiltoniano). En cambio la secuencia  $1, a_1, 2, a_2, 3, a_3, 4, a_{10}, 6, a_9, 5, a_8, 1$  si es un ciclo Hamiltoniano.

En la figura 1.2 b) encontramos una gráfica acíclica.

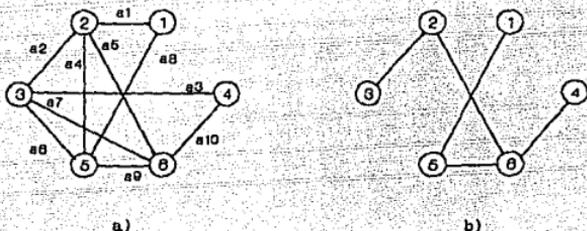


Figura 1.2

Un conjunto de nodos o aristas  $S$  en una gráfica es llamado minimal con respecto a la propiedad PROP, si el conjunto tiene la propiedad PROP, pero ningún subconjunto propio de  $S$  tiene esa propiedad. El conjunto  $S$  es llamado mínimo con respecto a la propiedad PROP, si de todos los subconjuntos de  $G$  que tengan la propiedad PROP,  $S$  es de cardinalidad mínima. En la figura 1.3 vemos una gráfica ponderada no dirigida, en la cual el conjunto de aristas  $\{(1,2), (3,4)\}$  es minimal con respecto a la propiedad que la suma del peso de las aristas sea mayor o igual a 2, sin embargo no es un conjunto mínimo. El conjunto  $\{(2,4)\}$  si es, en cambio, mínimo.

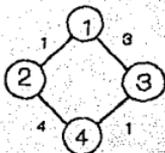


Figura 1.3

A continuación se define un tipo de gráfica muy especial sobre la cual diferentes problemas de la teoría del acoplamiento poseen una estructura particular que facilita su solución.

Si el conjunto de nodos de una gráfica  $G$  puede ser particionado en dos conjuntos ajenos no vacíos,  $S$  y  $T$ , tal que todas las aristas de  $G$  unen un nodo de  $S$  con un nodo de  $T$ , llamamos a  $G$  bipartita y nos referiremos a  $S$  y  $T$  como la bipartición. En este caso algunas veces llamamos a los conjuntos  $S$  y  $T$  las clases de color de  $G$ . Así, definimos una coloración de nodos de una gráfica  $G$ , como una asignación de enteros positivos (o colores) a los nodos de  $G$  tal que no pueden existir dos nodos adyacentes etiquetados con el mismo entero (o color).  $G$  es llamada  $n$ -colorable si  $G$  tiene una coloración de nodos minimal utilizando  $n$  colores.

Por lo tanto una gráfica bipartita es también llamada una gráfica 2-colorable o una bigráfica.

En la figura 1.4 se aprecia un ejemplo de una bigráfica, con los conjuntos  $S = \{ 1, 4, 5, 6, 8, 9 \}$  y  $T = \{ 2, 3, 7, 10 \}$ .

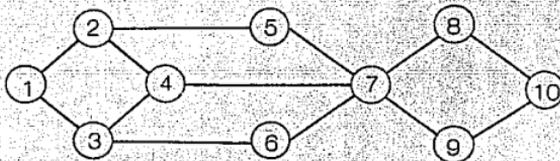


Figura 1.4

Sea  $G=(N,A)$  una gráfica dirigida.

Sea  $i \in N$ , llamamos sucesor de  $i$  a todo vértice  $j \in N$  tal que existe  $(i,j) \in A$ . Llamamos predecesor de  $i$  a todo vértice  $j \in N$  tal que existe  $(j,i) \in A$ . Definimos las funciones  $\Gamma^+, \Gamma^-, \Gamma : N \rightarrow p(N)$  como

$$\Gamma^+(i) = \{ \text{Sucesores de } i \} = \{ j \in N \mid (i,j) \in A \}, i \in N$$

$$\Gamma^-(i) = \{ \text{Predecesores de } i \} = \{ j \in N \mid (j,i) \in A \}, i \in N$$

$$\Gamma(i) = \Gamma^+(i) \cup \Gamma^-(i), i \in N$$

Un camino es una secuencia alternante de nodos y arcos en la cual el vértice final de un arco es el vértice inicial del que le sigue en la secuencia.

Un camino simple es un camino  $a_1, a_2, \dots, a_q$  tal que  $a_k \in A$  y  $a_k \neq a_l$  si  $k \neq l$ .

Un camino elemental es un camino  $i_1, i_2, \dots, i_q$  tal que  $i_k \in N$  e  $i_k \neq i_l$  si  $k \neq l$ .

Un circuito es un camino tal que  $a_1 = a_q$  con  $a_k \in A$ .

Sea  $i \in N$ . El grado exterior de  $i$  es el número de arcos que tienen a  $i$  como vértice inicial y se denota  $g^+(i)$ . El grado interior de  $i$  es el número de arcos que tienen a  $i$  como vértice final y se denota  $g^-(i)$ . Notemos que  $g^+(i) = |\Gamma^+(i)|$  y  $g^-(i) = |\Gamma^-(i)|$ .

Existen ciertas clases de gráficas que son de utilidad y que a continuación se describen:

Sea  $G = (N, A)$  una gráfica. Una gráfica parcial de  $G$  es una gráfica  $G_p = (N_p, A_p)$ , donde  $A_p \subset A$ . Una subgráfica inducida de  $G$  es la gráfica  $G_s = (N_s, A_s)$  donde  $N_s \subset N$  e  $(i, j) \in A_s$  si y sólo si  $i, j \in N_s$  e  $(i, j) \in A$ .

En una gráfica  $G$ , la contracción de un arco  $(i, j)$ , denotada como  $G \text{ ctr } \{(i, j)\}$ , consiste en el reemplazo de los nodos  $i$  y  $j$  por un nuevo nodo  $k$ , añadiendo en la gráfica resultante un arco (arista) por cada arco  $(i, m)$  ó  $(j, m)$  de la gráfica original. Notemos que con la contracción de un arco, la gráfica resultante puede ser una multigráfica.

En la figura 1.5 se ilustran estos últimos conceptos:

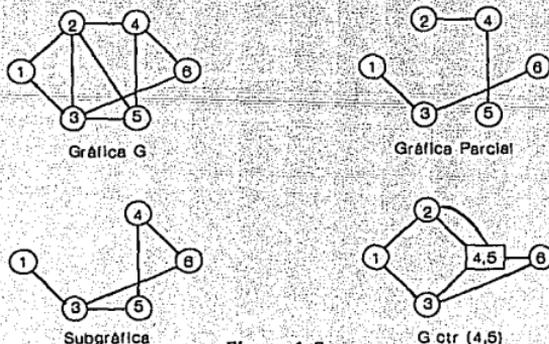


Figura 1.5

Algebraicamente las gráficas pueden ser representadas por medio de matrices.

Sea  $G=(N,A)$  una gráfica. La matriz de adyacencia de  $G$  es una matriz  $B$  cuadrada de  $n \times n$ , donde  $n$  es el número de vértices de  $G$ , con elementos

$$b_{ij} = \begin{cases} 1 & \text{Si } (i,j) \in A. \\ 0 & \text{Si } (i,j) \notin A. \end{cases}$$

La matriz de incidencia de  $G$  es una matriz  $M$  de dimensión  $n \times m$ , donde  $n$  y  $m$  son el número de vértices y de arcos respectivamente.

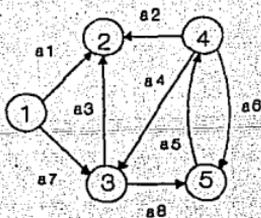
Si  $G$  es no dirigida, entonces los elementos de  $M$  son :

$$m_{ij} = \begin{cases} 1 & \text{Si la arista } j \text{ incide en el nodo } i \\ 0 & \text{En otro caso} \end{cases}$$

Si  $G$  es dirigida, entonces los elementos de  $M$  son :

$$m_{ij} = \begin{cases} 1 & \text{Si } i \text{ es extremo inicial del arco } a_j \\ -1 & \text{Si } i \text{ es extremo final del arco } a_j \\ 0 & \text{En otro caso.} \end{cases}$$

En la figura 1.6 observamos una digráfica, con sus respectivas matrices de adyacencia e incidencia.



$$B = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \end{matrix}$$

MATRIZ DE ADYACENCIA

$$M = \begin{matrix} & \begin{matrix} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & -1 & 1 \\ 0 & 1 & 0 & 1 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & -1 \end{bmatrix} \end{matrix}$$

MATRIZ DE INCIDENCIA

FIGURA 1.6

Observemos que en general, si se suman los elementos del  $i$ -ésimo renglón de la matriz de adyacencia para una digráfica, se obtiene  $g^+(i)$  para todo  $i \in N$ . Si se suman los elementos de la  $i$ -ésima columna se obtiene  $g^-(i)$ , para todo  $i \in N$ . Para una gráfica no dirigida, si sumamos los elementos del  $i$ -ésimo renglón o de la  $i$ -ésima columna de la matriz de adyacencia obtenemos  $g(i)$ .

Notemos también que por construcción la matriz de adyacencia ( a veces llamada también matriz de nodos vs nodos ) de una gráfica no dirigida es siempre simétrica. ( lo que no ocurre para una digráfica ).

Otra posible representación para una gráfica es por medio de listas. La lista de adyacencia para un vértice  $i \in N$  es una lista de todos los vértices  $j$  adyacentes a  $i$ . Una gráfica puede ser representada por  $|N|$  listas de adyacencia, una para cada vértice.

En la figura 1.7 a) vemos las listas de adyacencia para cada vértice  $i \in N$ , para la digráfica de la figura 1.6. Una representación tabular de las listas de adyacencia se muestra en la figura 1.7 b), las primeras cinco entradas corresponden a vértices y el arreglo apuntador[i] nos señala el primer vértice de la lista de adyacencia para  $i$ . Cuando apuntador[i] = 0 no hay vértices adyacentes. Las entradas restantes del arreglo representan los arcos de la gráfica. El arreglo valor contiene los vértices de las listas de adyacencia.

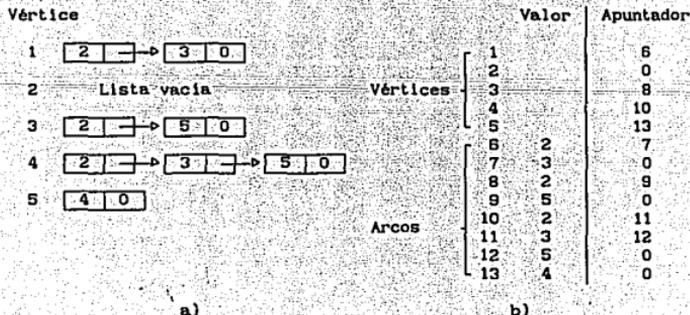


Figura 1.7

La representación de una digráfica por medio de listas de adyacencia requiere de memoria proporcional a  $|N|+|A|$ . Para una gráfica no dirigida aumenta a  $|N| + 2|A|$ , porque cada arco es representado dos veces, por ejemplo el arco  $(i,j)$  tiene al nodo  $j$  en la lista de adyacencia del nodo  $i$  y viceversa.

Así, esta representación es usada a menudo cuando  $|A| \ll |N|^2$ .

Un concepto muy importante es el de conexidad, por que puede relacionarse fácilmente con la idea intuitiva de poder unir dos nodos cualesquiera por medio de una trayectoria. La formalización de esta idea es como sigue:

Sea  $G=(N,A)$  una gráfica, decimos que  $G$  es conexa si para todo par de nodos  $i,j \in N$  existe una cadena que los une. Si una gráfica es conexa y carece de ciclos es un árbol. Este último concepto es muy utilizado en la solución de varios problemas de teoría de gráficas, de redes y en aplicaciones en la ciencia de la computación.

Decimos que una digráfica es completa si para cualesquiera dos nodos  $i,j \in N$ , existe el arco  $(i,j) \in A$  ó existe el arco  $(j,i) \in A$ .

Por último mencionaremos otro concepto importante que en algunos problemas nos sirve como herramienta para detectar cuando se alcanza la óptimalidad.

Un conjunto de arcos o aristas  $A'$  en una gráfica conexa  $G$  es una cortadura o corte si la subgráfica  $G_{A'} = (N, A - A')$  es no conexa. Una definición similar es dada para un conjunto de nodos.

Como una parte de la tesis tratará con gráficas bipartitas enseguida se caracterizarán este tipo de gráficas por medio de dos proposiciones:

Sea  $G = (S,T,A)$  una gráfica bipartita, donde  $S,T$  es la bipartición de los nodos de  $N$ . Si en una bigráfica tomamos una trayectoria y la recorremos es fácil ver que al avanzar por los arcos se tocan nodos que alternativamente pertenecen a  $S$  y a  $T$ . Partiendo de esta observación, podemos concluir que si  $G$  es bipartita, entonces, cualquier ciclo en ella necesariamente tiene longitud par. El recíproco también es cierto y el resultado se enuncia en la siguiente proposición.

### Teorema 1.1

Una gráfica  $G = (N, A)$  es bipartita si y sólo si carece de ciclos de longitud impar.

#### Demostración.

##### *Condición necesaria.*

Como  $G$  es bipartita existen conjuntos  $S, T$  tales que  $S \cup T = N$  y  $S \cap T = \emptyset$ , tales que cada arco en  $A$  tiene un extremo en  $S$  y otro en  $T$ .

Supongamos que en la gráfica existe un ciclo de cardinalidad impar  $i_1, i_2, \dots, i_k = i_1$ , supongamos además que  $i_1 \in S$  (el otro caso es análogo), esto nos lleva a que  $i_2 \in T, i_3 \in S$ , etc., es decir  $i_j \in S$  para  $j$  impar e  $i_j \in T$  para  $j$  par. Si la longitud del ciclo es impar,  $i_{k-1}$  debe ser impar, por lo que  $i_k$  es par, entonces  $i_k \in T$ , pero  $i_k = i_1 \in S$ !

Por lo tanto, la gráfica no contiene ciclos de cardinalidad impar.

##### *Condición suficiente.*

Supongamos que cada ciclo de  $G$  tiene longitud par. Elija cualquier nodo  $i$  de  $G$  y píntelo de color azul. Después realicemos el siguiente procedimiento iterativo: colorear de gris los nodos adyacentes a uno pintado de azul y de azul los adyacentes a uno pintado de gris. Realicemos este proceso de coloración hasta que :

- i) Todos los nodos están coloreados y todo arco tiene un extremo azul y otro gris.
- ii) Algún nodo  $i'$  pintado de un color puede ser pintado de otro color.
- iii) Todos los nodos adyacentes a nodos coloreados también lo están pero existen algunos no coloreados.

En el caso i) defina los conjuntos  $S$  y  $T$  formados por los nodos coloreados de azul y de gris respectivamente. En este caso  $S \cup T$  son los nodos de  $G$ ,  $S \cap T = \emptyset$  y todos los arcos tienen un extremo en  $S$  y el otro en  $T$ , por lo tanto  $G$  es bipartita.

En el caso ii) el nodo  $i'$  recibe color azul mediante una trayectoria  $P_1: i \rightarrow i'$  en la cual los nodos están coloreados alternativamente de azul y gris; análogamente recibe color gris mediante una trayectoria  $P_2: i \rightarrow i'$  con las mismas características. Sea  $i'$  el penúltimo nodo común a las dos trayectorias. Sean  $P_1': i' \rightarrow i'$  y  $P_2': i' \rightarrow i'$  las trayectorias distintas, partes de  $P_1$  y  $P_2$  respectivamente. Si  $i'$  tiene color azul entonces  $P_1$  es de cardinalidad par y  $P_2$  de cardinalidad non. Pero la unión de  $P_1$  y  $P_2$  constituye un ciclo de cardinalidad impar lo cual contradice la hipótesis. Por lo tanto este caso no puede presentarse.

En el caso iii) la red resultó ser no conexa. Repitiendo el proceso de coloración en cada componente conexa de  $G$ , éste se reduce al caso i).

De esto se concluye que toda red que no contenga ciclos de longitud impar es bipartita. ■

La segunda caracterización de una gráfica bipartita consiste en agrupar los nodos de cada conjunto de la bipartición en la matriz de adyacencia y como por definición dos nodos que se encuentren en el mismo conjunto no son adyacentes, se generarán dos submatrices, de las cuatro formadas con esta agrupación, con todas sus entradas cero. El resultado se enuncia en la siguiente proposición.

Proposición 1.2

$G$  es una gráfica bipartita si y sólo si sus nodos pueden numerarse de manera tal que la matriz de adyacencia sea de la forma:

$$A = \begin{matrix} & \begin{matrix} S & T \end{matrix} \\ \begin{matrix} S \\ T \end{matrix} & \left[ \begin{array}{c|c} 0 & B \\ \hline B^t & 0 \end{array} \right] \end{matrix}$$

donde  $B$  es una submatriz de  $A$ , cuyo elementos son cero o uno.

Habíamos dicho que la gráfica de la figura 1.3 es una gráfica bipartita, su matriz de adyacencia es la siguiente:



## 1.2 PROBLEMAS QUE DAN ORIGEN A LA TEORIA DEL ACOPLAMIENTO.

En esta sección veremos algunos problemas que fueron el origen y que pueden ser resueltos por medio de la teoría del acoplamiento, comenzaremos con la definición formal de acoplamiento.

Sea  $G=(N,A)$  una gráfica, un acoplamiento  $M$  es un subconjunto de arcos  $M \subseteq A$  tal que dos arcos no incidan en un nodo común.

A la cardinalidad del acoplamiento máximo en  $G$  se le llama el número de acoplamiento de  $G$  y es denotado  $\nu(G)$ .

Un subconjunto de nodos  $S \subseteq N$  es una cubierta de nodos si cada arco de  $G$  tiene al menos un extremo en el conjunto  $S$ .

La cardinalidad de la cubierta de nodos mínima es denotada por  $\tau(G)$  y es conocida como la cardinalidad de la cubierta de nodos.

Un nodo expuesto relativo al acoplamiento  $M$  es un nodo que no es incidente a ningún arco del acoplamiento.

Un acoplamiento que no deja nodos expuestos es llamado acoplamiento perfecto o completo. Es claro que un acoplamiento completo requiere que  $|N|$  sea par.

En el capítulo dos se demuestra que para cualquier gráfica  $G$  se cumple  $\nu(G) \leq \tau(G)$ ; para el caso de gráficas bipartitas se da la igualdad, lo que nos permite obtener, a partir de la solución de uno de los problemas, la solución al otro.

En la figura 1.8 los arcos ondulados forman parte de un acoplamiento, los cuadros en los nodos una cubierta de nodos. El acoplamiento en a) no es de máxima cardinalidad, en cambio el de b) si lo es. La cubierta de nodos en c) no es de mínima cardinalidad, en d) se muestra una cubierta de cardinalidad mínima.

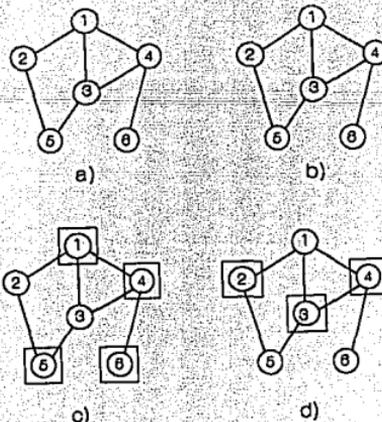


Figura 1.8

Los problemas a resolver de la teoría del acoplamiento en gráficas pueden consistir de diferentes criterios de optimización, como son maximizar el número de arcos de un acoplamiento, maximizar o minimizar el peso del acoplamiento, o el número de arcos que puedan incidir en un nodo, en este trabajo el máximo número de arcos que pueden incidir en un nodo es uno, que es el problema original de la teoría del acoplamiento; si se permite incidir un número mayor de arcos lo consideraremos una generalización de este problema. Enseguida se darán algunos problemas básicos en la teoría del acoplamiento con ejemplos que los motivaron:

### 1) PROBLEMA DEL ACOPLAMIENTO MÁXIMO.

Dada una gráfica  $G = (N, A)$  se desea determinar el acoplamiento con el mayor número de arcos posibles, es decir el de máxima cardinalidad.

#### Ejemplo 1.1

Spongamos que una firma farmacéutica desea probar  $n$  antibióticos en  $n$  personas voluntarias. Sin embargo pruebas preliminares mostraron que ciertas personas son alérgicas a ciertos antibióticos. ¿Puede diseñarse un experimento en el cual cada persona tome exactamente un antibiótico al que no sea alérgico y que cada antibiótico sea tomado por exactamente una persona ?

Modelando esta situación con una gráfica bipartita  $G$  en las que los dos subconjuntos de la bipartición  $S, T$  consisten de las  $n$  personas y de los  $n$  antibióticos respectivamente y uniendo con un arco a la persona con el antibiótico si y sólo si el sujeto no es alérgico a él, la respuesta a la pregunta realizada es "sí", si y sólo si el acoplamiento de máxima cardinalidad en la gráfica  $G$  no deja nodos expuestos, es decir es un acoplamiento perfecto, por lo tanto es un apareamiento de personas con antibióticos, que use a cada persona y a cada antibiótico una y sólo una vez.

### Ejemplo 1.2

Ahora supongamos que tenemos dos computadoras disponibles y  $p$  trabajos para ser procesados en estas máquinas. Supondremos que cualquier trabajo puede ser corrido en cualquier máquina. Es decir, suponemos que las computadoras son idénticas. Supongamos además que los  $p$  trabajos diferentes pueden ser ordenados de tal manera que para dos trabajos  $J_i$  y  $J_k$  si  $J_i \leq J_k \rightarrow J_i$  debe ser completado antes de que  $J_k$  pueda ser iniciado. Si todos los trabajos requieren una cantidad igual de tiempo para ser acabados, ¿Cuál es el tiempo más corto posible, suficiente para correr los  $p$  trabajos ?

Para modelar esta situación usaremos una gráfica  $G$  como sigue: Sean los nodos de  $G$  los trabajos  $J_1, \dots, J_p$  y unamos por un arco  $J_i$  y  $J_k$  si y sólo si no son comparables en el orden parcial (En otras palabras, si pueden correrse simultáneamente). Es claro, entonces que para diseñar un horario óptimo, debemos usar ambas máquinas simultáneamente tanto como sea posible: esto es debemos encontrar un acoplamiento de cardinalidad máxima. Notemos que este problema ya no utiliza una gráfica bipartita.

### II) *PROBLEMA DEL ACOPLAMIENTO Max-Min ó PROBLEMA DEL CUELLO DE BOTELLA.*

Dada una gráfica ponderada  $G = (N, A)$  el problema es encontrar el acoplamiento de máxima cardinalidad cuyo arco de peso mínimo tenga el mayor peso posible.

### Ejemplo 2.1

El planteamiento más común para este problema es el siguiente: En un taller industrial se tienen " $n$ " trabajadores para ser asignados a " $n$ " estaciones de trabajo de una línea de producción en serie. Como los obreros tienen distintas velocidades de trabajo en las diferentes estaciones de la línea, es claro que para cada asignación habrá al menos una estación que sea la más lenta de todas -"El cuello de botella"- y que determinará la velocidad de producción de toda la línea. Modelando este problema en una bigráfica, los obreros y las estaciones de trabajo son los nodos de los conjuntos  $S$  y  $T$  respectivamente, con la velocidad del obrero  $i$  en la tarea  $j$  como el

peso de la arista  $(i, j)$ . El problema puede ser resuelto encontrando la asignación (acoplamiento perfecto obreros-estaciones de trabajo) donde la más lenta de las estaciones sea la más rápida posible.

### III) PROBLEMA DEL ACOPLAMIENTO DE PESO MÁXIMO (MÍNIMO).

Sea  $G = (N, A)$  una gráfica ponderada. El objetivo es encontrar un acoplamiento en el cual la suma de los pesos de sus arcos sea máxima. Cuando  $G$  es bipartita y  $|S| = |T|$  el problema del acoplamiento ponderado es fácilmente reconocido como un problema de asignación, pues es una modelación natural a la búsqueda de la asignación óptima de trabajadores (elementos del conjunto  $S$ ) a tareas (elementos de  $T$ ) cuando se conoce la eficiencia del obrero  $i$  en la tarea  $j$ , que se denota por el peso  $w_{ij}$ .

#### Ejemplo 3.1

En un área de un departamento, se cuenta con 5 profesores que pueden impartir solamente un curso de entre 5 diferentes; se cuenta con evaluaciones sobre el rendimiento, en escala del 1 al 10, que se obtienen de asignar cada profesor a cada curso de acuerdo a la siguiente tabla:

	CURSO	1	2	3	4	5
PROFESOR	A	10	8	6	10	10
B	9	9	8	9	8	
C	9	9	7	9	9	
D	10	8	9	10	9	
E	10	8	9	8	10	

¿Cómo deben asignarse los cursos a los profesores para maximizar el rendimiento total a obtener?

La respuesta a esta pregunta se obtiene con un acoplamiento perfecto de peso máximo.

Observemos que si la pregunta fuera ¿Cómo deben asignarse los cursos a los profesores para que el rendimiento mínimo de un grupo sea máximo?, la respuesta la podemos dar resolviendo el problema del acoplamiento max-min.

#### IV) ACOPLAMIENTO DE GALE-SHAPLEY

En este problema los elementos involucrados tienen una preferencia en la formación de parejas y esto nos permite manejar la característica de estabilidad en las parejas formadas y que aparece en algunos problemas de acoplamiento.

##### Ejemplo

Supongamos que  $n$  hombres y  $n$  mujeres elaboran lista de preferencia de las personas del sexo opuesto para contraer matrimonio. Entonces, para cualquier acoplamiento dado, siempre podremos preguntar si habrá un hombre y una mujer que no forman pareja entre sí, pero se prefieren más el uno al otro que el posible conyuge asignado por el acoplamiento. Cuando se da este caso, el conjunto de parejas propuesto no es estable, pues al menos habrá un hombre y una mujer dispuestos a dejar a sus respectivos conyuges para formar pareja entre sí. Cuando un acoplamiento es estable (nadie desea de sustituir a su pareja recibe respuesta) y además ocurre que cualquier hombre está por lo menos tan satisfecho en él como en cualquier otro acoplamiento estable, se tiene un acoplamiento óptimo (para los hombres) llamado hombre-óptimo; en la descripción del algoritmo veremos que en realidad quien determina un acoplamiento hombre-óptimo son las mujeres. El Teorema de Gale-Shapley es una prueba constructiva de que tal óptimo existe, sin importar el orden en las listas de preferencia.

Este modelo se puede utilizar para diversos problemas tales como: la asignación de solicitantes de empleo a puestos, de profesores a cursos universitarios, de médicos pasantes a hospitales.

##### V) Otros problemas

Algunos problemas que no son directamente convertibles en problemas de acoplamiento pueden, sin embargo, ser resueltos con la ayuda de la teoría del acoplamiento.

El Problema del Cartero Chino (Mel-ko Kwan [16] ) fue presentado como sigue: Un cartero debe recorrer, entregando correspondencia, todas las calles de una población. ¿Cómo debe él dejar la oficina de correos, realizar el recorrido completo de todas las calles entregando correspondencia y regresar a la oficina postal habiendo recorrido la mínima distancia ?

Aquí no existe una translación del problema directamente a un problema de acoplamiento. Posteriormente se verá que este problema puede ser resuelto eficientemente resolviendo primero un conjunto de problemas de rutas más cortas y resolviendo entonces un problema de acoplamiento perfecto de peso mínimo.

Modificando el problema, supongamos ahora que el cartero sólo tiene que recoger la correspondencia de los buzones localizados únicamente en la intersecciones de las calles. De nuevo, él desea encontrar una ruta más corta que resuelva su problema. La principal diferencia entre este problema y el anterior es que sólo tiene que visitar todas las intersecciones pero no atravesar todas las calles. Una formulación más común de este problema es la siguiente: Un agente viajero tiene que visitar todas las capitales de los estados de algún país de modo tal que minimice la longitud ( o el tiempo, o el costo) de la ruta. Este problema es conocido como el Problema del agente viajero y ha recibido gran atención por sus aplicaciones al mundo "real".

Como un caso especial, el vendedor puede estar interesado en encontrar una ruta que evite ciertas conexiones. En tal caso utilizamos los arcos que representen las conexiones permitidas y borramos los demás arcos. El problema se reduce a encontrar un Circuito Hamiltoniano (i.e. un ciclo a través de todos los nodos) en la gráfica dada. Este problema es llamado el Problema del circuito Hamiltoniano.

Superficialmente el problema del agente viajero parece muy similar al problema del cartero chino y al problema del acoplamiento ponderado perfecto, sin embargo el problema del agente viajero ( y aún su caso especial, el problema del circuito Hamiltoniano) son significativamente más difíciles. Particularmente el problema del cartero chino es resuelto en tiempo polinomial, pero el problema del agente viajero es "NP-duro" (i.e. pertenece a los más difíciles problemas combinatorios).

## CAPITULO 2

En este capítulo trataremos la teoría del acoplamiento en gráficas bipartitas; comenzaremos demostrando las relaciones existentes entre la teoría del acoplamiento en gráficas bipartitas y la teoría de flujo en redes. Se adaptan tres teoremas de la teoría de flujo en redes al contexto del acoplamiento en gráficas bipartitas: El teorema de la trayectoria aumentante, el teorema de integralidad y el teorema de dualidad flujo máximo-cortadura mínima. Después veremos dos algoritmos para la obtención de un acoplamiento de máxima cardinalidad en gráficas bipartitas, uno es una especialización del algoritmo de Ford y Fulkerson para flujos y el otro basado en el teorema de la trayectoria aumentante, se mencionan dos problemas de tipo combinatorio que pueden ser resueltos por medio del acoplamiento máximo en bigráficas. Se tratan además el problema del acoplamiento de peso máximo y el problema del acoplamiento max-mín o cuello de botella en bigráficas, por último se trata el acoplamiento de Gale-Shapley.

### 2.1 ANTECEDENTES

Los acoplamientos en gráficas bipartitas han sido estudiados desde la década de los 50, tanto en Investigación de Operaciones (I. O.) como en Análisis Combinatorio con diferente terminología y motivaciones: En I. O. uno de los primeros problemas de optimización que se estudiaron fue el problema de asignación, el cual consiste en encontrar un subconjunto de elementos de una matriz dada de  $n \times n$ , tal que haya exactamente un elemento en cada renglón, uno en cada columna y que la suma de los elementos sea mínima. El problema de asignación es equivalente al problema del acoplamiento perfecto de peso mínimo en bigráficas. En Análisis Combinatorio tenemos un resultado muy conocido e importante que es el teorema de Hall y que responde a la pregunta siguiente: Supongamos que tenemos un conjunto finito  $T$  y una colección

de subconjuntos de  $T : \{T_1, T_2, \dots, T_k\}$  ¿ Cuándo es posible elegir un elemento de cada subconjunto  $T_j$ , de modo que los  $k$  elementos escogidos sean distintos entre sí?. Este conjunto es llamado un Sistema de Representantes Distintos (SDR por sus siglas en inglés). El problema de determinar un SDR es equivalente al problema de encontrar un acoplamiento de cardinalidad máxima en bigráficas. Estas dos equivalencias serán demostradas posteriormente.

Gracias a las técnicas para resolver estos problemas en el caso de bigráficas, junto con un tratamiento muy ingenioso debido a Edmonds [9], los problemas de acoplamientos en gráficas generales fueron resueltos eficientemente a mitad de la década de los 60.

## 2.2 EQUIVALENCIAS ENTRE PROBLEMAS DE ACOPLAMIENTO EN GRAFICAS BIPARTITAS Y PROBLEMAS DE OPTIMIZACION.

En esta sección veremos dos equivalencias de problemas de acoplamiento en gráficas bipartitas con problemas de optimización. Primero veremos que a todo problema del acoplamiento máximo en bigráficas le corresponde un problema de flujo máximo, posteriormente veremos que el problema del acoplamiento perfecto de peso mínimo en bigráficas y el problema de asignación son equivalentes.

### Lema 2.1

La cardinalidad de un acoplamiento máximo en una gráfica bipartita  $B=(S,T,A)$  con  $m+n$  nodos es igual al valor del flujo máximo  $o \rightarrow d$  en una red  $G(B)$  con  $m+n+2$  nodos.

### Demostración.

Sea  $B = (S, T, A)$  una gráfica bipartita y sean  $|S| = m$   $|T| = n$ . Construyamos una gráfica dirigida  $G(B) = (S \cup T \cup \{o\} \cup \{d\}, A \cup \{U1\} \cup \{U2\})$  con  $m + n + 2$  nodos y  $|A| + m + n$  arcos.

Con arcos  $(S_i, T_j)$  con capacidad  $\infty$  si  $(S_i, T_j) \in A$ .

$U1 = \{ \text{arcos } (o, i) \text{ para todo } i \in S, \text{ con capacidad } 1 \}$ .

$U2 = \{ \text{arcos } (j, d) \text{ para todo } j \in T, \text{ con capacidad } 1 \}$ .

Ahora, dado  $X$  un acoplamiento en la gráfica bipartita  $B$ , construimos un flujo factible en  $G(B)$  como sigue:

Para cada arista  $(i,j) \in X$  asignemos una unidad de flujo a los arcos  $(o,i)$ ,  $(j,d)$  e  $(i,j)$  en  $G(B)$ , claramente el flujo es factible y de valor  $|X|$ .

Recíprocamente, dado un flujo entero de valor  $V$  en  $G(B)$  existe un acoplamiento de cardinalidad  $V$ . En efecto, consideremos el conjunto  $X$  formado por todos los arcos  $(s,t) \in A$  que tengan flujo 1. De la construcción de  $G(B)$ , tenemos que  $X$  es un acoplamiento, porque si dos arcos en  $A$  inciden en un nodo  $u$ , uno de los arcos correspondientes en  $G(B)$  debe tener flujo cero pues éste debe ser entero, de otra forma la capacidad de los arcos  $(o,u)$  ó  $(u,d)$  si  $u \in S \cup T$  es violada. Además el acoplamiento tiene cardinalidad igual a  $V$ , porque todos los extremos  $t$  de los arcos  $(s,t)$  en el acoplamiento en  $G(B)$  permiten salir exactamente una unidad de flujo ya que tienen grado exterior uno. ■

En la gráfica siguiente se muestra la relación existente entre el problema del acoplamiento máximo en bigráficas y su correspondiente problema de flujos en redes. En la figura siguiente los números entre paréntesis  $(f_{ij}, q_{ij})$  representan el flujo y la capacidad del arco  $(i,j)$

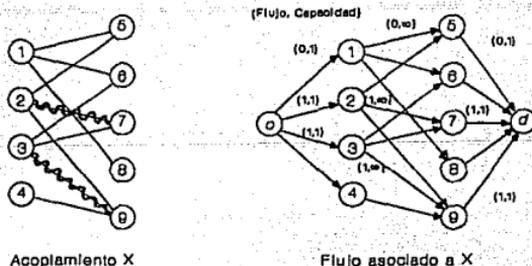


Figura 2.1

El problema de asignación es ampliamente conocido en su versión de minimización y consiste en encontrar un subconjunto de elementos de una matriz de  $n \times n$ , tal que haya exactamente un elemento en cada renglón, uno en cada columna y que la suma de los elementos sea mínima.

La modelación natural para este problema, en términos de gráficas, consiste en construir una bigráfica completa  $B = (S, T, A)$  con  $|S| = |T| = n$ , donde cada nodo en  $S$  y cada nodo en  $T$  corresponden respectivamente, a un renglón y a una columna de la matriz de asignación y el arco  $(i, j) \in A$  si y sólo si  $i \in S$  y  $j \in T$ .

Es fácil ver que a partir de esta bigráfica, existen 2 maneras de resolver este problema: El primero es asociar la gráfica de flujos  $G(B)$ , y resolver un problema de flujo a costo mínimo. Cada arco de la forma  $(o, i)$  o  $(j, d)$  de la red tendrá costo cero y los arcos restantes tendrán como costo la entrada  $[i, j]$  de la matriz del problema de asignación. El segundo es utilizar un algoritmo para encontrar un acoplamiento perfecto de peso mínimo en  $B$ . Veremos que el problema de asignación y el problema del acoplamiento perfecto de peso mínimo son equivalentes.

Al modelar el problema de asignación en una bigráfica completa, una solución óptima tendrá  $n$  arcos de la forma  $(i, j)$  con  $i \in S$  y  $j \in T$ , no dejando nodos expuestos, y por lo tanto como cada nodo  $i \in S$  está relacionado con un nodo  $j \in T$  y no existen nodos expuestos, esta solución es un acoplamiento perfecto de peso mínimo.

Inversamente para construir un problema de asignación a partir de uno de acoplamiento perfecto de peso mínimo en bigráficas haremos lo siguiente:

Sea  $G = (S, T, A)$  con  $|S| = n = |T|$  una gráfica en la que se define un problema de acoplamiento perfecto de peso mínimo. Construyamos una gráfica  $G' = (S, T, A')$  con conjunto de arcos  $A' = A \cup A''$  con  $A'' = \{ (i, j), \text{ si } (i, j) \in A, \text{ con } w_{ij} = \infty \}$ , es decir agregamos arcos ficticios, con un peso infinito, para tener una bigráfica completa. Si los nodos de  $S$  los interpretamos como renglones y a los nodos de  $T$  como columnas, la solución al problema de acoplamiento perfecto de peso mínimo así planteado, consiste en encontrar un acoplamiento de renglones con columnas que cubre los requisitos para ser una asignación. Por lo tanto los problemas de asignación (en el caso de minimización) y el problema del acoplamiento perfecto de peso mínimo son equivalentes.

### 2.3 ALGUNAS PROPIEDADES TEORICAS DE LOS ACOPLAMIENTOS EN GRAFICAS BIPARTITAS

En esta sección veremos algunas propiedades de los acoplamientos en bigráficas, así como la adaptación de 3 teoremas de la teoría de flujos en redes al contexto de acoplamiento en gráficas bipartitas. Estos tres teoremas son el teorema de la trayectoria aumentante, el teorema de integralidad y el teorema de dualidad flujo máximo-cortadura mínima.

Sea  $G = (N, A)$  una gráfica y  $X$  un acoplamiento de ella.

Una trayectoria alternante relativa al acoplamiento  $X$  es una trayectoria  $a_1, a_2, \dots, a_n$  tal que si  $a_i \in X, a_{i+1} \notin X$ .

Una trayectoria aumentante relativa al acoplamiento  $X$  es una trayectoria alternante que une dos nodos expuestos.

En la figura 2.2 a) vemos una gráfica con un acoplamiento formado por el conjunto de aristas  $X = \{ (1,4), (3,5) \}$ . La trayectoria  $(6,4), (4,1), (1,3), (3,5)$  es alternante y la trayectoria  $(6,4), (4,1), (1,2)$  es aumentante con respecto al acoplamiento. Se muestran respectivamente en la figura 2.2 b) y c) respectivamente.

El concepto de trayectoria aumentante es de gran importancia pues es una herramienta de optimalidad.

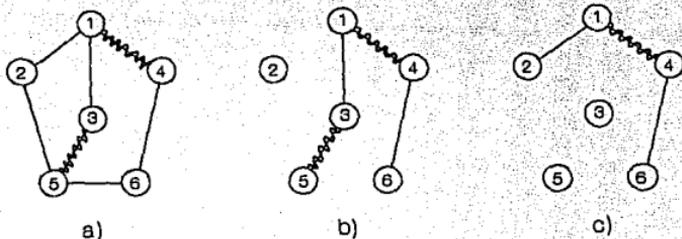


Figura 2.2

Particularmente el teorema de la trayectoria aumentante se generaliza a cualquier gráfica, no solamente gráficas bipartitas. Este teorema es debido a Berge [5] y nos dice que un acoplamiento  $X$  contiene un número máximo de arcos si y sólo si no existen trayectorias aumentantes con respecto a él. Para demostrar este teorema, utilizaremos los siguientes dos resultados:

### Teorema 2.1

Sea  $G=(N,A)$  una gráfica, sea  $X$  un acoplamiento y sea  $P$  una trayectoria aumentante con un conjunto de arcos  $E'$ , donde  $|E'| = m$ .

Sea  $X' = X \cup E'$ . Entonces

$$(X-X') \cup (E'-X') = (X-E') \cup (E'-X)$$

es un acoplamiento con un arco más que  $X$ .

### Demostración.

$P$  comienza y termina con arcos que no están en  $X$ , así que hay un número  $k$  tal que  $P$  tiene  $k$  arcos en  $X$  y  $k+1$  arcos que no están en  $X$ . Pero el primer y el último vértices de  $P$  son no acoplados y todos los demás vértices en  $P$  están acoplados por  $X'$ , así que ningún arco  $X-X'$  es adyacente a algún vértice de  $P$ . De esta forma los arcos  $X-X'$  no tienen extremos en común con los arcos  $E'-X'$ . Más aún, dado que  $P$  es una trayectoria alternante y  $E'-X'$  consta de los arcos de ella que no están en el acoplamiento,  $E'-X'$  no son adyacentes. Por lo tanto  $(X-X') \cup (E'-X')$  es un acoplamiento que posee  $m-k+k+1 = m+1$  arcos.

Por teoría de conjuntos se puede demostrar fácilmente que

$$(X-X') \cup (E'-X') = (X-E') \cup (E'-X) \blacksquare$$

La importancia de este teorema es que al detectar una trayectoria aumentante con respecto a un acoplamiento, podemos construir un acoplamiento de cardinalidad mayor en uno al anterior, simplemente intercambiando los arcos de la trayectoria aumentante que pertenezcan al acoplamiento por aquellos de la trayectoria que no pertenecen al acoplamiento. Un ejemplo se muestra en la figura 2.3

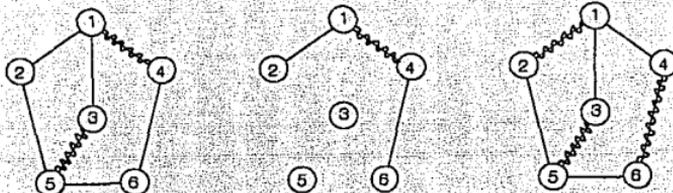


Figura 2.3

**Teorema 2.2**

Sean  $X_1$  y  $X_2$  dos acoplamientos en una gráfica  $G=(N,A)$ . Sea  $G'=(N',A')$  la subgráfica inducida de  $G$  con un conjunto de arcos

$$X_1 \oplus X_2 = (X_1 - X_2) \cup (X_2 - X_1).$$

es decir  $G$  contiene a los arcos que están en uno sólo de los dos acoplamientos; entonces cada componente conexa de  $G'$  es de uno de los siguientes tipos

- 1.- Un circuito de longitud par cuyos arcos están alternadamente en  $X_1$  y en  $X_2$ .
- 2.- Una trayectoria cuyos arcos están alternadamente en  $X_1$  y  $X_2$  y cuyos vértices finales no están saturados en alguno de los dos acoplamientos.

**Demostración.** Sea  $v \in N'$

Caso 1  $v \in N(X_1 - X_2)$  y  $v \notin N(X_2 - X_1)$

donde  $N(X_i - X_j)$  denota el conjunto de vértices de los arcos en  $X_i - X_j$ .

En este caso  $v$  es el extremo de un arco en  $X_1 - X_2$  pero  $X_1$  es un acoplamiento y ningún otro arco de  $X_1 - X_2$  es incidente a  $v$ . Aun más, ningún otro arco de  $X_2 - X_1$  es incidente a  $v$ , por que  $v \notin N(X_2 - X_1)$ . Por lo tanto en este caso el grado de  $v$  en  $G'$  es igual a 1.

Caso 2  $v \in N(X_1 - X_2)$  y  $v \in N(X_2 - X_1)$  en este caso un único arco de  $X_1 - X_2$  es incidente a  $v$  y un arco único de  $X_2 - X_1$  es incidente a  $v$ . Por lo tanto el grado de  $v$  es igual a 2.

Como los 2 casos son exhaustivos, el grado máximo en  $G'$  es 2 y las componentes conexas serán de uno de los tipos descritos en el teorema. ■

### Teorema 2.3

### Teorema de la Trayectoria Aumentante.

Sea  $G = (N, A)$  una gráfica y  $X$  un acoplamiento,  $X$  es un acoplamiento de máxima cardinalidad si y sólo si no existe en la gráfica una trayectoria aumentante con respecto a él.

#### Demostración.

Si existe una trayectoria aumentante  $\rightarrow X$  no es de máxima cardinalidad (teorema 2.1).

Supongamos que no existe tal trayectoria aumentante y sea  $M$  un acoplamiento de máxima cardinalidad. P.D:  $|X|=|M|$

Como  $X = (X \cap M) \cup (X - M)$  y  $M = (X \cap M) \cup (M - X)$  basta demostrar que  $|X - M| = |M - X|$ .

Sea  $G' = (N, A)$  con  $A = (X - M) \cup (M - X)$ . Consideremos primero un circuito en  $G'$ , por el teorema anterior tal circuito tiene longitud par y los arcos en el circuito están alternadamente en  $X - M$  y en  $M - X$ , también podemos decir que cada circuito en  $G'$  tiene el mismo número de arcos en  $X - M$  que en  $M - X$ . Ahora consideremos un componente de  $G'$  que sea una trayectoria, igualmente por el teorema anterior los arcos están alternadamente en  $X - M$  y en  $M - X$ . Además los vértices finales de esta trayectoria son expuestos con respecto a  $X$  o  $M$ . Supongamos que la trayectoria sea impar, entonces el vértice final de la trayectoria será expuesto en el mismo acoplamiento, pero esto implica que existe una trayectoria aumentante en alguno de estos dos acoplamientos!, por lo que cada componente de  $G'$  tiene un número igual de arcos en  $X - M$  y en  $M - X$ . Como los arcos de  $G'$  constituyen el conjunto  $(X - M) \cup (M - X)$ , tenemos  $|X - M| = |M - X|$  y por lo tanto  $|X| = |M|$ . ■

Apoyándonos en el hecho que podemos caracterizar a las gráficas bipartitas con el teorema 1.1 que nos dice que una gráfica es bipartita si y sólo si carece de ciclos de longitud impar, tenemos el siguiente resultado debido a Mendelsohn-Dulmage y que nos indica cómo obtener un nuevo acoplamiento en gráficas bipartitas que cumpla ciertas condiciones a partir de dos acoplamientos dados.

**Teorema 2.4**

Mendelsohn-Dulmage

Sean  $G=(S,T,A)$  una gráfica bipartita y  $X_1, X_2$  dos acoplamientos en  $G$ . Entonces existe un acoplamiento  $X \subseteq X_1 \cup X_2$  tal que  $X$  cubre todos los nodos de  $S$  cubiertos por  $X_1$  y todos los nodos de  $T$  cubiertos por  $X_2$ .

**Demostración**

Hagamos  $X_1 \oplus X_2 = (X_1 \cup X_2) - (X_1 \cap X_2)$ , como  $G$  es bipartita los ciclos posibles deben de tener longitud par, y deben cumplir que cualquier trayectoria en  $X_1 \oplus X_2$

- i) Comienza y termina con arcos de  $X_1$ .
- ii) Comienza y termina con arcos de  $X_2$ .
- iii) Comienza con un arco de  $X_1$  y termina con un arco de  $X_2$ .
- iv) Comienza con un arco de  $X_2$  y termina con un arco de  $X_1$ .

Por lo tanto los únicos tipos de trayectoria y ciclos que pueden aparecer al realizar  $X_1 \oplus X_2$  son los mostrados en la figura 2.4

En cualquiera de los casos que aparecen en la figura existe un acoplamiento  $X' \subseteq X_1 \cup X_2$  tal que  $X'$  cubre todos los nodos de  $S$  cubiertos por  $X_1 - X_2$  y todos los nodos de  $T$  cubiertos por  $X_2 - X_1$ , entonces  $X = X' \cup (X_1 \cap X_2)$  es el acoplamiento buscado.

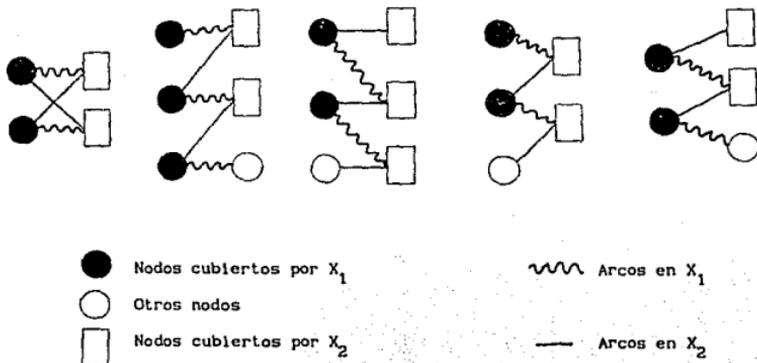


Fig 2.4

El problema del acoplamiento de cardinalidad máxima en gráficas bipartitas puede ser escrito como problema de programación lineal de dos maneras, la primera utilizando la matriz de incidencia de la gráfica y la segunda como un caso especial del problema del acoplamiento de peso máximo. Enseguida veremos el primer enfoque, ya que al plantear el problema dual surge de manera natural el problema de la cubierta de nodos mínima que será de utilidad para demostrar la equivalencia del teorema de flujo máximo-cortadura mínima. El segundo enfoque se analizará en general para el problema del acoplamiento de peso máximo en bigráficas.

Sea  $G=(S,T,A)$  una gráfica bipartita con  $|S| = m_1$ ,  $|T| = m_2$ ,  $|A| = n$ , numerando los arcos en  $A$  de 1 a  $n$ , los nodos en  $S$  de 1 a  $m_1$ , los nodos en  $T$  de  $m_1+1$  a  $m_1+m_2$ , la matriz de incidencia  $A$  de la gráfica bipartita  $G$  la podemos escribir de la forma siguiente:

$$A = \begin{bmatrix} A_s \\ A_t \end{bmatrix} \quad A = \begin{matrix} 1 & 2 & \dots & n \\ \vdots & \vdots & \vdots & \vdots \\ 1 & & & \\ \vdots & & & \\ m_1 & & & \\ m_1+1 & & & \\ \vdots & & & \\ m_1+m_2 & & & \end{matrix} \left[ \begin{array}{c} A_s \\ \hline A_t \end{array} \right]$$

donde  $A_s$  es una matriz de  $m_1 \times n$  y  $A_t$  es una matriz de  $m_2 \times n$ ,  $m_1$  y  $m_2$  son los renglones de  $A$ , en los nodos  $S$  y  $T$ , respectivamente. Cualquier columna de  $A_s$  o de  $A_t$  es un vector unitario.

Debido al siguiente teorema ampliamente conocido en el álgebra lineal (El teorema de Heller-Tompkins) que se enuncia sin demostración podemos concluir que la matriz de incidencia  $A$  de una gráfica bipartita es totalmente unimodular.

#### Teorema 2.5 Heller-Tompkins

Sea  $A$  una matriz de  $m \times n$  cuyos elementos son 0, -1 o 1 y tal que cada columna contiene a lo más dos elementos no nulos,  $A$  es totalmente unimodular (i.e. toda submatriz cuadrada tiene determinante 0, -1 o 1) siempre que  $A$  pueda ser particionada en dos conjuntos de renglones  $F_1$ ,  $F_2$  tales que cumplan:

- 1) Si los dos elementos diferentes de cero de una columna tienen igual signo, entonces uno de ellos está en el conjunto de renglones  $F_1$  y el otro en los de  $F_2$ .
- 11) Si los dos elementos diferentes de cero de una columna tienen signos contrarios, entonces ambos están en el conjunto de renglones  $F_1$  o ambos están en el conjunto  $F_2$ .

Para plantear el problema de programación lineal definimos las variables

$$X_i = \begin{cases} 1 & \text{Si el arco } i \text{ está en el acoplamiento} \\ 0 & \text{En otro caso} \end{cases} \quad i \in \overline{1, n}$$

El problema de programación lineal que resuelve el problema del acoplamiento de cardinalidad máxima para una gráfica bipartita  $G = (S, T, A)$  donde  $|S+T| = m$  y  $|A| = n$  es

$$\begin{aligned} \text{Max } Z &= X_1 + X_2 + \dots + X_n \\ \text{s.c.} & \end{aligned}$$

$$(P) \quad \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & & & a_{2n} \\ \vdots & & & \vdots \\ a_{m1} & \dots & & a_{mn} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{bmatrix} \leq \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

$$X_i \geq 0 \quad i \in \overline{1, n}$$

donde  $a_{ij}$  es el elemento en el renglón  $i$ , columna  $j$  de la matriz de incidencia de  $G$ . Reescribiendo  $P$  en forma estándar tenemos

$$\begin{aligned} \text{Max } Z &= X_1 + X_2 + \dots + X_n \\ \text{s.c.} & \end{aligned}$$

$$\left[ \begin{array}{c|c} A & I \end{array} \right] \begin{bmatrix} X_1 \\ \vdots \\ X_n \\ h_1 \\ \vdots \\ h_m \end{bmatrix} = \begin{bmatrix} 1 \\ \vdots \\ 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

$$X_i \geq 0 \quad i \in \overline{1, n} \quad h_j \geq 0 \quad j \in \overline{1, m}$$

donde  $I_{m \times m}$  es una matriz identidad que representa las  $m$  variables de holgura  $h_1$ . Utilizando el teorema de Heller-Tompkins mencionado anteriormente la matriz  $[A : I]$  del problema (P') es totalmente unimodular, esto nos garantiza que la base óptima  $B$ , que es una submatriz cuadrada de  $[A : I]$ , tenga determinante con valor  $\pm 1$  y por lo tanto la solución

$$X = B^{-1} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \text{ es entera.}$$

por tanto la variable  $X_j$  toma sólo los valores cero o uno y las  $m$  restricciones del problema (P) las podemos interpretar de la siguiente manera: Si asociamos al nodo  $j$ , para  $j \in \overline{1, m}$ , la  $j$ -ésima restricción

$$a_{j1} X_1 + a_{j2} X_2 + \dots + a_{jn} X_n \leq 1$$

nos dice que un nodo sólo puede tener un grado menor o igual a uno, con respecto al acoplamiento.

Al escribir el problema dual (D) del problema (P) tenemos

$$\begin{aligned} \text{Min } W &= Y_1 + Y_2 + \dots + Y_m \\ \text{s.c.} \end{aligned}$$

$$\begin{bmatrix} A^t \end{bmatrix} \begin{bmatrix} Y_1 \\ \vdots \\ Y_m \end{bmatrix} \geq \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$$

$$Y_j \geq 0 \quad j \in \overline{1, m}$$

donde  $A^t$  es la matriz traspuesta de la matriz de incidencia de  $G$ , que también es totalmente unimodular porque si  $B$  es una submatriz cuadrada de  $A^t$ , entonces  $B^t$  es una submatriz cuadrada de  $A$  y el  $\det(B) = \det(B^t) = 0, -1$  ó  $1$ . Por lo tanto la solución dual óptima  $Y$  es entera.

Si identificamos a las variables  $Y_j$  con  $j \in \overline{1, m}$  del problema dual (D) con los nodos de la gráfica  $G$ , la restricción

$$a_{11}^t Y_1 + a_{21}^t Y_2 + \dots + a_{m1}^t Y_m \geq 1$$

representa la condición de que en el nodo 1 al menos incida un arco de la gráfica  $G$ .

Esto nos lleva a que el problema dual (D) del problema del acoplamiento máximo es el de la cubierta de nodos de cardinalidad mínima en gráfica bipartita. Como la solución del problema primal (P) está acotada por  $Z = 0$  (Cuándo el acoplamiento es vacío) y por  $Z = \lfloor N/2 \rfloor$  (La parte entera del número de nodos  $N$  entre dos) entonces el problema primal (P) tiene óptimo finito y por el teorema fundamental de dualidad en programación lineal tenemos que  $\text{Max } Z = \text{Min } W$ .

Es decir, la cardinalidad del acoplamiento máximo es igual a la cardinalidad de la cubierta de nodos mínima en una gráfica bipartita.

Este teorema es uno de los más importantes resultados de la teoría de acoplamientos y es conocido como el teorema de König-Egerváry. Existen otras formas de demostrarlo, una de ellas es la sugerida por Ford y Fulkerson [13] que consiste en realizar la equivalencia con su teorema de dualidad flujo máximo-cortadura mínima. El que se presentará aquí, será expuesto en términos de cubiertas de nodos y es debido a Lovasz [18]. Enseguida recordaremos unas definiciones que nos serán de utilidad

A la cardinalidad del acoplamiento máximo de  $G$  se le llama el número de acoplamiento de  $G$  y es denotada  $\nu(G)$ .

Un conjunto de nodos  $C$  de  $G$  es una cubierta de nodos si cada arco de  $G$  tiene al menos un extremo en el conjunto  $C$ .

La cardinalidad de la cubierta de nodos mínima es denotada por  $\tau(G)$  y es conocida como la cardinalidad de la cubierta de nodos.

Para demostrar el teorema de König-Egerváry utilizaremos el siguiente lema:

#### Lema 2.2

Para cualquier gráfica  $\nu(G) \leq \tau(G)$

#### Demostración.

Sea  $C$  una cubierta de nodos en  $G$  y  $X$  un acoplamiento, entonces para cada arco  $e \in X$ , existe un vértice  $v_e$  en  $C$  que es incidente a  $e$ . Aún más si  $e, f \in X$ , entonces  $v_e \neq v_f$ . Entonces para cualquier acoplamiento  $X$  y cualquier cubierta de nodos en  $G$

$$|X| \leq |C| \quad \Rightarrow \quad \nu(G) \leq \tau(G). \quad (1) \quad \blacksquare$$

Teorema 2.6

## Teorema Max-Min de König-Egervály

Para cualquier gráfica bipartita, el máximo número de arcos en un acoplamiento es igual al mínimo número de nodos en una cubierta de nodos. Es decir  $\tau(G) = \nu(G)$

Demostración.

Para cualquier gráfica ya demostramos que  $\tau(G) \geq \nu(G)$ .

P.D.  $\tau(G) \leq \nu(G)$

De G quitemos arcos mientras  $\tau$  no cambie, la gráfica minimal obtenida será denotada  $G'$ . Entonces  $\tau(G') = \tau(G)$ , pero  $\tau(G'-e) < \tau(G) \forall$  arco  $e \in G'$ . Por lo tanto, dos arcos de  $G'$  son no adyacentes.

Supongamos que no, entonces existen 2 arcos  $x$  e  $y$  adyacentes a un nodo  $l$  en  $G'$ . Considere  $G'-x$ , por la minimalidad de  $G'$  existe una cubierta de nodos  $S_x$  cubriendo  $G'-x$  con  $|S_x| = \tau(G')-1$ , por supuesto ningún extremo de  $x$  está en  $S_x$ . Análogamente, existe un conjunto  $S_y$  cubriendo  $G'-y$  que no contiene ningún extremo de  $y$ , con  $|S_x| = |S_y|$ .

Hagamos la subgráfica inducida  $G''$  de  $G'$ ,  $G'' = (\{l\} \cup \{S_x \cup S_y\}, A'')$ .

Sea  $t = |S_x \cap S_y|$  entonces  $|N(G'')| = 2[\tau(G')-1-t] + 1$  y dado que  $G''$  es bipartita (por que es una subgráfica de  $G$ ) existe un conjunto  $T$  (El menor de las 2 clases de color de  $G''$ ) que cubre  $G''$  y  $|T| \leq \tau(G')-1-t$ .

Pero  $T' = T \cup (S_x \cap S_y)$  cubre  $G'$ , supongamos que  $z$  es algún arco en  $G'$ . Si  $z \neq x$  ó  $z \neq y$ , entonces  $z$  es cubierto por  $S_x$  y  $S_y$ , i.e. es cubierto por  $S_x \cap S_y$  o conecta  $S_x - S_y$  con  $S_y - S_x$ , en el segundo caso es un arco de  $G'$  y es cubierto por  $T$ . finalmente  $x, y$  son arcos de  $G'$  y por esto cubiertos por  $T$ . Así

$$\begin{aligned} \tau(G') &\leq |T'| = |T \cup (S_x \cap S_y)| = |T| + |S_x \cap S_y| \leq \\ &\leq \tau(G')-1-t+t = \tau(G')-1. \end{aligned}$$

Por lo tanto  $G'$  consiste en arcos no adyacentes como se quería demostrar.

$$\text{Y } \tau(G) = \tau(G') = \nu(G') \leq \nu(G) \quad (2)$$

finalmente combinando las desigualdades (1) y (2) tenemos que

$$\nu(G) = \tau(G). \quad \blacksquare$$

Para formular el problema del acoplamiento máximo en bigráficas como un caso especial del problema de programación lineal que resuelve el problema del acoplamiento de peso máximo en bigráficas, haremos lo siguiente:

Sea  $G=(S,T,A)$  una gráfica bipartita con  $|S| = m$   $|T| = n$ , si definimos las variables

$$X_{ij} = \begin{cases} 1 & \text{Si el arco } (i,j) \text{ está en el acoplamiento} \\ 0 & \text{En otro caso} \end{cases}$$

el problema de programación lineal que resuelve el problema del acoplamiento de peso máximo en bigráficas es

$$\begin{aligned} \text{Max } z &= \sum_i \sum_j W_{ij} X_{ij} \\ \text{s.c. } & \sum_j X_{ij} \leq 1 \quad (i=1,2,\dots,m) \\ & \sum_i X_{ij} \leq 1 \quad (j=1,2,\dots,n) \\ & X_{ij} \geq 0 \end{aligned} \quad (P) \qquad (R1)$$

en el cual cada variable  $X_{ij}$  es cero o uno y cada  $W_{ij}$  es 1.

Este problema puede ser visto como un problema de flujo máximo y utilizando el teorema de integralidad de flujos en redes, podemos garantizar que existe una solución óptima en la cual el flujo a través de cada arco es entero y en particular cero o uno, por lo tanto el problema anterior tiene una solución óptima finita entera. Así tenemos el traslado del teorema de integralidad a gráficas bipartitas y que se enuncia en el siguiente teorema.

**Teorema 2.7** De Integralidad para acoplamientos en bigráficas.

El poliedro de acoplamiento definido por las restricciones (R1), en el problema (P) tiene solamente vértices enteros.

### Demostración

Las restricciones del problema (P), corresponden a la matriz de incidencia de la gráfica bipartita que es totalmente unimodular por el teorema de Heller-Tompkins, por lo tanto la solución óptima al problema es entera. Aún más, los únicos valores que pueden tomar las variables  $x_{ij}$  son cero o uno. ■

Si  $m = n$ , entonces cualquier solución factible  $x = [x_{ij}]$  que satisface las restricciones con igualdad es una matriz doblemente estocástica, es decir una matriz no negativa en la cual la suma de las entradas en cada renglón y en cada columna es la unidad. Una solución factible con valores ceros y unos tiene la forma de una matriz de permutación, es decir una matriz con exactamente un 1 en cada renglón y en cada columna. Así se obtiene el siguiente corolario del teorema de integralidad: Toda matriz doblemente estocástica es una combinación convexa de matrices de permutación, conocido como teorema de Birkoff-Von Newman.

Para demostrar este teorema necesitaremos el siguiente resultado:

### Teorema 2.8

Para toda matriz doblemente estocástica  $X_{n \times n} = [x_{ij}]$ , existe una matriz de permutación  $X'_{n \times n} = [x'_{ij}]$  tal que  $x'_{ij} = 0$  siempre que  $x_{ij} = 0$ .

### Demostración.

Consideremos una red con nodos  $r_1, r_2, r_3, \dots, r_n$ ;  $s_1, s_2, s_3, \dots, s_n$  en la cual  $(r_i, s_j)$  es un arco  $\Leftrightarrow x_{ij} > 0$ . Asignando una unidad de disponibilidad a cada nodo  $r_i$  y una unidad de demanda a cada nodo  $s_j$ , olvidandonos de los coeficientes de costo, el problema resultante de transbordo tiene una solución factible, en efecto las entradas  $x_{ij}$  de la matriz  $X$  constituyen una. Por el teorema de integralidad, el problema tiene una solución  $x^*$  de valor entera. Los números  $x^*_{ij}$  son las entradas de la matriz  $X$  de permutación deseada. (Para obtenerla podemos también aplicar el algoritmo del acoplamiento máximo). ■

Teorema 2.9 de Birkoff-Von Newman

Toda matriz doblemente estocástica es una combinación convexa de matrices de permutación.

Demostración.

La siguiente es una demostración constructiva para expresar una matriz doblemente estocástica  $M$  como una combinación convexa de matrices de permutación  $M_1, M_2, M_3, \dots, M_k$ .

Paso 1 Hacer  $t_1 = 1$   $X = M$   $k = 1$

Paso 2 Se tienen matrices de permutación  $M_1, M_2, \dots, M_{k-1}$  ( $M_0 = [0]$ ). Una matriz  $X$  doblemente estocástica y números no negativos  $t_1, t_2, \dots, t_k$  tal que  $M = t_1 M_1 + t_2 M_2 + t_3 M_3 + \dots + t_{k-1} M_{k-1} + t_k X$  y  $t_1 + t_2 + \dots + t_k = 1$ . Si  $X$  es una matriz de permutación, entonces hacer  $M_k = X$  y detenerse.

De otra forma use el teorema anterior para encontrar una matriz de permutación  $X^*$  tal que  $x_{ij}^* = 0$  siempre que  $x_{ij} = 0$ .

Paso 3 Las  $n$  entradas  $x_{ij}$  de  $X$  para las cuales  $x_{ij}^* = 1$  son todas positivas; denote la menor de ellas por  $c$ . (Como  $X^*$  no es una matriz de permutación entonces  $c < 1$ ). Hacer  $t = t_k$ ;  $t_k = ct$ ;  $t_{k+1} = (1-c)t$  y  $M_k \rightarrow X^*$ .

Entonces reemplazar  $X$  por  $(1/c)(X - cM_k)$ , reemplazar  $k$  por  $k+1$  y regresar al paso 2.

En cada iteración de este algoritmo, el número de entradas cero en la matriz  $X$  se incrementa, por lo tanto el algoritmo debe terminar.

2.4 ALGORITMOS PARA LA OBTENCION DEL ACOPLAMIENTO MAXIMO EN GRAFICAS BIPARTITAS.

En esta sección veremos dos algoritmos para obtener la cardinalidad del acoplamiento máximo en una gráfica bipartita. El problema del acoplamiento máximo consiste en obtener un acoplamiento de una gráfica dada, tal que contenga el número máximo de arcos. El primero está basado en una especialización del algoritmo de Ford y Fulkerson para flujos en redes; el segundo en el teorema y la idea de las trayectorias aumentantes.

## 2.4.1 ESPECIALIZACION DEL ALGORITMO DE FLUJO MAXIMO PARA EL PROBLEMA DEL ACOPLAMIENTO MAXIMO EN GRAFICAS BIPARTITAS.

Ya vimos que todo problema del acoplamiento máximo en bigráficas es equivalente a uno de flujo máximo, y dada la estructura particular de este problema, el algoritmo de Ford y Fulkerson puede ser especializado para producir un algoritmo más eficiente para resolver este problema.

Enseguida veremos las características de las trayectorias aumentantes del problema de flujo máximo en la gráfica  $G(B)$  asociada al problema del acoplamiento máximo en una gráfica bipartita  $B$ .

Las trayectorias aumentantes de flujo de  $o$  a  $d$  comienzan utilizando un arco adyacente a  $o$  con flujo cero, utilizan alternadamente arcos de  $A$  con flujo cero, recorridos positivamente y arcos de  $A$  con flujo uno (es decir que están en el acoplamiento) recorridos negativamente, los dos últimos arcos de la trayectoria aumentante tienen flujo cero, uno pertenece a  $A$  y el otro es adyacente a  $d$ .

Si en estas trayectorias aumentantes de flujo eliminamos el primero y el último arcos, ver que corresponden a trayectorias aumentantes relativas al acoplamiento y viceversa es sencillo:

Sea  $P: o \rightarrow s_1 \rightarrow t_1 \leftarrow s_2 \rightarrow t_2 \leftarrow \dots \rightarrow t_j \rightarrow d$  con  $s_1 \in S$  y  $t_1 \in T$   $i \in \overline{1, j}$  una trayectoria aumentante de flujo, entonces

$$s_1 \rightarrow t_1 \leftarrow s_2 \rightarrow t_2 \leftarrow \dots \rightarrow t_j$$

es una trayectoria aumentante relativa al acoplamiento  $X$ , porque

$$f(s_i, t_i) = 0 \quad i = \overline{1, j}$$

$$f(s_{i+1}, t_i) = 1 \quad i = \overline{1, j-1}$$

por lo tanto  $(s_i, t_i) \notin X$  y  $(s_{i+1}, t_i) \in X$ .

Recíprocamente, si  $s_1 \rightarrow t_1 \leftarrow s_2 \rightarrow t_2 \leftarrow \dots \rightarrow t_j$  es una trayectoria aumentante relativa al acoplamiento  $X$  definido en la red bipartita  $G=(S, T, A)$  como

$$(s_i, t_i) \notin X \quad \text{entonces} \quad f(s_i, t_i) = 0 \text{ y}$$

$(s_{i+1}, t_i) \in X$  entonces  $f(s_{i+1}, t_i) = 1$

además, como  $(o, s_1)$  y  $(t_j, d)$  tienen flujo cero en la red de flujo asociada  $G(B)$ ,  $P: o \rightarrow s_1 \rightarrow t_1 \leftarrow s_2 \rightarrow t_2 \leftarrow \dots \rightarrow t_j \rightarrow d$  es una trayectoria aumentante de flujo. Ver la figura 2.5

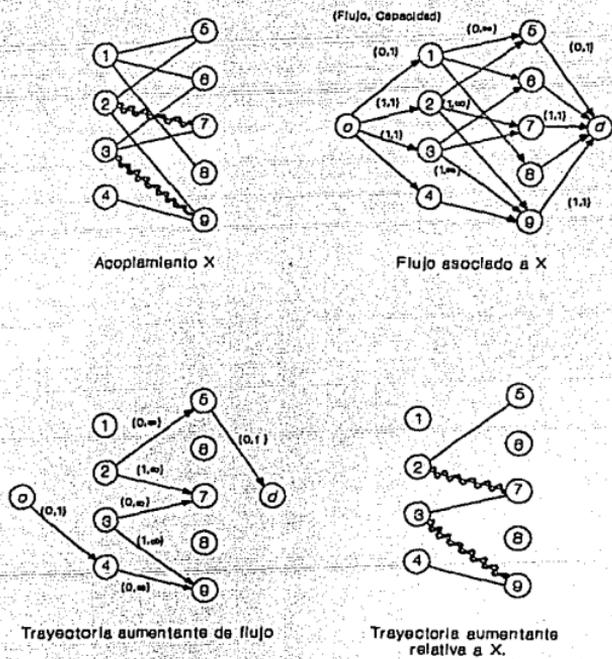


Figura 2.5

Notación : para el arco  $(s, t) \in X$  denotaremos  $t$  como  $X(s)$ .  
 Definimos entonces

$$\text{Dominio } X = \text{Dom } X = \{ s \in S \mid \exists t \in T \text{ con } (s, t) \in X \}$$

$$\text{Rango } X = \{ t \in T \mid \exists s \in S \text{ con } (s, t) \in X \}$$

En la primera figura del diagrama del ejemplo, tenemos:  $X(2)=7$  y  $X(3)=9$

y  $\text{Dom } X = \{ 2, 3 \}$

$\text{Rango } X = \{ 7, 9 \}$

Sea  $S_X = \{ s \in S \mid \exists t \in T - \{ \text{Rango } X \}, \text{ con } (s, t) \in A \}$ , (i.e.  $s \in S_X$  si tiene algún vecino expuesto); en el ejemplo  $S_X = \{ 1, 2, 3 \}$

Para determinar una trayectoria aumentante de flujo deberá construirse una secuencia

$o, X(s'), s', X(s''), s'', \dots$

que comience en  $\{ S \} - \{ \text{Dom } X \}$  y termine en algún nodo de  $S_X$  (nodo acoplado que tienen un nodo vecino expuesto), donde

$(s', X(s')), (s'', X(s'')), \dots \in A$

En el ejemplo tenemos:  $4, X(3), 3, X(2), 2$  donde:

$4 \in S - \{ \text{Dom } X \}$  y  $2 \in S_X$

esta secuencia define una trayectoria aumentante de flujo

$R: o \rightarrow 4 \rightarrow 9 \leftarrow 3 \rightarrow 7 \leftarrow 2 \rightarrow 5 \rightarrow d$

Las trayectorias que se detectan durante el algoritmo pueden expresarse en términos de los elementos de  $S$ . Es decir basta con usar la etiqueta  $F(s') = s$ , para denotar que  $s'$  se alcanza desde  $s$ .

Antes de describir el algoritmo veremos que el problema de la cubierta mínima equivale al problema de la cortadura mínima para la red de flujo asociada al problema del acoplamiento máximo definido en una bigráfica.

#### Lema 2.2

El problema de la cubierta mínima en una gráfica bipartita  $B$  es equivalente al problema de la cortadura mínima en la red de flujo asociada  $G(B)$ .

#### Demostración.

Sea  $G(B) = (N, A)$  la red de flujo asociada al problema de acoplamiento máximo definido en una bigráfica  $B$ .

Las cortaduras con capacidad finita están formados por arcos adyacentes a  $o$  y  $d$ , es decir son de la forma  $Q = (L, N-L)$  donde  $L \subset N$ ,

$o \in L$  y  $d \in L$  y no existe un arco  $(s,t) \in A$  con  $s \in L$  y  $t \in L$  (Porque estos arcos tienen capacidad infinita). Cada arco de  $Q$  adyacente a  $d$  corresponde a un nodo  $s \in S$  tal que  $s \in L$ ; igualmente, cada arco de  $Q$  adyacente a  $o$  corresponde a un nodo  $t \in T$  tal que  $t \in L$ . Como estos arcos tienen capacidad superior igual a uno.

$$\text{Cap}(Q) = |S-L| + |T \cap L|$$

por lo tanto estos cortes corresponden uno a uno a las cubiertas  $C$  donde la correspondencia puede expresarse  $C = (S-L) \cup (T \cap L)$  y  $|X| = |Q|$  ■

Enseguida se describe detalladamente el algoritmo

**ALGORITMO DEL ACOPLAMIENTO MAXIMO EN GRAFICAS BIPARTITAS BASADO EN UNA ESPECIALIZACION DEL ALGORITMO DE FORD Y FULKERSON PARA FLUJOS.**

- Paso 1** Iniciar con un acoplamiento  $X$  cualquiera, puede ser el vacío.
- Paso 2** Sea  $B = \text{Dom } X$  y  $W = S - \{ \text{Dom } X \}$
- Paso 3** Si  $W = \emptyset$ , terminar.  $X$  es un acoplamiento máximo y  $B$  una cubierta mínima.  
Si  $W \neq \emptyset$ , ir al paso 4.
- Paso 4** Sea  $k \in W$ .  
Si existe  $l \in \{ T - \{ \text{Rango } X \} \}$  con  $(k, l) \in A$  ir al paso 5.  
En otro caso ir a 5.
- Paso 5** Para todo  $s' \in B \cap S$  con  $(k, X(s')) \in A$ , actualizar  
 $B = B \cup \{ X(s') \} - \{s'\}$ ;  $W = W \cup \{s'\}$ ;  $F(s') = k$ .  
Sea  $W = W - \{k\}$ , ir al paso 3.
- Paso 6** Generar la secuencia  $k = k^0, k^1, \dots, k^r$  donde  $k^1 = F(k^{1-1})$ , para  $l = 1, \dots, r$  y  $k^r$  no está en el dominio de  $F$ .  
Actualizar:  
 $X = X - \{ (k^0, X(k^0)), \dots, (k^{r-1}, X(k^{r-1})) \} \cup \{ (k^0, l), (k^1, X(k^0)), \dots, (k^r, X(k^{r-1})) \}$   
Ir al paso 2.

Como ejemplo de su aplicación utilizaremos este algoritmo para determinar el acoplamiento máximo en la gráfica bipartita con acoplamiento inicial que se muestra en la figura 2.6 a)

### Iteración 1

Los conjuntos B y W son  $B = \{ 2, 4 \}$   $W = \{ 1, 3, 5 \}$

Como  $W \neq \emptyset$  con  $k = 3$  se cumple que existe el nodo  $7 \in \{ T\text{-Rango } X \}$  tal que el arco  $(3, 7) \in A$ , es decir, se detectó una trayectoria aumentante, el nuevo acoplamiento es  $X = \{ (2, 6), (4, 9), (3, 7) \}$  que se muestra en la figura 2.6 b)

### Iteración 2

Ahora tenemos que  $B = \{ 2, 3, 4 \}$   $W = \{ 1, 5 \}$

Como existe  $k = 1 \in W$  pero no existe ningún arco  $(1, l) \in A$  con  $l \in \{ T\text{-Rango } X \}$  se realizan las actualizaciones del paso 5

$B \cap S = B$ ; actualizando tenemos

$X(2) = 6$   $X(3) = 7$   $X(4) = 9$  como  $(1, 6) \in A$  y  $(1, 9) \in A$

$F(2) = 1$   $F(4) = 1$

$B = \{ 3, 6, 9 \}$   $W = \{ 2, 3, 4 \}$

Elegimos  $k = 2 \in W$  y como existe el arco  $(2, 8) \in A$  con el nodo  $8 \in \{ T\text{-Rango } X \} = \{ 8, 10, 11 \}$  hemos encontrado una trayectoria aumentante (que se muestra en la figura 2.6 b con líneas punteadas) con

$k = 2$   $F(2) = 1$  con  $1 \notin W$

Actualizando el acoplamiento tenemos

$X = X - \{ k^0, X(k^0) \} \cup \{ (k^0, l), (k^1, X(k^0)) \}$

$X = X - \{ (2, 6) \} \cup \{ (2, 8), (1, 6) \}$

por lo tanto  $X = \{ (4, 9), (3, 7), (2, 8), (1, 6) \}$  el cual se muestra en la figura 2.6 c)

### Iteración 3

Ahora  $B = \{ 1, 2, 3, 4 \}$   $W = \{ 5 \}$

como  $W \neq \emptyset$  con  $k = 5$  no existe  $l \in \{ T\text{-Rango } X \} = \{ 10, 11 \}$

Tenemos que  $B \cap S = B$

$X(1) = 6$   $X(2) = 8$   $X(3) = 7$   $X(4) = 9$  y como  $(5, X(1)) \in A$  y  $(5, X(4)) \in A$

$F(1) = 5$   $F(4) = 5$   $B = \{ 2, 3, 6, 9 \}$   $W = \{ 1, 4 \}$

se elige  $k = 4 \in W$ , como existe  $(4, 11)$  con  $11 \in \{ T\text{-Rango } X \} = \{ 10, 11 \}$  hemos detectado una trayectoria aumentante (ver figura 2.6 c) con

$k = 4$   $F(4) = 5$  con  $5 \in W$

Actualizando tenemos

$X = X - \{ (4, 9) \} \cup \{ (4, 11), (5, 9) \}$

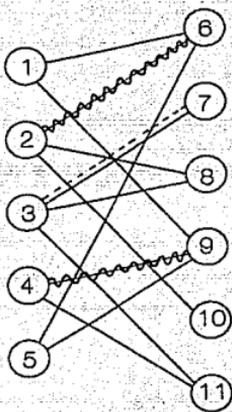
$X = \{ (1, 6), (2, 8), (3, 7), (4, 11), (5, 9) \}$

que se muestra en la figura 2.6 d)

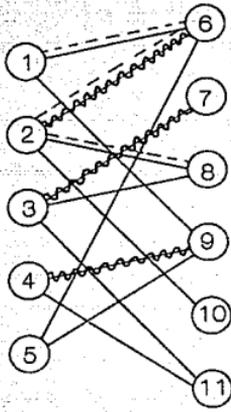
### Iteración 4

$B = \{ 1, 2, 3, 4, 5 \}$   $W = \emptyset$

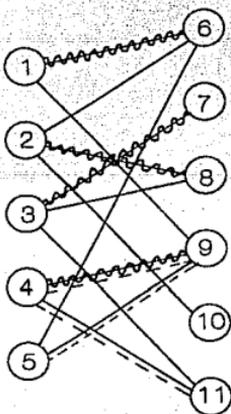
Como  $W = \emptyset$  el acoplamiento  $X = \{ (1, 6), (2, 8), (3, 7), (4, 11), (5, 9) \}$  es de cardinalidad máxima,  $B$  es una cubierta mínima y se muestran en la figura 2.6 d)



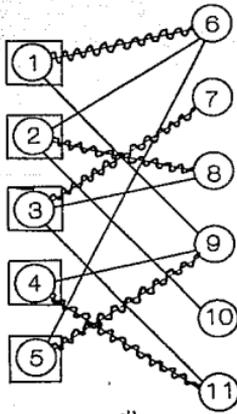
a)



b)



c)



d)

Figura 2.6

El segundo enfoque está basado en la idea de encontrar una trayectoria aumentante con respecto a un acoplamiento, para obtener así un nuevo acoplamiento con cardinalidad mayor en uno al acoplamiento original. Si no podemos encontrar una trayectoria aumentante entonces el acoplamiento es de cardinalidad máxima. Veamos la siguiente definición que nos ayudará en el tratamiento del algoritmo:

Sea  $G=(S,T,A)$  una gráfica bipartita y  $X$  un acoplamiento tal que  $X \subseteq A$ .

Un árbol alternante relativo al acoplamiento  $X$  es un árbol que satisface las siguientes dos condiciones. Primero el árbol contiene exactamente un nodo expuesto de  $S$ , que le llamaremos raíz. Segundo todas las trayectorias entre la raíz y cualquier otro nodo en el árbol son trayectorias alternantes.

El algoritmo comienza con cualquier acoplamiento factible, posiblemente el acoplamiento vacío. Cada nodo expuesto en  $S$  es tomado como la raíz de un árbol alternante, nodos y arcos son agregados a los árboles por medio de una regla de etiquetación. En algún momento tiene que ocurrir alguno de los dos siguientes eventos:

i) Algún nodo expuesto en  $T$  es alcanzado desde alguno de los árboles, por lo tanto se ha detectado una trayectoria aumentante y se puede construir un nuevo acoplamiento de cardinalidad mayor en uno al acoplamiento anterior, el procedimiento de la búsqueda de una trayectoria aumentante se repite con el nuevo acoplamiento.

ii) No es posible agregar más nodos y arcos a ninguno de los árboles; en este caso la etiquetación es húngara.

Los árboles resultantes que poseen la propiedad de que todas sus trayectorias alternantes maximales son de cardinalidad par son llamados árboles húngaros y pueden ser usados para construir una solución óptima dual que consiste en la unión de todos los nodos fuera del árbol en  $S$  y de todos los nodos dentro del árbol en  $T$ .

El algoritmo se describe enseguida.

*Algoritmo del acoplamiento de máxima cardinalidad para gráficas bipartitas.*

**Paso 0**

Sea  $G=(S,T,A)$  una gráfica bipartita. Sea  $X$  cualquier acoplamiento, posiblemente el acoplamiento vacío. Ningún nodo está etiquetado.

**Paso 1 Etiquetación**

1.0 Dar la etiqueta "0" a cada nodo expuesto en  $S$ .

1.1 Si no hay etiquetas sin examinar, ir al paso 3. De otra manera encontrar un nodo  $i$  con una etiqueta sin examinar. Si  $i \in S$ , entonces ir al paso 1.2; Si  $i \in T$  ir al paso 1.3.

1.2 Examinar la etiqueta en el nodo  $i$  ( $i \in S$ ) como sigue. Para cada arco  $(i,j) \in X$  incidente al nodo  $i$ , dar al nodo  $j$  la etiqueta "1", a menos de que el nodo  $j$  esté ya etiquetado.

Regresar al paso 1.1 .

1.3 Examinar la etiqueta en el nodo  $i$  ( $i \in T$ ) como sigue. Si el nodo  $i$  es expuesto, ir al paso 2. De otra manera identificar el único arco  $(i,j) \in X$  incidente al nodo  $i$  y dar al nodo  $j$  la etiqueta "1".

Regresar al paso 1.1 .

**Paso 2 Aumentación**

Se ha detectado una trayectoria aumentante que termina en el nodo  $i$  (identificado en el paso 1.3). Los nodos que preceden al nodo  $i$  en la trayectoria son identificados hacia atrás. Esto es, si la etiqueta en el nodo  $i$  es "j", el penúltimo nodo al final de la trayectoria es el nodo  $j$ . Si la etiqueta en el nodo  $j$  es "k", el antepenúltimo nodo de la trayectoria es el nodo  $k$  y así sucesivamente. El nodo inicial en la trayectoria fue etiquetado con "0". Aumentar  $X$  intercambiando los arcos de la trayectoria que no estén en el acoplamiento por lo que si lo estén. Remover todas las etiquetas de los nodos.

Regresar al paso 1.0 .

### Paso 3 Etiketación Húngara

La etiquetación es húngara, no existen trayectorias aumentantes y el acoplamiento  $X$  es de cardinalidad máxima. Sea  $L \subseteq S \cup T$  que denota el conjunto de nodos etiquetados. Entonces  $C = (S-L) \cup (T \cap L)$  es una cubierta de nodos de cardinalidad mínima, dual a  $X$ . Alto.

Aplicaremos este algoritmo para determinar el acoplamiento máximo en la bigráfica con acoplamiento inicial mostrado en la figura 2.7 a)

#### Iteración 1

Asignamos la etiqueta al único nodo expuesto en  $S$  [ $2$ ] = 0, al revisar esta etiqueta asignamos a los nodos en  $T$ , adyacentes a 2 las siguientes etiquetas [ $7$ ] = 2 y [ $8$ ] = 2

Al revisar la etiqueta de 7, el nodo 4 recibe la etiqueta [ $4$ ] = 7 y al revisar la etiqueta de 8, el nodo 3 recibe la etiqueta [ $3$ ] = 8. Al examinar la etiqueta de 3 tenemos que [ $5$ ] = 3, cuando se revisa la etiqueta del nodo 5 notamos que es un nodo expuesto en  $T$ , por lo tanto se ha detectado una trayectoria aumentante. La trayectoria es 5,3,8,2 que se ilustra en la figura 2.7 b). Modificamos el acoplamiento

$X = \{ (3,8), (1,6), (4,7) \} \cup \{ (2,8), (3,5) \} - \{ (3,8) \}$  entonces

$X = \{ (1,6), (4,7), (2,8), (3,5) \}$  y como ya no existen nodos expuestos en  $S$  la solución es óptima, la cubierta es  $C = (S - L) \cup (T \cap L)$  pero  $L = \emptyset$  por lo tanto  $C = S$ .

El acoplamiento y la cubierta óptimos se muestran en la figura 2.7 c)

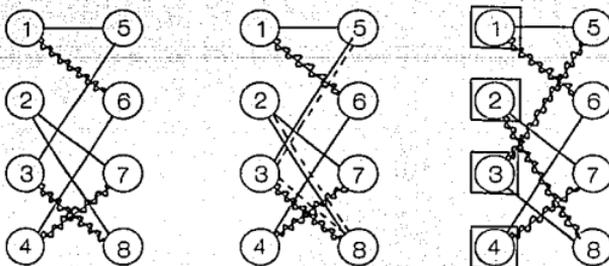


Figura 2.7

## 2.5 APLICACIONES DEL ACOPLAMIENTO DE CARDINALIDAD MÁXIMA

En esta sección se presentan dos problemas de tipo combinatorio que pueden ser resueltos con el algoritmo del acoplamiento de cardinalidad máxima en gráficas bipartitas.

### SISTEMAS DE REPRESENTANTES DISTINTOS

Existen dos versiones básicas del teorema de Hall y de las cuales se desprende un gran número de versiones que pueden ser reducidas a estas dos versiones básicas; la primera versión la enunció Hall en un contexto teórico de conjuntos, veamos: Sea  $S$  un conjunto finito y  $\{S_1, S_2, \dots, S_n\}$  subconjuntos, no necesariamente distintos de  $S$ . ¿Cuándo es posible encontrar un conjunto de  $n$  elementos distintos  $s_1, s_2, \dots, s_n$  con  $s_i \in S_i$ ? Tal conjunto de elementos es llamado SISTEMA DE REPRESENTANTES DISTINTOS (SDR). Este conjunto finito  $S$  junto con una colección de sus subconjuntos se llama hipergráfica.

Veamos con un ejemplo que, resolviendo un problema de acoplamiento máximo en gráficas bipartitas solucionamos también el problema de representantes distintos.

Un despacho de 6 asesores atiende a 5 empresas distintas. Si enumeramos a los asesores del 1 al 6, la forma en que están asignados a las empresas es la siguiente

Empresa 1 : {1,2,3}  
Empresa 2 : {1,4}  
Empresa 3 : {4,5,6}  
Empresa 4 : {4}  
Empresa 5 : {1,4}

Para la reunión anual de evaluación del despacho, se desea formar un grupo de 5 asesores, representando cada uno de ellos a una empresa diferente. ¿Cómo se puede integrar el grupo deseado?

Modelando esta situación con una gráfica bipartita  $G = (S, T, A)$  donde  $S = \{\text{Empresas}\}$  y  $T = \{\text{Asesores}\}$ , un arco  $(i, j) \in A$  con  $i \in S$  y  $j \in T$  si la empresa "i" tiene asignado al asesor "j", el problema se

puede resolver encontrando un acoplamiento máximo en ella, (ver figura 2.8) . Si el acoplamiento máximo contiene menos de 5 arcos, el problema de los representantes no tiene solución y entonces alguno de los asesores deberá representar a más de una empresa en la reunión de evaluación.

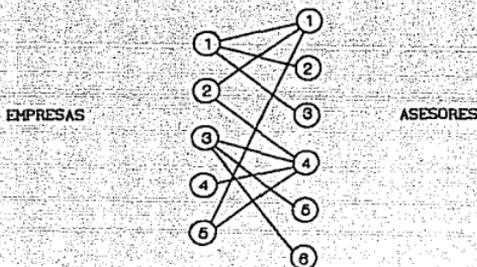


Figura 2.8

Hall no demostró su versión en términos de acoplamientos ni de teoría de gráficas, sin embargo la segunda versión básica es un teorema muy importante dentro de la teoría del acoplamiento en gráficas bipartitas, además de que lo podemos utilizar en el teorema de SDR; por esta razón veremos primero la demostración de la versión en acoplamientos, después veremos el teorema de Hall en combinatoria y por último veremos que estos dos teoremas son equivalentes.

Diremos que un conjunto de nodos  $N$  puede ser acoplado a otro conjunto de nodos  $N'$ , si existe un acoplamiento que no deje nodos expuestos en  $N$ . Así el teorema de Hall nos dice lo siguiente:

Teorema 2.10 De Hall

Sea  $G=(S,T,A)$  una gráfica bipartita,  $G$  posee un acoplamiento de  $S$  a  $T$   $\Leftrightarrow |\Gamma(S')| \geq |S'| \forall S' \subseteq S$

Demostración.

Supongamos que  $S$  puede ser acoplado a un subconjunto de  $T$ . Entonces, dado que cada vértice de  $S$  es acoplado con un vértice distinto de  $T$ , es claro que  $|\Gamma(S')| \geq |S'| \forall S' \subseteq S$ .

Por el contrario supongamos que  $G$  es bipartita y que  $S$  no puede ser acoplado a un subconjunto de  $T$ . Sea  $X$  un acoplamiento máximo en  $G$ . Por suposición, los arcos de  $X$  no son incidentes a todos los vértices de  $S$ . Sea  $u$  un vértice no acoplado por  $X$  y sea  $V$  el conjunto de todos los vértices de  $G$  conectados a  $u$  por una trayectoria alternante de  $X$ . Dado que  $X$  es un acoplamiento máximo entonces ( por el teorema de trayectoria aumentante, acoplamiento máximo )  $u$  es el único vértice no acoplado de  $V$ . Sea  $S' = V \cap S$  y  $T' = V \cap T$ . Claramente, los vértices de  $S' - \{u\}$  son acoplados con vértices de  $T'$ , aun más  $|T'| = |S'| - 1$  y  $T' \subseteq \Gamma(S')$ , en efecto  $T' = \Gamma(S')$  dado que todo vértice en  $\Gamma(S')$  es conectado a  $u$  por una trayectoria alternante. Pero entonces  $|\Gamma(S')| = |S'| - 1 < |S'|$  ■

**Teorema 2.11** De Hall en combinatoria.

Una colección de conjuntos  $\{S_1, \dots, S_n\}$  tiene un SDR si y sólo si para todo  $k$ ,  $0 \leq k \leq n$ , la unión de cualquiera  $k$  de los conjuntos  $S_1, \dots, S_n$  tiene cardinalidad al menos  $k$ .

**Demostración.**

La implicación de izquierda a derecha es trivial.

Supongamos al contrario que para todo  $k$ ,  $0 \leq k \leq n$  y para todo subconjunto  $\{i_1, \dots, i_k\}$  de  $\{1, \dots, n\}$ , tenemos  $|\bigcup_{j=1}^k S_{i_j}| \geq k$ . Constrúyase una bigráfica  $G=(S, T, A)$  como sigue: Los nodos de  $S$  son los conjuntos  $S_1, \dots, S_n$  y los nodos de  $T$  son los elementos de  $\bigcup_{j=1}^n S_j$ . Una el nodo  $S_i$  de  $S$  al nodo  $s_j$  de  $T$  si y sólo si  $s_j \in S_i$ . Escógase cualquier  $S' \subseteq S$ . Por definición de  $G$  tenemos que  $|\Gamma(S')| = \left| \bigcup_{S_i \in S'} S_i \right| \geq |S'|$  pero por el teorema 2.10,  $S$  puede ser acoplado en  $T$  y se sigue que los nodos finales en  $T$  de este acoplamiento deben ser un SDR. ■

A continuación se prueba que el teorema 2.11 implica el teorema 2.10, utilizando la formulación de una red con flujo de la siguiente manera:

**Demostración.**

Definamos la red bipartita  $G=(S, T, A)$  con  $S = \{x_1, \dots, x_m\}$ ,  $T = \{y_1, \dots, y_n\}$ ,  $A = \{(x_i, y_j) \mid s_i \in S_j\}$ . Asociamos una demanda  $b(y_j) = 1$  a cada nodo de  $T$  y una oferta  $a(x_i) = 1$  a cada nodo de  $S$ . Las capacidades de todos los arcos son infinitas. Ver la figura 2.9:

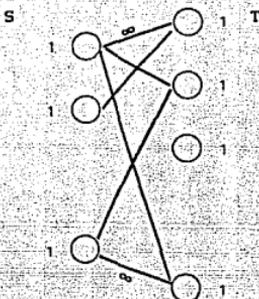


Figura 2.9

Es claro que un flujo factible entero de  $S$  a  $T$  escoge un SDR para  $S_1, \dots, S_n$  y viceversa. (Como solamente requerimos que  $f(S, y_i) \geq 1$  el mismo conjunto puede ser representado más de una vez).

Si las condiciones de Hall se satisfacen, entonces para cualquier subconjunto  $T' \subseteq T$ , existen al menos  $|T'|$  nodos de  $S$  que están unidos a nodos de  $T'$ , y el flujo  $f_{T'}$  existe y es factible, por lo tanto  $\{S_1, \dots, S_n\}$  admite un SDR. ■

Como ya dijimos existen muchas variaciones y generalizaciones del teorema de Hall, muchas de las que pueden ser probadas por reducción a una de las versiones básicas. Presentaremos como corolario la siguiente versión del teorema de Hall, referente a acoplamientos en bigráficas:

Teorema 2.12 (del matrimonio)

Una gráfica bipartita  $G=(S,T,A)$  admite un acoplamiento completo o perfecto  $\bullet$   $|S|=|T|$  y  $\forall S' \subseteq S$   $|S'| \leq |\Gamma(S')|$ .

Se llama así por la siguiente interpretación. Dada una lista de  $n$  hombres y un conjunto de  $n$  mujeres, cada hombre (mujer) elabora una lista con los nombres de las mujeres (hombres) con las que se quisiera casar. Entonces cada hombre (mujer) puede ser casado con una mujer (hombre) en su lista, si y sólo si, para todo valor de  $k$  ( $1 \leq k \leq n$ ) la unión de cualquiera  $k$  listas contienen al menos  $k$  nombres.

## DESCOMPOSICION EN CADENAS.

Uno de los resultados esenciales en la teoría de conjuntos parcialmente ordenados es debido a Dilworth (1950). En esta sección veremos que tal Teorema es equivalente al teorema de König-Egerváry.

Sea  $P = \{a_1, \dots, a_n\}$  cualquier conjunto finito. Cualquier relación  $R \subseteq P \times P$  se llama orden parcial de  $P$  si  $R$  es reflexiva, antisimétrica y transitiva. Un conjunto  $P$  que posee tal relación se llama conjunto parcialmente ordenado. Denotamos  $R$  por " $\leq$ "; esto es  $(a_i, a_j) \in R$  si y sólo si  $a_i \leq a_j$ . Escribimos también  $a_i < a_j$  si y sólo si  $a_i \leq a_j$  y  $a_i \neq a_j$ . Un subconjunto  $\{a_{i_1}, a_{i_2}, \dots, a_{i_k}\}$  de  $P$  es totalmente ordenado o una cadena si existe una permutación  $\Pi$  de los subíndices  $i_1, i_2, \dots, i_k$  tal que  $a_{\Pi(i_1)} \leq a_{\Pi(i_2)} \leq \dots \leq a_{\Pi(i_k)}$ . Cualesquiera dos elementos  $a_i$  y  $a_j$  son llamados comparables si pertenecen a alguna cadena y son incomparables en otro caso. Cualquier conjunto de elementos incomparables se llama anticadena.

Una colección  $\pi = \{C_1, \dots, C_r\}$  de cadenas (anticadenas) en  $P$  se llama partición en cadenas (anticadenas) de  $P$ , si los  $C_i$  son mutuamente ajenos y su unión es  $P$ . Observemos que cualquier conjunto parcialmente ordenado puede ser representado como la unión de cadenas ajenas; un ejemplo trivial es la partición formada por las cadenas con un elemento  $\{a_1\}, \{a_2\}, \dots, \{a_n\}$ . Así tiene sentido hablar de particiones en cadenas de  $P$  de cardinalidad mínima.

Por otro lado, dado que la cardinalidad de una anticadena no puede exceder  $|P|$ , debemos hablar de anticadenas de cardinalidad máxima.

Uno de los resultados fundamentales de la teoría de conjuntos parcialmente ordenados relaciona estos dos conceptos y es debido a Dilworth (1950).

### Teorema 2.13 De Dilworth

En cualquier conjunto finito parcialmente ordenado la cardinalidad de la anticadena máxima es igual a la cardinalidad de cualquier partición mínima en cadenas.

En la figura 2.10 ilustramos el teorema de Dilworth, con 11 nodos. En el ejemplo podemos ver que la cardinalidad de la anticadena más grande es igual a la cardinalidad de la menor partición en cadenas y son iguales a 4. Si  $i$  está situado más abajo que  $j$  y existe en la figura el arco  $(i, j)$ , entonces  $i < j$ .

El conjunto  $\{a_5, a_7, a_8, a_9\}$  en cajas es una anticadena de cardinalidad máxima y las rutas en negro corresponden a las cuatro cadenas ajenas  $C_1 = \{a_1, a_4, a_7, a_{10}\}$ ,  $C_2 = \{a_2, a_5\}$ ,  $C_3 = \{a_6, a_{11}\}$  y  $C_4 = \{a_3, a_8, a_7\}$ .

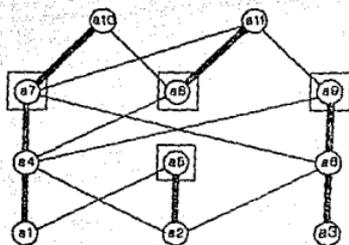


Figura 2.10

Como un ejemplo de Descomposición en Cadenas veamos el siguiente problema:

Un periódico desea cubrir todas las conferencias de una importante reunión internacional a celebrarse en determinada fecha.

Para ese día se ofrecen 10 conferencias con la duración y horario que se muestra a continuación en la tabla

Conferencia	1	2	3	4	5	6	7	8	9	10
Inicio	9:00	8:00	8:00	9:00	9:30	10:30	8:30	8:30	9:00	8:30
Duración en minutos	60	60	90	90	90	60	60	90	90	60

¿Cuál es el número mínimo de reporteros que debe enviar el periódico para cubrir el total de las conferencias ?

Las conferencias se pueden ordenar de manera que  $a < b$  signifique que la conferencia "a" termina antes de que la conferencia "b" inicie. Podemos modelar en una gráfica el orden parcial estricto existente entre las conferencias, donde cada nodo representa una conferencia, para tener la condición  $a < b$  tiene que existir el arco  $(a,b)$  y  $b$  debe estar situado en un nivel más abajo que  $a$ . Ver la figura 2.11

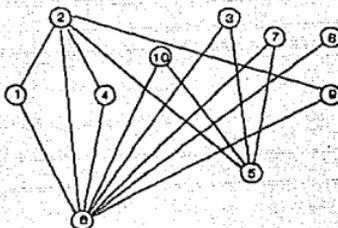


Figura 2.11

Cada cadena en la gráfica anterior representa una secuencia de conferencias que no se traslapan y que pueden ser cubiertas por un reportero.

Para reducir el problema a uno de acoplamiento máximo sea  $S = \{ 1, \dots, 10 \}$  el número de conferencia.  $G = (S, T, A)$  una bigráfica con  $T = S$  e  $(i, j) \in A$  si la conferencia  $i <$  conferencia  $j$ . La gráfica para nuestro ejemplo es la que se muestra en la figura 2.12

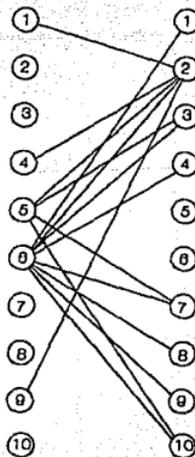


Figura 2.12

Encontrar un acoplamiento máximo en esta figura es equivalente a encontrar un subconjunto de pares de elementos de  $S$  que son comparables de modo que ningún elemento aparece más de una vez como primer elemento ni más de una vez como segundo elemento de un par. Identificando los elementos comunes en los pares encontrados, los "unimos" formando cadenas ajenas.

Por ejemplo si en un acoplamiento máximo para la figura 2.12 aparecen los arcos  $(6,4), (4,2)$  se puede construir la cadena  $6 < 4 < 2$ . Los elementos que no sean cubiertos por los arcos en el acoplamiento máximo se consideran como cadenas unitarias y así se tiene la descomposición en cadenas de  $S$  que se buscaba.

Para demostrar la equivalencia del teorema de Dilworth con el teorema de König, seguiremos el tratamiento de Fulkerson [13], y que se basa en la construcción adecuada de una gráfica bipartita, dado un conjunto parcialmente ordenado. Para probar el teorema de Dilworth, supongamos que el teorema de König-Egerváry es válido.

Sea  $P = \{x_1, \dots, x_n\}$  un conjunto parcialmente ordenado. Construyamos una gráfica bipartita  $G = (S, T, A)$  donde

$$S = \{a_1, a_2, \dots, a_n\} \quad T = \{b_1, b_2, \dots, b_n\}$$

y  $(a_i, b_j) \in A$  si y sólo si  $x_i < x_j$ .

Necesitaremos además los siguientes dos lemas.

#### Lema 2.14

Sea  $M$  un conjunto de arcos independientes en  $G$  (i.e. un acoplamiento) entonces existe una partición  $P$  en cadenas de  $P$  tal que  $|M| + |P| = n$

#### Demostración.

Supongamos que  $M = \{(a_{1_1}, b_{1_2}), (a_{1_3}, b_{1_4}), \dots, (a_{1_{2k-1}}, b_{1_{2k}})\}$ . Así  $x_{1_1} < x_{1_2}, x_{1_3} < x_{1_4}, \dots, x_{1_{2k-1}} < x_{1_{2k}}$  y de entre los diferentes elementos de estos  $2k$   $x_j$ , tenemos un conjunto  $P'$  de cadenas donde

cada una de ellas tiene longitud de al menos dos y son ajenas por pares dado que  $M$  es un acoplamiento. Se puede extender  $p'$  a una partición en cadenas  $p$  de  $P$  de manera natural, lográndose esto agregando todas las cadenas unitarias que no estén presentes en  $p'$ . Sea  $n_j =$  El número de elementos en la  $j$ -ésima cadena de  $p$ . Entonces tenemos que

$$n = \sum_{j=1}^{|P|} n_j = \sum_{j=1}^{|P|} (n_j - 1) + |P| = |M| + |P|$$

dado que hay precisamente  $n_j - 1$  arcos en la  $j$ -ésima cadena de  $p$ . ■

### Lema 2.15

Si  $C_G \subseteq SUT$  es una cubierta mínima de una bigráfica  $G=(S,T,A)$  de un conjunto parcialmente ordenado  $P$ , entonces existe una anticadena  $U$  contenida en  $P$  con  $|C_G| + |U| \geq n$

### Demostración

Sea  $C_G = \{a_{1_1}, a_{1_2}, \dots, a_{1_k}, b_{j_1}, b_{j_2}, \dots, b_{j_m}\}$  una cubierta mínima de  $G$ .  $C_G$  corresponde a un subconjunto  $Q$  de  $P$  con  $|Q| \leq |C_G|$ . Sea  $U = P - Q$ . Entonces los elementos de  $U$  son no relacionados por pares en  $P$  dado que  $C_G$  es una cubierta en  $G$ . Aun más por construcción  $|C_G| + |U| \geq |Q| + |U| = |Q| + |P - Q| = |P| = n$ . ■

Ahora estamos en condiciones de demostrar el teorema de conjuntos parcialmente ordenados debido a Dilworth.

### Demostración Del teorema 2.13 de Dilworth.

Sea  $P$  cualquier conjunto parcialmente ordenado y  $G=(S,T,A)$  la bigráfica correspondiente a  $P$ . Sea  $L_G$  un conjunto máximo de arcos independientes en  $G$  y  $C_G \subseteq SUT$  una cubierta mínima de  $G$ , utilizando el primer lema existe una partición en cadenas  $p$  de  $P$  tal que  $|L_G| + |p| = n$ . Por el segundo lema existe una anticadena  $U \subseteq P$  con  $|C_G| + |U| \geq n$ . Pero por el teorema Max-Min de König-Egerváry, tenemos  $|L_G| = |C_G|$  y por lo tanto  $|p| \leq |U|$ .

Por otra parte, es una observación elemental notar que para cualquier partición arbitraria en cadenas  $p'$  de  $P$  y cualquier anticadena arbitraria  $U'$  de  $P$ , debemos tener  $|U'| \leq |p'|$  dado que dos elementos de  $U'$  no pueden existir en la misma cadena de  $p'$ . Así  $|p| = |U|$  y de aquí tenemos que  $p$  es una partición en cadenas mínima de  $P$  y  $U$  una anticadena máxima. ■

Notemos que esta prueba del teorema de Dilworth también brinda un algoritmo polinomial para encontrar una partición en cadenas mínima y una anticadena máxima en un conjunto parcialmente ordenado.

Para demostrar el teorema de König-Egerváry suponiendo que se cumple el teorema de Dilworth, se hace lo siguiente:

Construimos la gráfica bipartita  $G=(S,T,A)$  de un conjunto parcialmente ordenado, donde la relación  $x > y$  define arcos  $(x,y) \in A$ , correspondientes a  $x \in S$ ,  $y \in T$ . La implicación deseada se sigue de los dos resultados dados a continuación

i) Correspondiente a cualquier descomposición  $p$  de  $P$  existe un conjunto de arcos independientes  $M$  de  $G$  con  $|p|+|M|=|S \cup T|$ .

ii) Correspondiente a cualquier conjunto  $U \subseteq P$  de elementos incomparables, existe una cubierta  $C$  de  $G$  con  $|U|+|C|=|S \cup T|$ .

## 2.6 ACOPLAMIENTO MAX-MIN

En esta sección trataremos el problema del acoplamiento max-min en gráficas bipartitas, también conocido como problema del cuello de botella. El problema consiste en encontrar en una gráfica ponderada un acoplamiento de máxima cardinalidad tal que el peso mínimo de los arcos que la formen sea máximo, este problema utiliza la siguiente filosofía: "La fuerza de una cadena está dada por su eslabón mas débil".

Un procedimiento que obtiene acoplamientos max-min en gráficas bipartitas de cualquier cardinalidad, incluyendo la máxima, es el siguiente.

Iniciemos con el acoplamiento vacío y un conveniente "valor de umbral"  $W$ . En el paso general del algoritmo, un acoplamiento max-min de cardinalidad  $k$  ha sido obtenido. Entonces tratamos de encontrar una trayectoria aumentante en la subgráfica conteniendo todos los arcos  $(i, j)$  para los cuales  $w_{ij} \geq W$ . Si el aumento es posible, encontramos un acoplamiento max-min de cardinalidad  $k+1$ . Si el aumento no es posible, el valor de entrada de  $W$  se reduce en lo necesario para permitir que el aumento ocurra.

Sin embargo, es posible aplicar un algoritmo mejor, que se debe a Lawler [17] y se basa en asociar un número  $\pi_j$  a cada nodo  $j$  en  $T$ . Este número indica el nivel al cual el valor de umbral debe ser reducido, para que  $j$  pueda ser agregado al árbol alternante. En otras palabras  $\pi_j$  toma el valor del máximo  $w_{ij}$ , tal que  $(i, j)$  es un arco y el nodo  $i$  está en un árbol alternante. Los nodos están etiquetados completamente, pero ningún nodo etiquetado  $j$  en  $T$  se revisa, a menos que  $\pi_j \geq W$ . Cuando no hay más nodos elegibles para revisar,  $W$  es reducido al máximo valor de  $\pi_j$  estrictamente menor que  $W$ , esto permite que al menos un nodo adicional se agregue al árbol. Debe ocurrir que el aumento sea posible o que la etiquetación sea húngara.

Así enseguida se describe el algoritmo que obtiene acoplamientos max-min en gráficas bipartitas.

*Algoritmo del valor de entrada para acoplamientos MAX-MIN.*

**Paso 0** Inicio

Sea  $G=(S,T,A)$  una gráfica bipartita y  $w_{ij}$  el peso de cada arco  $(i,j) \in A$ . Hacer  $X = \emptyset$ ,  $W = +\infty$ , y  $\pi_j = -\infty$  para cada nodo  $j \in T$ . Ningún nodo está etiquetado.

**Paso 1** Etiquetación

1.0 Dar la etiqueta "0" a cada nodo expuesto en S.

1.1 Si no hay etiquetas no revisadas, ir al paso 3. Si hay etiquetas no revisadas, pero cada etiqueta no revisada está en un nodo  $i \in T$  para el cual  $\pi_j < W$ , entonces hacer  $W = \max \{ \pi_i \mid \pi_i < W \}$ .

1.2 Encontrar un nodo  $i$  con una etiqueta no revisada, donde  $i \in S$

o

$i \in T$  y  $\pi_i \neq W$ . Si  $i \in S$ , ir al paso 1.3; si  $i \in T$ , ir al paso 1.4.

1.3 Revisar la etiqueta del nodo  $i$  ( $i \in S$ ) como sigue. Para cada arco  $(i,j) \in X$  incidente a  $i$ , si  $\pi_j < w_{ij}$  y  $\pi_j < W$ , entonces dar al nodo  $j$  la etiqueta "i" (reemplazando cualquier etiqueta existente) y hacer  $\pi_j = w_{ij}$ . Regresar al paso 1.1.

1.4 Revisar la etiqueta del nodo  $i$  ( $i \in T$ ) como sigue. Si el nodo  $i$  es expuesto, vaya al paso 2. De otra manera, identificar el único arco  $(i,j) \in X$  incidente al nodo  $i$  y dar al nodo  $j$  la etiqueta "i". Regresar al paso 1.1.

**Paso 2** Aumentación

Una trayectoria aumentante ha sido encontrada, terminando en el nodo  $i$  (identificada en el paso 1.4). Los nodos precediendo al nodo  $i$  en la trayectoria se identifican hacia atrás de etiqueta a etiqueta. Aumentar  $X$  agregándole todos los arcos de la trayectoria aumentante que no estén en  $X$ , y remover aquellos que sí están. Remover todas las etiquetas de los nodos. Hacer  $\pi_j = -\infty$ , para cada nodo  $j$  en  $T$ . Regresar al paso 1.0.

**Paso 3** Etiquetación húngara

No existen trayectorias aumentantes, y el acoplamiento  $X$  es un acoplamiento max-min de cardinalidad máxima. Alto.

Como un ejemplo de la aplicación del algoritmo del umbral, sea la bigráfica ponderada mostrada en la figura 2.13, con su correspondiente matriz de pesos. Al aplicar el algoritmo, el acoplamiento max-min que se obtiene es de cardinalidad tres y el peso mínimo de sus arcos es  $-4$ , que coincide con el último valor del umbral  $w = -4$  y se muestra en el diagrama c) de la figura 2.13.

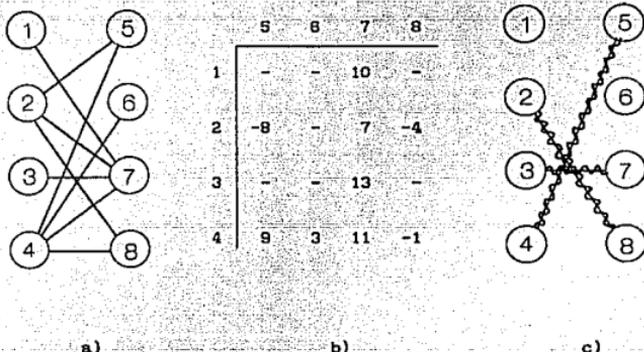


Figura 2.13

#### Primera iteración

Comenzamos con el acoplamiento vacío  $X = \emptyset$ ,  $W = +\infty$ ,  
 $\Pi_5 = \Pi_6 = \Pi_7 = \Pi_8 = -\infty$ .

Asignamos a cada nodo expuesto en  $S$  la siguiente etiqueta  $[1] = \theta$ ,  
 $[2] = \theta$ ,  $[3] = \theta$ ,  $[4] = \theta$ . Al revisar las etiquetas de los nodos  
 expuestos tenemos

$$\begin{aligned} \pi_5 &= 9 & \pi_6 &= 3 & \pi_7 &= 13 & \pi_8 &= -1 \\ [5] &= 4, [6] &= 4, [7] &= 3, [8] &= 4 \end{aligned}$$

como las únicas etiquetas no revisadas están en nodos  $i \in T$  con  $\pi_j < W$ , tomando  $\max \{ 9, 3, 13, -1 \}$  reducimos el valor de  $W$  quedando como  $W = 13$  y procedemos a revisar las etiquetas del nodo agregado al árbol que es este caso es  $i = 7$ . Como el nodo 7 es expuesto se ha encontrado una trayectoria aumentante:  $7, (7,3), 3$  ( ver figura 2.14 ); mejorando el acoplamiento tenemos  $X = \{ (3,7) \}$ , hacemos  $\pi_5 = \pi_6 = \pi_7 = \pi_8 = -\infty$  y realizamos la siguiente iteración.

#### Segunda iteración

Los nodos expuestos en  $S$  son  $[1] = 0, [2] = 0, [4] = 0$  al revisarlos obtenemos que

$$\begin{aligned} \pi_5 &= 9 & \pi_6 &= 3 & \pi_7 &= 11 & \pi_8 &= -1 \\ [5] &= 4, [6] &= 4, [7] &= 4, [8] &= 4 \end{aligned}$$

y tenemos que reducir  $W = \max \{ 9, 3, 11, -1 \}$ , por lo tanto  $W = 11$ , revisando al nodo 7, asignamos a  $[3] = 7$ ; no sucede nada al revisar al nodo 3, disminuimos  $W = 9$ , revisamos al nodo 5 encontrando que es expuesto, la trayectoria aumentante es  $5, (5,4), 4$  y mejoramos el acoplamiento  $X = \{ (3,7), (4,5) \}$ ; hacemos  $\pi_j = -\infty$  para todo  $j \in T$ . Ver la figura 2.14 b)

#### Tercera iteración

Los nodos expuestos en  $S$  son  $[1] = 0$  y  $[2] = 0$  al revisarlos tenemos que

$$\pi_5 = -8 \quad \pi_7 = 10 \quad \pi_8 = -4 \quad [5] = 2, [7] = 1, [8] = 2$$

como  $W = 9$ , revisando al nodo 7 asignamos  $[3] = 7$ , no sucede nada al revisar al nodo 3, disminuimos nuevamente  $W = -4$  revisamos el nodo 8 encontramos que es expuesto, la trayectoria aumentante es  $8, (8,2), 2$ , al mejorar el acoplamiento tenemos que  $X = \{ (3,7), (4,5), (2,8) \}$ ; hacemos  $\pi_j = -\infty$  para todo  $j \in T$ . Ver figura 2.14 c).

### Cuarta iteración

El único nodo expuesto es  $[1] = 0$  al revisarlo tenemos que  $U_7 = 10$ ,  $[7] = 1$ , el nodo 7 es revisado asignando la siguiente etiqueta  $[3] = 7$ , el nodo 3 es revisado sin que ocurra nada y como todos los nodos etiquetados están revisados no existen trayectorias aumentantes y el acoplamiento  $X = \{(3,7), (4,8), (2,8)\}$  es un acoplamiento max-min de cardinalidad máxima.

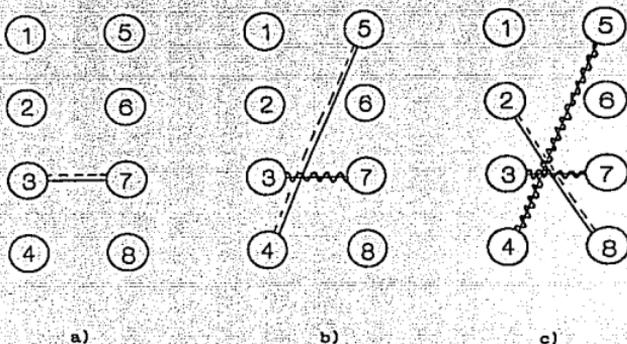


Figura 2.14

## 2.7 EL METODO HUNGARO PARA EL PROBLEMA DEL ACOPLAMIENTO DE PESO MAXIMO.

En esta sección trataremos el problema del acoplamiento de peso máximo en gráficas bipartitas, que consiste en encontrar un acoplamiento que cumpla que la suma de los pesos de sus arcos sea máximo. El procedimiento que se propone para el problema del acoplamiento de peso máximo en bigráficas ponderadas es un método primal-dual de programación lineal llamado húngaro por H. W. Kuhn en reconocimiento del matemático Egerváry.

Por simplicidad, supongamos una gráfica bipartita completa  $G=(S,T,SxT)$ , con  $|S| = m, |T| = n, m \leq n$ . Si tenemos una gráfica  $G'$  que no es completa, asignando un peso de cero a los arcos faltantes de  $SxT$  la solución obtenida para la gráfica  $G$  será válida para la gráfica  $G'$ . Formulando el problema de acoplamiento de peso máximo como un problema de programación lineal (P) tenemos :

$$\begin{aligned}
 \text{Max } z &= \sum_{i \in \overline{1, m}} \sum_{j \in \overline{1, n}} w_{ij} x_{ij} \\
 \text{s.c.} & \\
 \sum_{j \in \overline{1, n}} x_{ij} &\leq 1 \quad i \in \overline{1, m} \\
 \sum_{i \in \overline{1, m}} x_{ij} &\leq 1 \quad j \in \overline{1, n} \\
 x_{ij} &\geq 0.
 \end{aligned}
 \tag{P}$$

Donde

$$\begin{aligned}
 x_{ij} = 1 &\Rightarrow (i, j) \in X \\
 x_{ij} = 0 &\Rightarrow (i, j) \notin X \\
 w_{ij} &\text{ es el peso del arco } (i, j)
 \end{aligned}$$

Su P.P.L. dual (D) es

$$\begin{aligned}
 \text{Min } w &= \sum_{i \in \overline{1, m}} u_i + \sum_{j \in \overline{1, n}} v_j \\
 \text{s.c.} & \\
 u_i + v_j &\geq w_{ij} \\
 u_i &\geq 0 \\
 v_j &\geq 0
 \end{aligned}$$

utilizando el teorema de holguras complementarias de la teoría de P.L. las condiciones de ortogonalidad que son necesarias y suficientes para la óptimalidad de las soluciones primal y dual son :

$$x_{ij} > 0 \Rightarrow u_i + v_j = w_{ij} \tag{1}$$

$$u_i > 0 \Rightarrow \sum_{j \in \overline{1, n}} x_{ij} = 1 \tag{2}$$

$$v_j > 0 \Rightarrow \sum_{i \in \overline{1, m}} x_{ij} = 1 \tag{3}$$

El método húngaro mantiene la factibilidad primal y dual en todo el tiempo, además mantiene satisfactoriamente todas las condiciones de ortogonalidad excepto las condiciones (2). El número de tales condiciones no satisfechas decrece monótonamente durante el transcurso del algoritmo.

El procedimiento comienza con el acoplamiento factible  $X = \emptyset$  y con la solución factible dual  $u_i = W$ , donde  $W = \max \{ w_{ij} \}$ , y  $v_j = 0$ , para todo  $i, j$ . Estas soluciones primal y dual claramente satisfacen todas las condiciones (1) y (3) pero no las condiciones (2).

En el paso general del procedimiento,  $X$  es factible,  $u_i$  y  $v_j$  son dual factibles, todas las condiciones (1) y (3) se satisfacen, pero no algunas de las condiciones (2). Entonces tratamos, por medio del procedimiento de etiquetación de encontrar una trayectoria aumentante dentro de la subgráfica conteniendo únicamente arcos  $(i, j)$  para los cuales  $u_i + v_j = w_{ij}$ . En particular, una trayectoria aumentante está formada de un nodo expuesto  $i$  en  $S$  para el cual necesariamente  $u_i > 0$ . Si tal trayectoria puede ser encontrada el nuevo acoplamiento será factible, todas las condiciones (1) y (3) continúan satisfechas y una más de las condiciones (2) será ahora satisfecha. Si el aumento no es posible, entonces un cambio de  $\delta$  es hecho en las variables duales, sustrayendo  $\delta > 0$  de  $u_i$  para cada nodo en  $S$  etiquetado y sumando  $\delta$  a  $v_j$  para cada nodo etiquetado en  $T$ .

El cálculo del aumento es tal que solamente los arcos  $(i, j)$  para los cuales se cumple que  $u_i + v_j = w_{ij}$  son colocados en los árboles alternantes. Si la construcción de los árboles alternantes concluye sin que una trayectoria aumentante haya sido encontrada, entonces ocurre una de las siguientes dos cosas. La etiquetación es realmente húngara y el acoplamiento es de máxima cardinalidad o no es posible continuar agregando arcos a los árboles porque todos los arcos  $(i, j)$  disponibles para ese propósito son tales que  $u_i + v_j > w_{ij}$ .

El algoritmo empieza con el acoplamiento vacío,  $X_0$  y produce acoplamientos  $X_1, X_2, \dots, X_k$ , conteniendo  $1, 2, \dots, k$  arcos. Cada uno de estos acoplamientos es de peso máximo, con respecto a todos los demás de la misma cardinalidad, como se demostrará. (Notemos que al final del procedimiento se detecta el acoplamiento de peso máximo y este no necesariamente es de cardinalidad máxima).

Spongamos que queremos un acoplamiento de peso máximo, sujeto a la restricción de que no contenga más de  $k$  arcos. Al P.P.L. le podemos agregar una restricción mas:

$$\sum_{i,j} x_{ij} \leq k$$

Si relacionamos esta restricción con una variable dual  $\lambda$  y haciendo las modificaciones en el P.P.L. dual, las nuevas condiciones de ortogonalidad son:

$$\begin{aligned} x_{ij} > 0 & \rightarrow u_i + v_j + \lambda = u_{ij} \\ u_i > 0 & \rightarrow \sum_j x_{ij} = 1 \\ v_j > 0 & \rightarrow \sum_i x_{ij} = 1 \\ \lambda > 0 & \rightarrow \sum_{i,j} x_{ij} = k \end{aligned}$$

Sea  $X_k$  el acoplamiento de cardinalidad  $k$  obtenido por el algoritmo, y  $\bar{u}_i, \bar{v}_j$  sea la solución dual factible. Escojamos  $\lambda = \min \{ \bar{u}_i \}$ . Entonces  $X_k, \bar{u}_i - \lambda, \bar{v}_j, \lambda$  son soluciones factibles primal y dual para el problema de cardinalidad  $k$  y satisfacen las nuevas condiciones de ortogonalidad indicadas. Se sigue que  $X_k$  es de peso máximo, con respecto a todos los otros acoplamientos conteniendo  $k$  arcos.

Enseguida se escribirá el algoritmo que obtiene el acoplamiento de peso máximo en una gráfica bipartita.

### Algoritmo del acoplamiento de peso máximo en gráficas bipartitas

#### Paso 0 Inicio

Sea  $G = (S, T, A)$  gráfica bipartita y para cada arco  $(i, j) \in A$  su peso es  $w_{ij}$ . Haga  $X = \emptyset$ . Sea  $u_i = \max \{ w_{ij} \}$  para cada nodo  $i \in S$ . Sea  $v_j = 0$  y  $\pi_j = +\infty$  para cada nodo  $j \in T$ . Ningún nodo está etiquetado.

#### Paso 1 Etiquetación

1.0 Dar la etiqueta "0" a cada nodo expuesto en  $S$ .

1.1 Si no hay etiquetas sin revisar o si hay etiquetas sin revisar pero cada una de ellas está asociada a un nodo  $i \in T$  para el cual  $\pi_j > 0$ , vaya al paso 3.

1.2 Encontrar un nodo  $i$  con un etiqueta no revisada, donde  $i \in S$  ó  $i \in T$  y  $\pi_j = 0$ . Si  $i \in S$ , ir al paso 1.3; si  $i \in T$ , ir al paso 1.4.

1.3 Revisar la etiqueta en el nodo  $i$  ( $i \in S$ ) como sigue. Para cada arco  $(i, j) \in X$  incidente al nodo  $i$ , si  $u_i + v_j - w_{ij} < \pi_j$ , entonces dar al nodo  $j$  la etiqueta "1" (reemplazando cualquier etiqueta existente) y hacer  $\pi_j = u_i + v_j - w_{ij}$ . Regresar al paso 1.1.

1.4 Revisar la etiqueta en el nodo  $i$  ( $i \in T$ ) como sigue. Si el nodo  $i$  es expuesto, vaya al paso 2. De otra forma, identificar el único arco  $(i, j) \in X$  incidente al nodo  $i$  y dar al nodo  $j$  la etiqueta "1". Regresar al paso 1.1.

#### Paso 2 Aumentación

Una trayectoria aumentante ha sido encontrada, terminando en el nodo  $i$  (identificado en el paso 1.4). Los nodos precediendo al nodo  $i$  en la trayectoria son identificados hacia atrás utilizando las etiquetas. Aumentar  $X$  intercambiando los arcos en la trayectoria aumentante que no están en  $X$  por aquellos que estaban. Hacer  $\pi_j = +\infty$ , para cada nodo  $j$  en  $T$ . Remover todas las etiquetas de los nodos. Regresar al paso 1.0.

**Paso 3 Cambio en las variables duales**

Encuentre

$$\begin{aligned} \delta_1 &= \min \{ u_i \mid i \in S \}, \\ \delta_2 &= \min \{ \pi_j \mid \pi_j > 0, j \in T \}, \\ \theta &= \min \{ \delta_1, \delta_2 \} \end{aligned}$$

Restar  $\theta$  de  $u_i$ , para cada nodo etiquetado  $i \in S$ . Sumar  $\theta$  a  $v_j$  para cada nodo  $j \in T$  con  $\pi_j = 0$ . Restar  $\theta$  de  $\pi_j$  para cada nodo etiquetado  $j \in T$  con  $\pi_j > 0$ . Si  $\theta < \delta_1$  ir al paso 1.1. De otra manera,  $X$  es un acoplamiento de peso máximo y las variables  $u_i$  y  $v_j$  son una solución óptima dual. Alto.

Como un ejemplo de la aplicación de este algoritmo, utilizaremos la bigráfica ponderada mostrada en la figura 2.15, con su correspondiente matriz de pesos. Al aplicar el algoritmo, el acoplamiento de peso máximo que se obtiene es de cardinalidad cuatro (Note que no es de cardinalidad máxima) y el valor de su función objetivo es 24. La suma de las variables duales finales suma igualmente 24, lo que indica que ambas son soluciones óptimas. La solución se muestra en el diagrama c) de la figura 2.15.

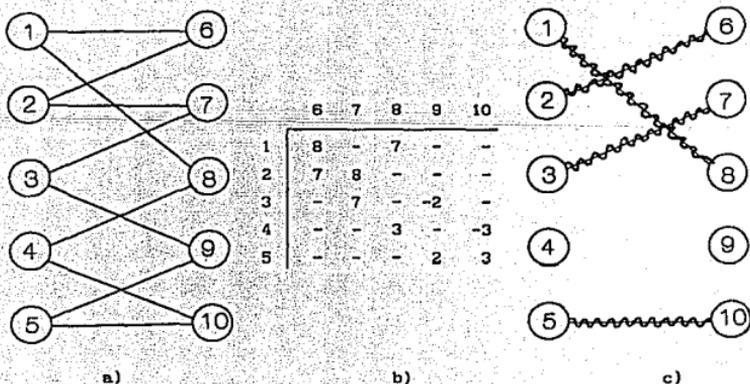


Figura 2.15

### Iteración 1

Comenzamos con el acoplamiento  $X = \emptyset$

$$\begin{array}{cccccc} u_1 = 8 & u_2 = 8 & u_3 = 8 & u_4 = 8 & u_5 = 8 \\ v_6 = 0 & v_7 = 0 & v_8 = 0 & v_9 = 0 & v_{10} = 0 \\ \pi_6 = \pi_7 = \pi_8 = \pi_9 = \pi_{10} = +\infty \end{array}$$

Ningún nodo está etiquetado, damos a los nodos expuestos en  $S$  las siguientes etiquetas

$$[1] = 0 \quad [2] = 0 \quad [3] = 0 \quad [4] = 0 \quad [5] = 0$$

al revisar la etiqueta del nodo [1] tenemos que  $[6] = 1$  y  $\pi_6 = 0$ ,  $[8] = 1$  y  $\pi_8 = 1$ , por lo tanto como  $\pi_6 = 0$ , el nodo 6 es candidato a entrar al árbol, al revisar su etiqueta nos damos cuenta que es expuesto, hemos descubierto una trayectoria aumentante que utiliza solo el arco (1,6), mejoramos el acoplamiento  $X = \{ (1,6) \}$ . Todas las etiquetas de los nodos se remueven.

### Iteración 2

En esta iteración fácilmente obtenemos la trayectoria aumentante que une a los nodos 2 y 7 por medio del arco (2,7). Mejoramos el acoplamiento y borramos etiquetas tenemos que  $X = \{ (1,6), (2,7) \}$ .

### Iteración 3

Damos a cada nodo expuesto en  $S$  las siguientes etiquetas

$$[3] = 0 \quad [4] = 0 \quad [5] = 0 \quad \pi_6 = \pi_7 = \pi_8 = \pi_9 = \pi_{10} = +\infty$$

revisando las etiquetas de los nodos expuestos de  $S$  llegamos a

$$[7] = 3 \quad [8] = 4 \quad [9] = 5 \quad [10] = 5 \quad \pi_7 = 1 \quad \pi_8 = 5 \quad \pi_9 = 6 \quad \pi_{10} = 5$$

como las únicas etiquetas no revisadas son de nodos  $j \in T$  para los cuales  $\pi_j > 0$  tenemos que realizar la modificación de las variables duales

$$\theta_1 = \min \{ 8, 8, 8, 8, 8 \} = 8$$

$$\theta_2 = \min \{ 1, 5, 6, 5 \} = 1$$

$$\theta = \min \{ \theta_1, \theta_2 \} = \theta_2 = 1$$

$$u_3 = 7 \quad u_4 = 7 \quad u_5 = 7 \quad \pi_7 = 0 \quad \pi_8 = 4 \quad \pi_9 = 5 \quad \pi_{10} = 4$$

como  $\theta < \theta_1$  al menos un nodo puede ser revisado (en este caso el nodo 7). Al revisar el nodo 7 etiquetamos  $[2] = 7$ . Al revisar el nodo 2 etiquetamos  $[6] = 2$  y  $\Pi_6 = 1$ , teniendo que modificar las variables duales

$$\theta_1 = \min \{ 8, 8, 7, 7, 7 \} = 7$$

$$\theta_2 = \min \{ 1, 4, 5, 4 \} = 1$$

$$\theta = \min \{ \theta_1, \theta_2 \} = \theta_2 = 1$$

al realizar los cambios tenemos que

$$u_1 = 8 \quad u_2 = 7 \quad u_3 = 6 \quad u_4 = 6 \quad u_5 = 6$$

$$v_6 = 0 \quad v_7 = 1 \quad v_8 = 0 \quad v_9 = 0 \quad v_{10} = 0$$

$$\Pi_6 = \Pi_7 = 0 \quad \Pi_8 = 3 \quad \Pi_9 = 4 \quad \Pi_{10} = 3$$

como  $\theta < \theta_1$  revisamos el nodo 6 dando la etiqueta  $[1] = 6$ , revisamos el nodo 1 que produce la etiqueta  $[8] = 1$  y  $\Pi_8 = 1$ , modificamos de nueva cuenta las variables duales con

$$\theta_1 = 6 \quad \theta_2 = 1 \quad \theta = 1 \quad \theta < \theta_1, \text{ obtenemos así}$$

$$u_1 = 7 \quad u_2 = 8 \quad u_3 = 5 \quad u_4 = 5 \quad u_5 = 5$$

$$v_6 = 1 \quad v_7 = 2 \quad v_8 = 0 \quad v_9 = 0 \quad v_{10} = 0$$

$$\Pi_6 = \Pi_7 = \Pi_8 = 0 \quad \Pi_9 = 3 \quad \Pi_{10} = 2$$

al revisar el nodo 8 descubrimos que es expuesto, por lo tanto forma parte de la trayectoria aumentante: 8, (8,1), 1, (1,6), 6, (6,2), 2, (2,7), 7, (7,3), 3; por lo tanto

$$X = \{ (1,6), (2,7) \} \cup \{ (1,8), (2,6), (3,7) \} - \{ (1,6), (2,7) \}$$

$$X = \{ (1,8), (2,6), (3,7) \}.$$

Que se muestra en la figura 2.16 a).

#### Iteración 4

Al dar a los nodos expuestos en S la etiqueta  $[4] = \theta$   $[5] = \theta$  y

revisarlos tenemos que  $\Pi_8 = 2$   $\Pi_9 = 3$   $\Pi_{10} = 2$

$$[8] = 4 \quad [9] = 5 \quad [10] = 5$$

al cambiar variables duales

$$\delta_1 = 5 \quad \delta_2 = 1 \quad \delta = 1 \quad \delta < \delta_1, \text{ obtenemos así}$$

$$u_1 = 3 \quad u_5 = 3 \quad \pi_6 = 0 \quad \pi_9 = 1 \quad \pi_{10} = 0$$

al revisar el nodo 10 descubrimos que es nodo expuesto con una trayectoria aumentante 10, (10,5), 5; mejoramos el acoplamiento:

$$X = \{ (1,8), (2,6), (3,7), (5,10) \}$$

Este acoplamiento se muestra en la figura 2.16 b)

Iteración 5

$$\text{Tenemos } \begin{array}{ccccc} u_1 = 7 & u_2 = 6 & u_3 = 5 & u_4 = 3 & u_5 = 3 \\ v_6 = 1 & v_7 = 2 & v_8 = 0 & v_9 = 0 & v_{10} = 0 \end{array}$$

el único nodo expuesto es 4, al revisarlo tenemos  $\{8\} = 4$ ,  $\{10\} = 4$  y  $\pi_8 = 0$ ,  $\pi_{10} = 6$ , revisando todos los nodos posibles llegamos a que los únicos que no están revisados son el nodo 9 y el nodo 10 pero  $\pi_9 = 0$ ,  $\pi_{10} = 6$ , por lo tanto realizamos un cambio en variables duales

$$\delta_1 = 3 \quad \delta_2 = 7 \quad \delta = 3, \text{ obtenemos así}$$

$$\begin{array}{ccccc} u_1 = 4 & u_2 = 3 & u_3 = 2 & u_4 = 0 & u_5 = 3 \\ v_6 = 4 & v_7 = 5 & v_8 = 3 & v_9 = 0 & v_{10} = 0 \end{array}$$

y como  $\delta = \delta_1$  estamos en el óptimo. La suma de las variables duales es 24, por lo que el valor del acoplamiento obtenido  $X = \{ (1,8), (2,6), (3,7), (5,10) \}$  debe ser también 24, lo que se cumple.

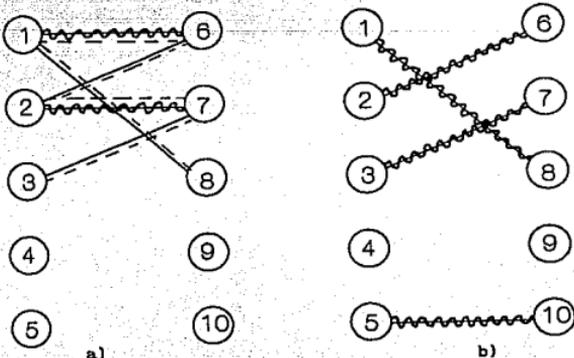


Figura 2.16

## 2.8 ACOPLAMIENTO GALE-SHAPLEY

En esta sección veremos un tipo especial de acoplamiento, para el cual Gale y Shapley desarrollaron un algoritmo que puede ser aplicado a varios problemas que involucran la preferencia en la formación de parejas. Supongamos que en cierta comunidad consistente de  $n$  hombres y  $n$  mujeres, cada persona ordena a todas las personas del sexo opuesto en concordancia a su preferencia para ser su pareja matrimonial.

Por ejemplo supongamos que hay cuatro hombres  $\alpha, \beta, \gamma, \delta$  y 4 mujeres A, B, C, y D sus preferencias pueden ser ilustradas en una matriz de orden como la que sigue :

	A	B	C	D
$\alpha$	1,3	2,3	3,2	4,3
$\beta$	1,4	4,1	3,3	2,2
$\gamma$	2,2	1,4	3,4	4,1
$\delta$	4,1	2,2	3,1	1,4

El primer número de cada par en la matriz es el orden en la preferencia de una mujer dado por un hombre, el segundo es el número de lugar dado a un hombre por una mujer. Así  $\alpha$  ordena a A como primera, B segunda, C tercera y D cuarta; mientras que A ordena  $\delta$  como primero,  $\gamma$  segundo,  $\alpha$  tercero y  $\beta$  cuarto.

Un acoplamiento completo de hombres y mujeres es llamado acoplamiento inestable si bajo el hay un hombre y una mujer que no están casados con sus parejas, pero prefieren otras que a sus parejas asignadas.

Existen tantos conjuntos posibles de matrimonios como el número de acoplamientos completos, es decir 4!. Pero algunos de estos son inestables, por ejemplo  $\delta$  con A,  $\beta$  con B,  $\gamma$  con C,  $\alpha$  con D. Bajo esta asignación  $\delta$  quisiera dejar a A, su cuarta elección, en favor de B, su segunda elección, mientras que  $\beta$  quisiera dejar a B, su cuarta elección, en favor de A, su primera elección.

Un acoplamiento estable de hombres y mujeres se llama acoplamiento hombre optimal si todo hombre está al menos tan bien bajo este acoplamiento como en cualquier otro acoplamiento estable.

Teorema 2.15 de Gale-Shapley.

Para cualquier conjunto de preferencias de hombres y mujeres, existe un acoplamiento hombre optimal.

Demostración. Será constructiva.

Para iniciar, cada hombre ofrece matrimonio a su chica favorita. Cada chica que recibe más de una propuesta rechaza a todos pero le da una esperanza a su favorito de entre los que le ofrecieron matrimonio, sin embargo no lo acepta, pero lo coloca en una lista con la posibilidad de que alguno mejor llegue más tarde.

Estamos listos para la segunda etapa. Aquellos hombres que fueron rechazados ahora ofrecen matrimonio a su segunda elección. Cada chica recibiendo propuestas escoge su favorito del grupo consistente de las nuevas propuestas y del hombre en su lista, si es que hay alguno. Ella rechaza otra vez a todos, pero al mejor le da una esperanza. Eventualmente ( a lo más en  $n^2 - n + 1$  etapas ) cada chica habrá recibido una propuesta, y mientras que alguna chica no haya recibido propuesta habrá rechazos y nuevas propuestas, pero como ningún hombre puede prometerle matrimonio a la misma chica más de una vez, toda chica recibe seguramente una propuesta en algún momento, tan pronto como la última chica tenga su propuesta, el cortejo se declara terminado y cada chica tiene que aceptar al hombre en su lista. ■

Tenemos que demostrar que al término del algoritmo el acoplamiento resultante es un conjunto estable de matrimonios:

Supongamos que un hombre X y una mujer Y no están casados entre si, pero X prefiere a Y que a su propia mujer. Entonces X debió haberle propuesto a Y en alguna etapa matrimonio y subsecuentemente fue rechazado en favor de alguno que Y consideraba mejor, entonces Y debe preferir a su esposo que a X y por lo tanto no hay inestabilidad.

Al aplicar el algoritmo a la matriz del ejemplo dado al inicio de esta sección tenemos lo siguiente:

El conjunto de propuestas  $P_i$  aparece en la siguiente tabla, las propuestas con asterisco fueron rechazados

ESTA TESIS NO DEBE  
SALIR DE LA BIBLIOTECA

Propuestas	P <sub>1</sub>	P <sub>2</sub>	P <sub>3</sub>	P <sub>4</sub>	P <sub>5</sub>	P <sub>6</sub>
$\alpha$	1	1	1	1	2	3
$\beta$	1	4	4	4	4	4
$\gamma$	2	2	2	1	1	1
$\delta$	4	4	2	2	2	2

entonces el conjunto final de matrimonios es:

hombre  $\alpha$  con mujer C

hombre  $\beta$  con mujer D

hombre  $\gamma$  con mujer A

hombre  $\delta$  con mujer B

de (1) Y (2)

$$\begin{aligned} |M_H \cup M_{G_0}^*| &\geq |X_H \cup X_{HG_0} \cup X_{G_0}| \\ &\geq |X_H| \end{aligned}$$

es decir  $M_H \cup M_{G_0}^*$  es un acoplamiento máximo en  $G$ . ■

Un ejemplo de un árbol húngaro se muestra en la figura 3.10

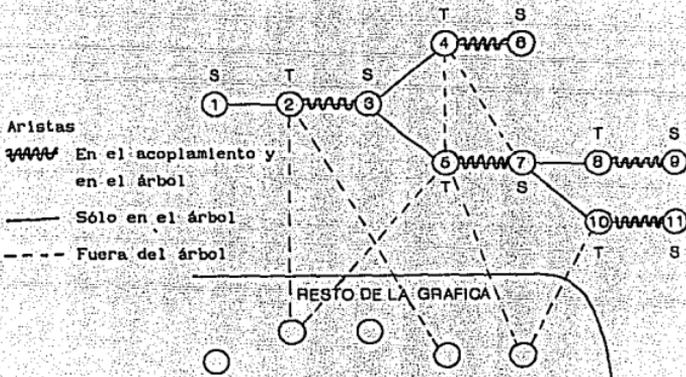


Figura 3.10

Por el análisis hecho anteriormente existen dos diferencias entre el algoritmo del acoplamiento máximo en bigráficas y en gráficas generales, que se deben incorporar :

Primero es necesario detectar y comprimir flores ; segundo, debe de ser posible descubrir trayectorias aumentantes apropiadas a través de las flores comprimidas, para que así pueda reconstruirse una trayectoria aumentante en la gráfica original. La detección de flores es sencilla y la compresión no es problema, sin embargo no es trivial realizar un algoritmo que haga estas operaciones de la manera más eficiente posible. En la siguiente sección se proporciona el algoritmo de solución al problema del acoplamiento de cardinalidad máxima en gráficas no bipartitas.

El algoritmo de solución al problema del acoplamiento máximo que daremos en esta sección se basa en una implementación del algoritmo de Edmonds. Se desarrollará un procedimiento de etiquetación que no requiere la actual contracción de flores en la gráfica, pero son tratadas como si estuvieran comprimidas.

Para esto solamente necesitamos tener un registro de las flores más externas, mismas que son identificadas por sus nodos base. Asociado con cada nodo  $i$ , un índice  $b(i)$  indica el nodo base de la flor en la que está contenido. Si un nodo  $i$  no está contenido en una flor entonces  $b(i) = i$ . Así dos nodos  $i, j$  están en la misma flor si y sólo si  $b(i) = b(j)$ .

Cuando se forma una nueva flor, el nodo base  $b$  de ésta se identifica y se asigna  $b(i)$  a  $b$ , para todos los nodos  $i$  en la flor. Así, una flor determinada queda identificada por su nodo base.

El algoritmo enraiza un árbol en cada nodo expuesto y agranda todos los árboles simultáneamente, tratando de encontrar una trayectoria aumentante o una flor con respecto a un acoplamiento. En el primer caso, aumentaremos el acoplamiento y en el segundo, identificamos la base de la flor, los nodos que la integran y regresamos a buscar una trayectoria aumentante.

Inicialmente se da la etiqueta 'S:  $\sigma$ ' a todos los nodos expuestos, para enraizar arboles. Posteriormente se asignan S-etiquetas y T-etiquetas a los nodos. Una S-etiqueta indica la existencia de una trayectoria alternante de longitud par al nodo raíz, y una T-etiqueta indica la existencia de una trayectoria de longitud impar ( un nodo recibe ambos tipos de etiquetas si y sólo si es un nodo no base de la flor más externa). Un ejemplo de trayectorias aumentantes que surgen entre los nodos raíz de dos diferentes árboles, se indica en la figura 3.11



La nueva flor contiene todos los nodos  $k$ , tal que

$$b(k) \in \{ b(i_m), b(i_{m+1}), \dots, b(i_p), b(j_{m+1}), b(j_{m+2}), \dots, b(j_q) \}$$

Como un ejemplo consideremos la situación que se muestra en la siguiente figura:

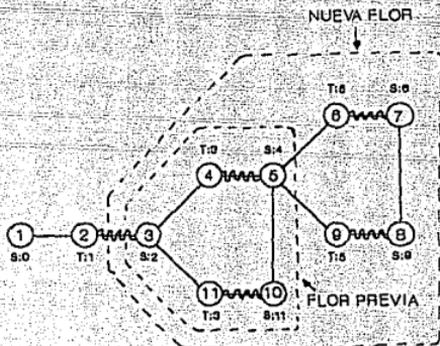


Figura 3.12

Aquí vemos S-etiquetas en los nodos 7 y 8 y  $(7,8) \in X$ . Como una flor ha sido detectada realizando el rastreo regresivo de los nodos 7 y 8 obtenemos la secuencias

$$1, 2, 3, 4, 5, 6, 7$$

y

$$1, 2, 3, 4, 5, 9, 8$$

En este caso  $i_m = j_m = 5$  y  $b(5) = 3$  y como el nodo 5 era ya parte de una flor con el nodo 3 como base, los nodos 3, 4, 5, 6, 7, 8, 9, 10 y 11 están en la nueva flor, el nodo 3 es su base y los nodos 1, 2 y 3 forman su tallo.

Entre cada nodo no base en la nueva flor y la raíz del árbol existe tanto una trayectoria alternante de longitud par como una impar. Este hecho debe ser indicado por la existencia tanto de una S-etiqueta y una T-etiqueta en cada uno de tales nodos.

Supongamos que la flor fue detectada realizando el rastreo regresivo de los nodos  $i = i_p$  y  $j = j_q$ , donde  $i$  y  $j$  tienen S-etiquetas y  $(i, j) \in X$  ( las reglas para el caso en que  $i$  y  $j$  tengan T-etiquetas y  $(i, j) \in X$  son semejantes ).

Veremos el manejo de los nodos  $i_{m+1}, i_{m+2}, \dots, i_p$ . ( las reglas para los nodos  $j_{m+1}, j_{m+2}, \dots, j_q$  son por supuesto similares). Las S-etiquetas en  $i_p, i_{p-2}, \dots, i_{m+2}$  y las T-etiquetas en  $i_{p-1}, i_{p-3}, \dots, i_{m+1}$  fueron utilizadas en el regreso, por esto cualquier etiqueta faltante deben ser una T-etiqueta en  $i_p, i_{p-2}, \dots, i_{m+2}$  o una S-etiqueta en  $i_{p-1}, i_{p-3}, \dots, i_{m+1}$ . La etiqueta asignada a cualquier nodo  $i_r$  será tal que el rastreo regresivo a partir de esa etiqueta nos da la secuencia de nodos  $i_r, i_{r+1}, \dots, i_p, j_q, j_{q-1}, \dots, j_1$

Para aclarar el modo de proceder asignemos etiquetas faltantes a  $i_{m+1}, i_{m+2}, \dots, i_p$  en orden. Supongamos que  $i_r$  necesita una S-etiqueta, entonces debe de suceder que  $(i_r, i_{r+1}) \in X$  e  $i_{r+1}$  necesite una T-etiqueta. Damos a  $i_r$  la etiqueta 'S:  $i_{r+1}$ '. La T-etiqueta la debemos asignar a  $i_{r+1}$  para que el regreso se lleve a cabo correctamente.

Análogamente supongamos que  $i_r$  necesita una T-etiqueta, entonces necesariamente  $(i_r, i_{r+1}) \in X$ . Si también  $i_{r+1}$  necesita una S-etiqueta, damos a  $i_r$  la etiqueta 'T:  $i_{r+1}$ '. La S-etiqueta la debemos asignar a  $i_{r+1}$  para que el regreso se lleve a cabo correctamente.

Pero ahora supongamos que  $i_r$  necesita una T-etiqueta y que  $i_{r+1}$  tiene ya una S-etiqueta. Entonces  $i_r$  debe ser el nodo base de una flor preexistente, conteniendo  $i_{r+1}$ , y realizando el rastreo regresivo de la S-etiqueta en  $i_{r+1}$  nos conducirá de nuevo a  $i_r$ . Entonces es necesario corregir esta equivocación para no dar la etiqueta 'T:  $i_{r+1}$ ' al nodo  $i_r$ . Lo que tenemos que hacer para resolver este problema es encontrar el último nodo  $i_k$  en la secuencia  $i_r, i_{r+1}, \dots, i_p$  que está contenido en la flor preexistente con  $i_r$  como base. Necesariamente  $k \geq r+2$ . Entonces asignamos una etiqueta especial 'T:  $i_{k+1}, i_k$ ' a  $i_r$ . Esta etiqueta se interpreta como sigue: Existe

una trayectoria aumentante de longitud impar entre  $i_r$  y el nodo raíz. Para encontrar esta trayectoria, regresemos de la S-etiqueta en el nodo  $i_{k+1}$  a la raíz y también del nodo  $i_k$  al propio  $i_r$ . Los aristas así descubiertos junto con la arista  $(i_k, i_{k+1})$  adecuadamente ordenados, constituyen la trayectoria aumentante deseada.

Un ejemplo de la aplicación de etiquetas dentro de una flor se muestra en la figura 3.13 :

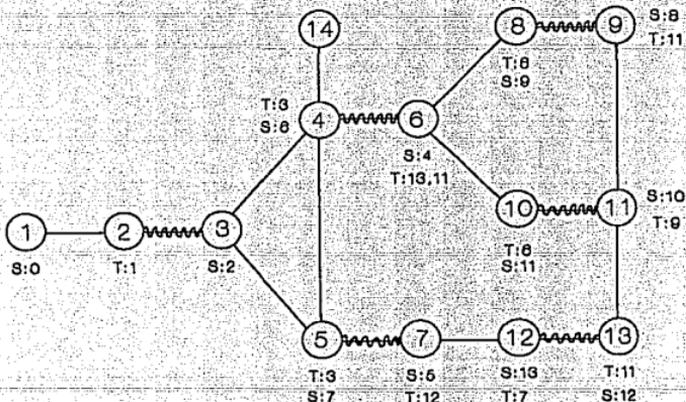


Figura 3.13

### 3.3.2 ALGORITMO DEL ACOPLAMIENTO DE MAXIMA CARDINALIDAD EN GRAFICAS NO BIPARTITAS.

#### Paso 0 Inicio

Sea  $G=(N,A)$  una gráfica dada. Sea  $X$  un acoplamiento, posiblemente el vacío. Hacer  $b(i) = 1$  para todo  $i \in N$ . Ningún nodo está etiquetado.

#### Paso 1 Etiquetación

- 1.0 Asignar la etiqueta 'S : 0' a cada nodo expuesto.
- 1.1 Si no hay etiquetas sin revisar ir al paso 4. De otra forma encontrar un nodo  $i$  con etiqueta no revisada. Si la etiqueta es una S-etiqueta ir al paso 1.2, si es una T-etiqueta ir al paso 1.3.
- 1.2 Revisar la S-etiqueta del nodo  $i$  realizando el siguiente procedimiento para cada arco  $(i,j) \in X$  incidente al nodo  $i$ . Si  $b(i) = b(j)$  no hacer nada. De otra manera, si el nodo  $j$  tiene una S-etiqueta, realizar el rastreo regresivo de las S-etiquetas en los nodos  $i,j$  y si nodos raíces diferentes son alcanzados, ir al paso 2; si el mismo nodo raíz es alcanzado ir al paso 3. Si el nodo  $j$  no está etiquetado asignar a  $j$  la etiqueta 'T : 1'. Cuando la revisión este completa regresar al paso 1.1.
- 1.3 Revisar la T-etiqueta del nodo  $i$  como sigue : encontrar el único arco  $(i,j) \in X$  incidente al nodo  $i$ . Si  $b(i)=b(j)$  no hacer nada. De otra manera, si el nodo  $j$  tiene una T-etiqueta, realizar el rastreo regresivo de las T-etiquetas en los nodos  $i,j$  y si nodos raíces diferentes son alcanzados, ir al paso 2; si el mismo nodo raíz es alcanzado ir al paso 3. Si el nodo  $j$  no está etiquetado asignar a  $j$  la etiqueta 'S : 1'. Regresar al paso 1.1.

#### Paso 2 Aumentación

Una trayectoria aumentante ha sido encontrada en el paso 1.2 o en el paso 1.3. Aumentar el acoplamiento  $X$ . Remover todas las etiquetas de los nodos y asignar  $b(i) = 1$ , para todo  $i$ . Regresar al paso 1.0.

### Paso 3 Floración

Una flor ha sido formada en el paso 1.2 o 1.3. Determinar los miembros y el nodo base de la nueva flor. Proporcionar etiquetas faltantes a todos los nodos no base en la nueva flor. Reasignar  $b(i)$  para todos los nodos  $i$  en la nueva flor. Regresar al paso 1.2 o 1.3 apropiadamente.

### Paso 4 Etiquetación húngara

La etiquetación es húngara. No existen trayectorias aumentantes y el acoplamiento  $X$  es de cardinalidad máxima. Alto.

Veamos la aplicación de este algoritmo a un ejemplo. Sea  $G = (N, A)$  una gráfica con  $|N| = 14$ ,  $|A| = 15$  con un acoplamiento inicial como el que se muestra en la figura 3.14. El elemento indicador de floraciones más externas  $b(j)$  para cada nodo  $j$ , es el siguiente

$j$	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$b(j)$	1	2	3	4	5	6	7	8	9	10	11	12	13	14

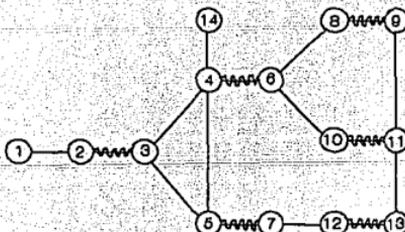
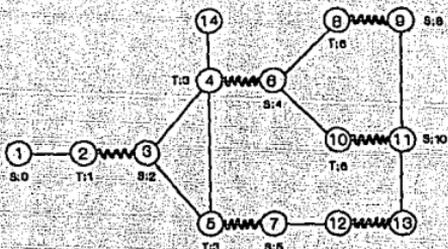


Figura 3.14

Al aplicar el algoritmo, comenzando a etiquetar desde el nodo expuesto 1, llegamos al etiquetado de la figura 3.15. Pero a partir del nodo 9, detectamos una S-etiqueta en el nodo 11, lo que implica que existe una flor  $B_1$ , formada por los nodos 6, 8, 9, 11 y 10, con nodo base igual a 6, por lo tanto los cambios en los identificadores  $b(j)$  son

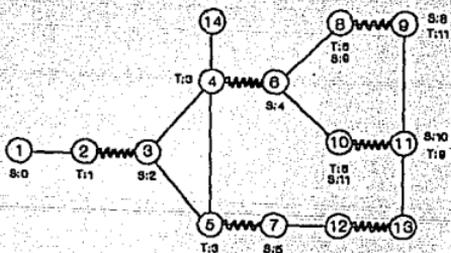
J	1	2	3	4	5	6	7	8	9	10	11	12	13	14
b(J)	1	2	3	4	5	6	7	6	6	6	6	12	13	14

Figura 3.15



Asignando etiquetas faltantes a los nodos no base de la flor B1, obtenemos el etiquetado de la figura 3.16 :

Figura 3.16



Al explorar el nodo 11, el arco (11,9) no es considerado porque  $b(9) = b(11)$ , i.e. pertenecen a la misma flor; en cambio por el arco (11,13) asignamos en el nodo 13 la T-etiqueta 'T: 11'. Al continuar revisando etiquetas, del nodo 12 que recibe una S-etiqueta, se descubre una nueva flor B2, porque el nodo 7 también está S-etiquetado, tomando  $i_p = 7$  e  $i_q = 12$  descubrimos el nodo-base de B2 que es  $i_m = 3$ . Las sucesiones  $i_{m+1}, i_{m+2}, \dots, i_p = 4, 6, 10, 11, 13, 12$  y  $j_{m+1}, \dots, j_q = 5, 7$ .

Cualquier nodo  $k$  en la flor debe cumplir que

$$b(k) \in \{ b(3), b(4), b(6), b(10), b(11), b(13), b(12), b(5), b(7) \}$$

o sea

$$b(k) \in \{ 3, 4, 6, 13, 12, 5, 7 \}$$

por lo tanto los nodos en B2 son 3,4,6,8,9,10,11,13,12,7,6. Los valores de  $b(j)$  son ahora:

J	1	2	3	4	5	6	7	8	9	10	11	12	13	14
b(j)	1	2	3	3	3	3	3	3	3	3	3	3	3	14

Asignamos fácilmente las etiquetas faltantes en los nodos no base 5,7 y 4 de la flor B2 como se aprecia en la figura 3.17 :

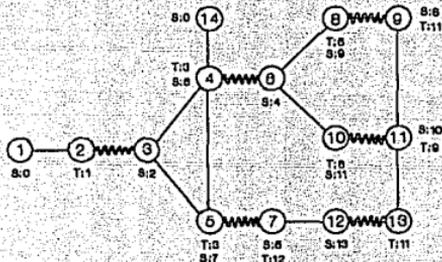


Figura 3.17

Pero el nodo 6 (que carece de T-etiqueta), tiene como nodo adyacente a 10 que tiene una S-etiqueta -esto sucede porque 6 es el nodo base de una flor B1, ya existente-. Así,  $i_k = 6$  es un nodo que carece de T-etiqueta, mientras que  $i_{k+1} = 10$  ya posee una S-etiqueta. El último nodo  $i_z$  de la sucesión  $i_{k+1}, \dots, i_p$  dentro de la flor preexistente B1 es  $i_z = 11$ . De esta manera, asignamos la etiqueta especial 'T : 13,11' al nodo 6, lo que indica la existencia de una trayectoria alternante impar entre el nodo raíz 1 y el nodo 6. Las etiquetas faltantes que restan por asignar a los nodos 12,13 nos dan el etiquetado de la figura 3.18 :



### 3.3.3 DUALIDAD.

Se establecerá un teorema de dualidad (cubiertas de nodos) para acoplamientos en gráficas no bipartitas, en lo que podría considerarse una generalización del teorema de König-Egerváry para el caso de bigráficas.

La dificultad que se presenta para que no se cumpla este teorema en gráficas no bipartitas es la presencia de ciclos impares conteniendo  $2r+1$  nodos. En estos ciclos, el acoplamiento máximo tiene  $r$  arcos y una cubierta debe tener al menos  $r+1$  nodos (una cubierta mínima debe tener exactamente  $r+1$  nodos). En la figura 3.20 observamos una gráfica  $G$  que es un ciclo impar de cardinalidad 3, el acoplamiento máximo consta de un sólo arco -el ondulado-, y necesitamos al menos 2 nodos (i.e.  $r+1=2$ ) para obtener una cubierta de nodos.

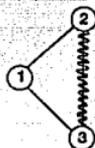


Figura 3.20

El problema dual del acoplamiento máximo en gráficas no bipartitas, es un problema de recubrimiento mínimo, pero de una clase especial de conjuntos de nodos, que son conocidos como conjuntos impares de nodos, y que veremos a continuación.

Sea  $G=(N,A)$  una gráfica y  $Q = \{ N_1, N_2, \dots, N_p \}$  una familia de subconjuntos de nodos de  $G$ , donde cada  $N_i$  contiene un número impar de elementos.

Si  $|N_i| = 1$ , decimos que  $N_i$  CUBRE todos los arcos que inciden en el único nodo de  $N_i$ , y que la CAPACIDAD de  $N_i$ , denotada  $c(N_i)$  es 1. Cuando  $|N_i| = 2k+1$ , con  $k \geq 1$ , entonces  $N_i$  cubre todos los arcos con ambos extremos inciendiendo en nodos que pertenezcan a  $N_i$  y la capacidad de  $N_i$  es  $c(N_i) = k$ . La familia  $Q$  es llamada RECUBRIMIENTO DE CONJUNTOS IMPARES DE NODOS, si cada arco de  $G$  es cubierto por al menos un subconjunto  $N_i \in Q$ . La capacidad de  $Q$ , denotada  $c(Q)$  es igual a la suma de las capacidades de todos los conjuntos impares contenidos en  $Q$ .

El teorema de dualidad para gráficas no bipartitas debido a Edmonds es el siguiente y nos dice que para cualquier gráfica G, el máximo número de arcos de un acoplamiento es igual a la capacidad mínima de un recubrimiento de conjuntos impares de nodos.

**Teorema 3.4** de Edmonds.

Sea  $G=(N,A)$  una gráfica,  $X_1$  acoplamiento en G, Q recubrimiento de conjuntos impares de nodos en G, entonces :

$$\text{Max } \{ |X_1| : X_1 \in G \} = \text{Min } \left\{ \sum_{N_i \in Q} c(N_i) \right\}$$

Para demostrar el teorema 3.4 utilizaremos el siguiente resultado

**Teorema 3.5**

Sean X un acoplamiento y Q un recubrimiento de conjuntos impares de nodos en una gráfica G, entonces siempre se cumple que

$$|X| \leq c(Q)$$

**Demostración.**

Por definición de Q, todo arco en X es cubierto por algún  $N_i \in Q$ , y como X es un acoplamiento a lo más hay un arco acoplado incidiendo en el nodo. Entonces, el número de arcos acoplados que pueden cubrir cada  $N_i \in Q$ , es a lo más, igual a su propia capacidad  $c(N_i)$ . Por lo tanto si sumamos las capacidades de todos los  $N_i$ , estaremos contando los arcos del acoplamiento X y además las posibles repeticiones con lo que este número al menos igual a  $|X|$ , es decir  $|X| \leq c(Q)$  ■

Una manera de demostrar el teorema de Edmonds consiste en construir un recubrimiento mínimo  $Q^*$ , a partir del acoplamiento  $X^*$  de máxima cardinalidad (obtenido en la sección anterior), es decir queremos

$$|X^*| = c(Q^*).$$

Recordemos que el algoritmo del acoplamiento máximo en gráficas no bipartitas avanza construyendo árboles alternantes, encogiendo flores, localizando trayectorias aumentantes y actualizando el acoplamiento hasta que los árboles obtenidos en la gráfica son húngaros, lo que significa que no se pueden hacer crecer y tampoco hay trayectorias aumentantes. Sea  $X^*$  este acoplamiento (i.e. es máximo y se puede obtener al aplicar el algoritmo de la sección anterior).

Para cualquier arista de  $G$ , al finalizar el algoritmo se cumple solo una de las siguientes condiciones:

- 1) La arista es incidente a un nodo T-etiquetado en un árbol.
- 2) La arista pertenece a una flor.
- 3) La arista es incidente a nodos sin etiquetar (están fuera de los árboles, probablemente forman la parte de la gráfica con actualización por trayectorias aumentantes más reciente, paso 2 del algoritmo).

Un ejemplo se muestra en la figura 3.21, para enseguida demostrar el teorema 3.4

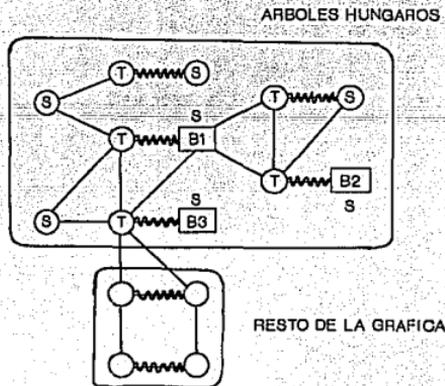


Figura 3.21

Demostración. (Del teorema 3.4)

Construiremos un recubrimiento de conjuntos impares de nodos  $Q^*$  en  $G$ , como sigue:

Cada nodo  $i$   $T$ -etiquetado será un conjunto  $N_i$  de capacidad 1 en  $Q^*$ . El número de  $T$ -nodos en el bosque húngaro es igual al número de aristas del acoplamiento  $X^*$  en dicho bosque.

Los nodos de cada flor  $B_j$  con  $2r+1$  nodos, forman un conjunto  $N_j$  de capacidad impar  $c(N_j)=2r+1$ , conteniendo  $r$  aristas de  $X^*$ .

Finalmente si hay  $n$  aristas con ambos extremos en nodos sin etiqueta, habrá  $2n$  de tales nodos y  $n$  aristas acoplados entre ellos. Si  $n=0$ , el recubrimiento  $Q^*$  está completo y tenemos  $|X^*|=c(Q^*)$ . Si  $n=1$ , eligiendo cualquiera de los extremos de la arista  $(p,q)$  como conjunto  $N_p$  de capacidad 1 y  $Q^*$  está así completo. Cuando  $n \geq 2$  elegimos alguno de los  $2n$  nodos como conjunto  $N_x$  de capacidad 1, y los restantes  $2n-1$  nodos forman un conjunto de capacidad impar conteniendo  $n-1$  aristas de  $X^*$ . Así se completa el recubrimiento  $Q^*$  que tiene la misma capacidad que  $|X^*|$ . Por la condición que demostramos en la primera parte  $X^*$  es maximal y  $Q^*$  es un recubrimiento de conjuntos impares de nodos de capacidad mínima en  $G$ . ■

### 3.4 PROBLEMA DEL ACOPLAMIENTO DE PESO MAXIMO PARA GRAFICAS NO BIPARTITAS

#### 3.4.1 FORMULACION DEL PROBLEMA DEL ACOPLAMIENTO DE PESO MAXIMO EN GRAFICAS NO BIPARTITAS COMO P.P.L.

En esta sección veremos la formulación del problema del acoplamiento de peso máximo en gráficas no dirigidas como un problema de programación lineal que nos ayudará en su resolución.

El problema del acoplamiento de peso máximo consiste en encontrar, de una gráfica ponderada dada  $G = (N, A, W)$  un acoplamiento tal que la suma de los pesos de las aristas sea máxima.

Al recordar el tratamiento de este problema en el caso de bigráficas, formulamos el siguiente problema de programación lineal:

$$\begin{aligned} \text{Max } F &= W X \\ \text{s. a.} \\ A X &\leq 1 & (P) \\ X &\geq 0 \end{aligned}$$

demonstrando que las soluciones del poliedro definido por (P) en el caso bipartito tiene solamente puntos extremos 0-1.

Sin embargo en el caso general, la presencia de ciclos impares hace que las soluciones del poliedro definido por (P) para gráficas no bipartitas ya no contenga solamente puntos extremos 0-1. Esto se postula en un teorema debido a Ballinski [2].

#### Teorema 3.6

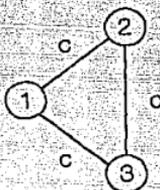
Todos los puntos extremos del poliedro definido por

$$\begin{aligned} AX &\leq 1 & \sum x_{ij} &\leq 1 \quad i \in N \\ x_{ij} &\geq 0 & j &\in \Gamma^+(i) & (i, j) \in A \end{aligned}$$

tiene coordenadas con valores 0, 1/2, 1.

La causa de los valores 1/2 de las coordenadas, como ya lo hemos mencionado, se debe a los ciclos de cardinalidad impar.

En la figura siguiente se muestra el ciclo impar más sencillo, con  $w_{ij} = c$ , donde  $c$  es una constante positiva para toda arista en  $G$ , la solución óptima al problema (P) es  $x_{ij} = 1/2 \forall (i,j) \in A$ , lo que claramente no es un acoplamiento.



El tratamiento para hacer del problema P un problema con soluciones enteras 0-1, se basa en el teorema general de dualidad - Teorema 3.5 - para el acoplamiento máximo, que introduce un conjunto de desigualdades, relacionadas con los conjuntos impares de nodos de  $G$ .

Sea  $R_k$  cualquier subconjunto impar de  $2r_k + 1$  nodos, entonces el acoplamiento máximo que se puede obtener en él es de cardinalidad  $r_k$ .

Así, definiendo la siguiente variable de decisión:

$$x_{ij} = \begin{cases} 1 & \text{Si el arco } (i,j) \text{ está en el acoplamiento} \\ 0 & \text{En otro caso.} \end{cases}$$

es claro, entonces que la desigualdad

$$\sum_{i \in R_k} \sum_{j \in R_k} x_{ij} \leq r_k$$

debe de ser satisfecha por cualquier acoplamiento. Representaremos el conjunto de todas las posibles restricciones de esta forma por

$$R x \leq r$$

donde  $R$  denota la matriz de incidencia de conjuntos impares de nodos contra arcos.

El vector  $r = (r_1, r_2, \dots, r_m)$  es tal que  $|R_k| = 2r_k + 1$  y  $r_k$  es la capacidad del conjunto  $R_k$ .

Procedamos a analizar el P.P.L. (P) , asociado con el problema del acoplamiento de peso máximo en gráficas generales, por medio de la teoría de dualidad.

$$\begin{aligned} \text{Max } f &= W \cdot X \\ \text{s.c.} \quad A \cdot x &\leq 1 & (P) \\ R \cdot x &\leq r \\ x &\geq 0 \end{aligned}$$

donde A es la matriz de incidencia de la gráfica.

W es el vector que representa el peso de los arcos.

### 3.4.2 DUALIDAD

Sabemos, del teorema de Edmonds, que el problema tiene solución entera cuando  $W = 1$ , pues en este caso se reduce al problema del acoplamiento máximo. Sin embargo no se ha establecido que se obtiene una solución entera en el caso más general. Esto se probará desarrollando un procedimiento que calcule una solución primal entera y una solución dual para cualquier conjunto de pesos en los arcos.

El problema dual (D) es

$$\begin{aligned} \text{Min } y &= \sum u_i + \sum r_k z_k \\ \text{s.c.} \quad A^t u + R^t z &\geq W \\ u, z &\geq 0 \end{aligned}$$

las variables duales  $u_i$  y  $z_k$  están asociadas con el nodo  $i$  y el conjunto impar de nodos  $R_k$ , respectivamente.

Las condiciones de ortogonalidad necesarias y suficientes para la optimalidad de las soluciones primal y dual están dadas por el teorema de holguras complementarias y son

$$x_{ij} > 0 \quad \Rightarrow \quad u_i + u_j + \sum_{R_k \ni (i,j)} z_k = w_{ij} \quad (1)$$

$$u_i > 0 \quad \Rightarrow \quad \sum_j x_{ij} = 1 \quad (2)$$

$$z_k > 0 \quad \Rightarrow \quad R_k \cdot x = r_k \quad (3)$$

Como en el caso bipartito, el procedimiento mantiene factibilidad primal y dual todo el tiempo, además mantiene satisfechas la mayoría de las condiciones de ortogonalidad excepto las condiciones (2). El número de tales condiciones no satisfechas decrece monótonamente durante el transcurso del cálculo. Este comienza con el acoplamiento inicial  $X = 0$  y con la solución dual factible

$$\begin{aligned} u_i &= W & \forall i \\ z_k &= 0 & \forall k \end{aligned}$$

donde  $W$  es suficientemente grande, por ejemplo

$$W = 0.5 \max \{ w_{ij} \}$$

Estas soluciones primal y dual satisfacen todas las condiciones (1) y (3) pero no algunas de las condiciones (2).

En el paso general del algoritmo tratamos de encontrar una trayectoria aumentante dentro de la subgráfica obtenida por la compresión de todas las flores  $k$  para las cuáles  $z_k > 0$  y de eliminar los arcos  $(i, j)$  si  $u_i + u_j + \sum z_k > w_{ij}$ .

Si se detecta una trayectoria aumentante, se extiende entre dos nodos expuestos  $i$  y  $j$  para los cuáles  $u_i = u_j > 0$ . Así, después de aumentar el acoplamiento al menos dos de las condiciones (2) llegaron a satisfacerse. Se realizan cambios en el acoplamiento dentro de cada flor comprimida tal que el acoplamiento continúa siendo maximal dentro de cada flor. Así, cada una de las condiciones (3) continúa siendo satisfecha después de cada iteración. Como la trayectoria aumentante involucra sólo arcos  $(i, j)$  para los cuáles  $u_i + u_j + \sum z_k = w_{ij}$  todas las condiciones (1) continúan siendo satisfechas.

Si no es posible agrandar los árboles, es debido a que todo arco  $(i, j)$  cumple que  $u_i + u_j + \sum z_k > w_{ij}$ , tratamos entonces de encontrar un valor apropiado  $\theta > 0$  para que al menos un arco  $(i, j)$  se pueda agregar a un árbol alternante, realizando los siguientes cambios en las variables duales. Para cada nodo  $i$  con una S-etiqueta y cada nodo  $i$  dentro de una flor más externa cuyo pseudonodo tenga una S-etiqueta, reste  $\theta$  de  $u_i$ .

Para cada nodo  $i$  con una T-etiqueta y para cada nodo  $i$  contenido en una flor más externa cuyo pseudonodo tenga una T-etiqueta, agregue  $\delta$  a  $u_i$ . Para cada flor más externa  $k$  cuyo pseudonodo tenga una S-etiqueta, sumamos  $2\delta$  a  $z_k$ , y para cada flor más externa  $k$  cuyo pseudonodo tenga una T-etiqueta, se resta  $2\delta$  de  $z_k$ .

Si un arco  $(i, j)$  está contenido en una flor, no hay efecto de todo esto en  $u_i + u_j + \sum z_k$  causado por los cambios en los valores duales. Pero si  $i$  tiene una S-etiqueta o  $i$  está contenida en una flor más externa cuyo pseudonodo tiene una S-etiqueta y  $j$  no está etiquetado, el efecto neto es  $-\delta$ .

Es de notar que en el caso del acoplamiento bipartito había dos restricciones para el valor máximo de  $\delta$ . Ahora por lo menos existen cuatro restricciones

- [4] Si  $i$  es un nodo S-etiquetado o está contenido en una flor más externa cuyo pseudonodo tiene una S-etiqueta, es requerido que

$$u_i - \delta \geq 0.$$

- [5] Si  $(i, j)$  es un arco tal que sus extremos  $i$  y  $j$  tienen S-etiquetas o sus arcos están contenidos en diferentes flores más externas cuyos pseudonodos tienen S-etiquetas, se requiere que

$$(u_i - \delta) + (u_j - \delta) = w_{ij}$$

- [6] Si el pseudonodo de una flor más externa tiene una T-etiqueta, entonces se requiere que

$$z_k - 2\delta \geq 0$$

- [7] Si  $(i, j)$  es un arco tal que  $i$  es un nodo con una S-etiqueta o está contenido en una flor más externa cuyo pseudonodo tiene una S-etiqueta, mientras que el nodo  $j$  es un nodo sin etiquetar o está contenido en una flor más externa cuyo pseudonodo está sin etiquetar, entonces se requiere que

$$(u_i - \delta) + u_j = w_{ij}$$

Supongamos que se elige  $\delta$  suficientemente grande, respetando a las condiciones [4] a [7]. Si la condición [4] está restringiendo a  $\delta$ , entonces la nueva solución dual es tal que todas las condiciones (2) son satisfechas. Tanto la solución primal como la solución dual son óptimas y un acoplamiento máximo ha sido obtenido. (Notemos que los valores iniciales fueron escogidos uniformes para  $u_i$  de aquí que el mínimo valor de  $u_i$  existe para cada nodo expuesto).

Si la condición [5] está restringiendo, entonces una trayectoria aumentante puede ser encontrada ó una nueva flor se forma.

Si la condición [6] está restringiendo, una flor más externa puede ser expandida.

Si la condición [7] está restringiendo, al menos una nueva arista puede ser agregada a uno de los árboles alternantes.

### 3.4.3 ALGORITMO DE SOLUCION.

Utilizando la información de la sección anterior podemos en este momento dar un primer esbozo del algoritmo a utilizarse para resolver el problema del acoplamiento de peso máximo.

## ALGORITMO DEL ACOPLAMIENTO DE PESO MÁXIMO.

### Paso 0 Inicio

Comenzar con el acoplamiento  $X = \emptyset$  y  $u_i = 0.5 \max \{w_{ij}\}$  como soluciones primal y dual respectivamente.

### Paso 1 Etiquetación

Enraizar un árbol alternante en cada nodo expuesto y proceder a construir árboles alternantes etiquetando solamente con aristas  $(i, j)$  para las cuales  $u_i + u_j + \sum z_k = w_{ij}$ .

Si una trayectoria aumentante es encontrada, ir al paso 2.

Si una flor ha sido formada, ir al paso 3.

Si los árboles son húngaros, ir al paso 4.

### Paso 2 Aumentación

Encontrar y reconstruir la trayectoria aumentante, a través de las flores encogidas. Aumentar el acoplamiento, remover todas las etiquetas de los nodos, pseudonodos y regresar al paso 1.

### Paso 3 Floraciones

Identificar y comprimir la flor en la gráfica. El pseudonodo representando a la flor recibe una S-etiqueta y su variable  $z$  es colocada en cero. Regresar al paso 1.

### Paso 4 Cambio en las variables duales

Determinar el valor máximo de  $\delta$ , acorde a las condiciones de [4] a [7] y realizar los cambios apropiados a las variables duales. Si la condición [4] esta restringiendo, alto; El acoplamiento y la solución dual son óptimas. De otra forma, expandir las flores más externas para las cuáles  $z_k = 0$  y regresar al paso 1.

Resolveremos un ejemplo del acoplamiento de peso máximo en gráficas generales, con la ayuda de este algoritmo.

Considere la gráfica ponderada de la figura 3.22, con  $|N| = 10$  y  $|A| = 16$

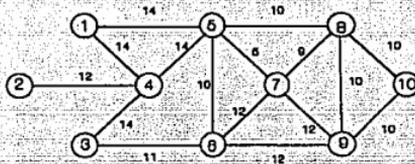


Figura 3.22

Iniciamos con el acoplamiento vacío  $X = \emptyset$ , los valores de las variables duales son

1	1	2	3	4	5	6	7	8	9	10
$u_1$	7	7	7	7	7	7	7	7	7	7

obtenidas de  $1/2 \max \{w_{ij}\} = .5(14) = 7$ . Existen en la gráfica 16 ciclos de cardinalidad impar, y sus variables  $Z_p$  son colocadas inicialmente en cero.

Formando árboles alternantes enraizados en nodos expuestos, con aristas  $(i,j)$  que cumplan  $u_i + u_j + z_p = w_{ij}$ , aumentamos el acoplamiento con las aristas  $\{(1,5), (3,4)\}$ , resultando que los árboles obtenidos en  $G$  son húngaros, como se observa en la figura 3.23

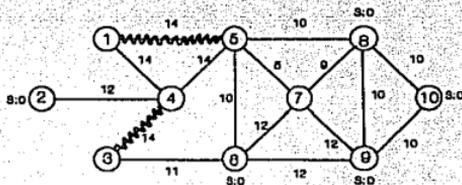


Figura 3.23

Necesitamos realizar la primera modificación a variables duales. El máximo valor  $\theta$  que obtenemos es  $\theta = 1$ , pues un valor mayor viola la condición [2] en la arista (6,7) porque  $u_i + u_j - w_{ij} \geq 2\theta \rightarrow 7+7-12 \geq 2\theta \rightarrow 1 \geq \theta$ . Realizando las modificaciones apropiadas tenemos

$u_i$	1	2	3	4	5	6	7	8	9	10
$u_j$	7	6	7	7	7	6	6	6	6	6

continuando con el algoritmo, construimos más árboles con las nuevas aristas (i,j) que cumplan  $u_i + u_j + z_p = w_{ij}$ , actualizando el acoplamiento agregamos la arista (6,7), obteniendo así el acoplamiento  $X = \{ (1,5), (3,4), (6,7) \}$ . Etiquetando a partir del nodo 9, detectamos la flor B1, de cardinalidad 3, formada por los nodos  $\{9,6,7\}$ , ver diagrama a de la figura 3.24; al contraer la flor, la gráfica resultante está formada por árboles húngaros y se muestra en el diagrama b de la figura 3.24.

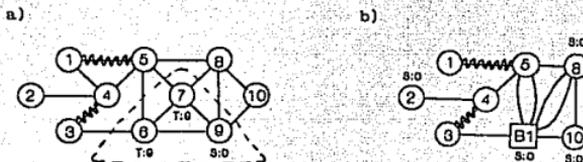


Figura 3.24

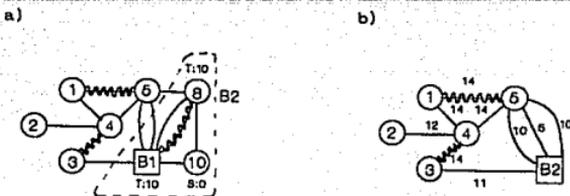


Figura 3.25

Modificamos por segunda ocasión las variables duales. El valor máximo  $\theta$ , es  $\theta = 1$ , porque la arista (3,6) está restringiéndolo. Realizando las modificaciones tenemos:

1	1	2	3	4	5	6	7	8	9	10
$u_1$	7	5	7	7	7	5	5	5	5	5

Como  $\theta = 1$ , el valor de  $Z_1$  asociado con el ciclo impar formado por las aristas con nodos en  $\{9,6,7\}$  toma el valor de  $Z_1 = 2$  y  $r_1 = -1$ , continuando con la construcción de árboles, encontramos que la arista (B1,8) con peso igual a 10 aumenta la cardinalidad del acoplamiento, como se aprecia en el diagrama a de la figura 3.25. Al realizar la asignación de etiquetas nuevamente, descubrimos la flor B2 formada por las aristas con nodos en  $\{B1,8,10\}$ , a partir del nodo 10; al comprimir B2 obtenemos la gráfica b de la figura 3.25, haciendo  $Z_2 = 0$  y  $r_2 = 2$ .

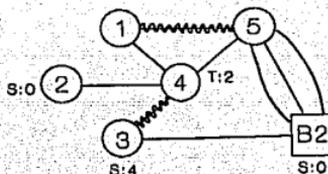


Figura 3.26

Al aplicar las etiquetas observamos que los árboles son húngaros, -ver figura 3.26-, así que buscamos modificar las variables duales nuevamente, obteniendo  $\theta = 1/2$  restringido por la arista (3,B2), los cambios en las variables duales nos lleva a

1	1	2	3	4	5	6	7	8	9	10
$u_1$	7	4.5	6.5	7.5	7	4.5	4.5	4.5	4.5	4.5

con  $Z_2 = 2(1/2) = 1$ . Al realizar la asignación de etiquetas, logramos etiquetar el nodo expuesto 2 por medio de una trayectoria aumentante que partió del nodo B2, lo que implica que podemos mejorar el acoplamiento, diagrama a de la figura 3.27, al modificar el acoplamiento obtenemos el diagrama b:

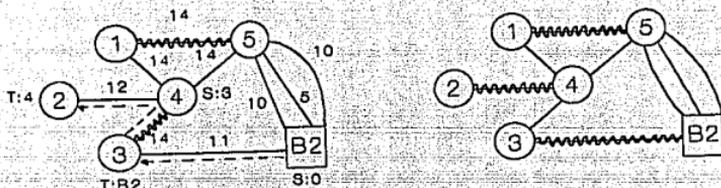


Figura 3.27

Como ningún nodo es expuesto, terminamos y el acoplamiento encontrado es de peso máximo. Al expandir los pseudonodos B2 y B3, actualizando el acoplamiento adecuadamente, obtenemos el acoplamiento final mostrado en la figura 3.28:

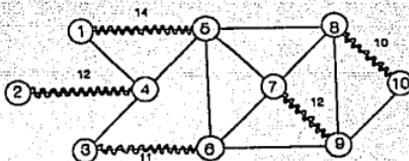


Figura 3.28

con el conjunto de aristas  $\{(1,5), (2,4), (3,6), (7,9), (8,10)\}$  y un peso de  $14+12+11+12+10 = 59$ . La solución dual debe, por lo tanto sumar igualmente 59. La suma de las  $u_i = 55$  y como encontramos 2 ciclos impares con  $Z_1 = 2$ ,  $r_1 = 1$  y  $Z_2 = 1$ ,  $r_2 = 2$  tenemos que  $\sum u_i + \sum r_p Z_p = 55 + 1(2) + 2(1) = 59$ , que es la solución óptima dual.

Si quisieramos implementar un programa computacional eficiente se tendrían que cuidar todavía muchos detalles y en la siguiente sección veremos algunos que propone Lawler [17].

### 3.4.4 ALGUNOS ASPECTOS NECESARIOS PARA IMPLEMENTAR UN PROGRAMA EFICIENTE QUE RESUELVA EL PROBLEMA DEL ACOPLAMIENTO DE PESO MÁXIMO.

#### *Tratamiento de flores*

El manejo de flores es mucho más complicado que en el caso del problema del acoplamiento máximo, pues no es suficiente registrar solamente las flores más externas porque cuando una flor es expandida, es necesario conocer cuáles flores consideradas están anidadas inmediatamente dentro de ella, para que así estas flores puedan ser consideradas de nuevo flores más externas. Cuando una aumentación ocurre, las flores con variables duales estrictamente positivas deben de ser mantenidas para su uso en la siguiente aplicación del procedimiento de etiquetación.

Por lo dicho, se debe mantener un registro explícito de todas las flores, sus nodos base y sus relaciones de andamio. Como antes, identificamos el nodo base  $b(i)$  de la flor más externa de la cual un nodo  $i$  forma parte. Las flores anidadas pueden tener el mismo nodo base (de aquí que las flores no pueden ser identificados únicamente por sus nodos base), pero dos flores más externas distintas no pueden tener el mismo nodo base. Llamaremos a un nodo  $i$  un nodo base si  $b(i) = i$ , aunque un nodo puede no estar contenido en una flor. De forma similar podemos hablar de "nodos contenidos en la misma flor más externa como  $i$ " aunque  $i$  no esté en una flor. En este caso, nos referimos solamente al nodo  $i$  en sí mismo.

En el problema del acoplamiento máximo solamente las flores más externas son de consecuencia y los nodos base de estas flores tienen únicamente S-etiquetas. Aquí existen 4 tipos distintos de flores:

- (1) Flores no etiquetadas, correspondiendo a pseudonodos no etiquetados. El nodo base no tiene etiqueta. La flor es más externa y su variable dual es estrictamente positiva.
- (2) S-flores, correspondiendo a pseudonodos S-etiquetados. El nodo base tiene una S-etiqueta, pero no una T-etiqueta. La flor es más externa y su variable dual puede ser positiva ó cero.

(3) T-flores, correspondiendo a pseudonodos T-etiquetados. El nodo base tiene una T-etiqueta, pero no una S-etiqueta. La flor es más externa y su variable dual es estrictamente positiva.

(4) Flores interiores, correspondiendo a pseudonodos comprimidos dentro de pseudonodos. El nodo base puede tener tanto una S-etiqueta como una T-etiqueta y las variables duales son estrictamente positivas.

Si queremos etiquetar una T-flor, debemos realizar lo siguiente: supongamos que la S-etiqueta en el nodo  $i$  es revisado y un arco  $(i, j) \in X$  es encontrado, donde  $u_i + u_j - w_{ij} = 0$ ,  $b(j) \neq b(i)$ ,  $b(j) \neq j$  y  $b(j)$  no está etiquetado. En esta situación,  $j$  está contenido en una flor no etiquetada que puede recibir una T-etiqueta de  $i$ . Por lo tanto aplicamos la etiqueta 'T :  $i, j$ ' a  $b(j)$ .

Para facilitar la descripción, colocaremos la etiqueta 'T :  $i, j$ ' a  $b(j)$  siempre que  $b(j) = j$ . Como no afecta, el segundo índice será ignorado en la recuperación de las trayectorias.

#### *Corrección de etiquetas después de una aumentación*

Después de cada aumentación, las flores deben ser retenidas para usarse en la siguiente aplicación de la técnica de etiquetación. También quisieramos retener las etiquetas en los nodos dentro de las flores. Pero las etiquetas en los nodos a través de las cuales la trayectoria aumentante pasó deben de ser corregidas, porque ya no son válidas, dado que las aristas en las flores varían, a pesar de permanecer igual en cardinalidad. Debemos entonces, cuidar los siguientes tres aspectos :

- 1) Retener las flores para su uso en el siguiente proceso de etiquetación.
- 2) Conservar todas las etiquetas de los nodos de las flores.
- 3) Corregir las etiquetas de los nodos que forman parte de la trayectoria aumentante.

Esta tarea se realizará como sigue, primero identificamos todas las flores (no solamente las más externas) a través de las cuales la trayectoria aumentante pasa. Para cada flor, encuentre su nuevo nodo base. (La trayectoria aumentante se extiende entre el viejo nodo base y el nuevo nodo base de cada flor por la que cruza).

Para todos los nodos en la trayectoria aumentante que no tienen nuevos nodos base ni viejos nodos base, simplemente intercambie los índices de las etiquetas. Esto es, si las etiquetas en tales nodos son 'S : i' y 'T : j' las nuevas etiquetas son 'S : j' y 'T : i'.

Para cada nodo b que es un nuevo nodo base, encuentre la flor más interna en la cual este contenida y el viejo nodo base b' de esta flor. Encuentre aristas (b,i) y (b',j) de la trayectoria aumentante donde i,j no están contenidas en la flor. Las nuevas etiquetas para b son 'S : i' y 'T : j, b'.

Para un nodo b' que sea un viejo nodo base, encuentre la flor más interna en la que este contenida y el nuevo nodo base b de esta flor. Si  $b = b'$ , simplemente intercambie los índices de las etiquetas en b'. De otra forma, realice un rastreo regresivo de la vieja T-etiqueta en b para descubrir una arista (b',j)  $\in X$ , donde j está en la flor, incidente a b'. Las nuevas etiquetas para b son 'S : i' y 'T : j'.

Veamos la corrección de las etiquetas después de un aumento en la figura 3.29; las aristas onduladas forman parte del acoplamiento y se ha encontrado una trayectoria aumentante entre los nodos 1 y 18. Las flores existentes son B1, que tiene a 5 como su nodo base, i.e.  $b'(1) = 5$ , B2 con  $b'(2) = 14$  y B3 con  $b'(3) = 3$ .

Una vez aumentado el acoplamiento y corregidas las etiquetas, los nuevos nodos-base en las flores B1, B2 y B3 son respectivamente  $b(1) = 9$ ,  $b(2) = 12$  y  $b(3) = 15$ . En la figura 3.30 se aprecian las correcciones de etiquetas en los nodos que forman parte de la trayectoria aumentante.

Como los nodos 4,6,7,8,10 y 13 no son ni nuevos ni viejos nodos-base, solamente intercambiamos índices en las etiquetas (por ejemplo el nodo 13 aparece con 'T : 14 y S : 12' antes del aumento y con 'T : 12 y S : 14' después). Para el nuevo nodo-base  $b1 = 9$ , la flor más interna que lo contiene es B1, la arista  $(b(1),1) = (9,10)$  y

la arista  $(b'(1), j) = (5, 4)$  con  $i, j$  fuera de  $B_1$ , tenemos que para  $b(1)$  las etiquetas son 'S : 10 y T : 4,5', procediendo de forma análoga aplicamos etiquetas para  $b_2$  y  $b_3$ . Con el viejo nodo-base  $b'(3)=3$ , la flor más interna que lo contiene es  $B_3$ , realizando el rastreo regresivo de la vieja T-etiqueta en el nuevo nodo-base  $b(3) = 15$ , encontramos la arista no acoplada  $(b'(3), j) = (3, 15)$  con el nodo 15 en  $B_3$ ; además la arista  $(b'(3), i)$  de la trayectoria aumentante con 'i' en  $B_3$  que incide en  $b'(3)$   $(b'(3), i) = (3, 4)$ , con lo que las etiquetas en  $b'(3)$  son 'S : 4 y T : 15'. Aplicamos de manera similar las etiquetas en  $b'(1)$  y  $b'(2)$

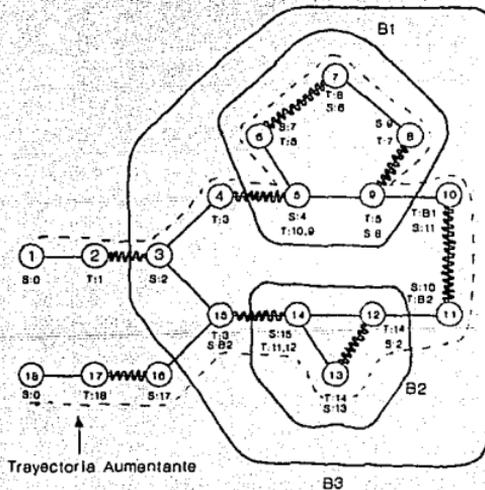


Figura 3.29

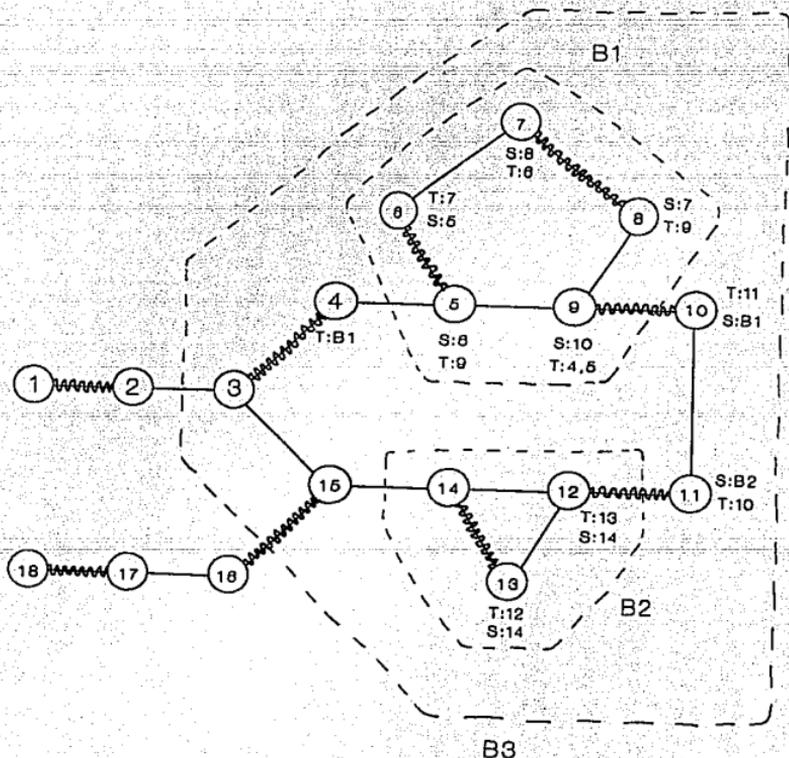


Figura 3.30

### 3.5 SOLUCION AL PROBLEMA DEL CARTERO CHINO

En esta sección daremos un algoritmo de solución al siguiente problema: Supongamos que un cartero parte de la oficina postal de una pequeña ciudad y debe recorrer, entregando correspondencia, todas las calles de la ciudad para regresar a la oficina de donde partió. ¿Cuál es la ruta que debe seguir para minimizar la distancia total recorrida?

Este problema fue originalmente propuesto por un matemático chino llamado Mei-ko Kwan, y por esta razón Edmonds lo llamó el problema del cartero chino.

Para modelar este problema en una gráfica  $G = (N, A)$  el conjunto de nodos representará las intersecciones de las calles y los arcos, las calles.

Observemos que si todas las intersecciones de las calles tienen un número par de estas, i.e. que en la gráfica correspondiente  $G$  el grado de todo nodo sea par entonces  $G$  es Euleriana, esto se debe al siguiente teorema que se enuncia sin demostración.

#### Teorema 3.6

Sea  $G = (N, A)$  una gráfica conexa no dirigida;  $G$  contiene un circuito Euleriano si y sólo si el número de nodos con grado impar es cero.

Por lo tanto el cartero puede seguir un circuito Euleriano y pasar por cada arista una sola vez. Una regla muy sencilla para encontrar un circuito Euleriano en una gráfica no dirigida fue dado por Fleury y es la siguiente:

Regla para encontrar un circuito Euleriano.

Iniciar de cualquier vértice  $i$  de la gráfica, cada vez que una arista ha sido atravesada borrarla de la gráfica. Nunca atravesar una arista si cuando remueva esa arista la gráfica se divide en dos componentes conexas.

Como el circuito pasa por cada arista una sola vez, esta ruta es óptima. En este caso la longitud mínima de un circuito es igual al número de aristas en la gráfica.

Supongamos ahora que la gráfica  $G$  es no Euleriana, el método de solución que propuso Mei-Ko Kwan fue comenzar con cualquier solución factible, descubrir ciclos negativos con ciertas aristas y realizar modificaciones en sus pesos tratando de llegar a la solución óptima.

Consideremos la gráfica mostrada en la figura 3.31, con 4 nodos de grado impar: 1,2,4,5. Según Kwan un conjunto factible de arcos para duplicar es  $\{(1,2)$  y  $(4,5)\}$ . Construyamos una nueva red, con los mismos nodos pero a cada arista duplicada le asignamos un peso de  $-w_{ij}$ . Si esta red contiene un ciclo negativo lo utilizaremos para encontrar la solución. Todo lo que tenemos que hacer es duplicar aquellas aristas que están en el ciclo sin duplicar y volver sencillas aquellas que estaban duplicadas, así las aristas duplicadas tienen longitud mínima si y sólo si no contienen ciclos negativos. El problema es que Kwan no menciona como detectar ciclos negativos en una gráfica no dirigida, porque el algoritmo de rutas más cortas no se aplica a redes no dirigidas con aristas que tengan pesos negativos.



Figura 3.31

Enseguida veremos la solución que propone Edmonds, que resuelve, como subrutinas, un problema de rutas más cortas y un problema de acoplamiento perfecto de peso mínimo.

Consideremos un Circuito del Cartero Chino (CCCh), i.e. un circuito que utilice todas las aristas. Reemplacemos toda arista por un conjunto de copias paralelas de ellas igual en cardinalidad al número de veces que el CCCh usa esa arista. Demostraremos que esta multigráfica contiene un circuito Euleriano. Recíprocamente si una gráfica Euleriana se obtiene de  $G$  multiplicando aristas, entonces una trayectoria Euleriana en esta gráfica da lugar a un CCCh en la gráfica original.

Primero demostraremos el siguiente resultado que se utilizará en la solución del problema.

Teorema 3.7

En una gráfica  $G=(N,A)$  el número de nodos con grado impar es par.

Demostración

Considere la gráfica no dirigida  $G = (N,A)$ , algunos de los nodos de  $N$ , digamos los que se encuentran en el conjunto  $N^+$  tienen grado par y algunos  $N^- = N - N^+$  tienen grado impar. Ahora la suma de los grados  $g(i)$  de todos los vértices  $x_i \in N$  es igual a 2 veces el número de aristas de  $A$  y por tanto obtenemos un número par  $2|A| = 2m$ . De esta observación

$$\sum_{i \in N} g(i) = \sum_{i \in N^+} g(i) + \sum_{i \in N^-} g(i) = 2m$$

y como  $\sum_{i \in N^+} g(i)$  es par  $\rightarrow \sum_{i \in N^-} g(i)$  también es par, lo que significa que como

todo  $g(i)$  en la última expresión es impar, el número  $|N^-|$  de vértices de grado impar es par. ■

Sea  $M$  un conjunto de trayectorias de  $G$  (digamos  $\mu_{ij}$ ) entre los vértices finales  $i, j \in N^-$ , tales que 2 trayectorias no tengan algún extremo en común, es decir las trayectorias son entre pares de vértices ajenos de  $N^-$  y constituyen un acoplamiento de parejas de vértices. El número de tales trayectorias  $\mu_{ij}$  en  $M$  es  $1/2 |N^-|$  y como se demostró que  $|N^-|$  es siempre par, el número de trayectorias siempre es entero. Suponga ahora que todas las aristas formando una trayectoria  $\mu_{ij}$  son añadidas a  $G$  como arcos artificiales en paralelo a las aristas de  $G$  existentes. (En primera instancia, esto significa que todas las aristas de  $G$  que forman  $\mu_{ij}$  son duplicadas). Esto se realiza para toda trayectoria  $\mu_{ij} \in M$  y la multigráfica resultante la denotamos  $G^-(M)$ . Como algunas aristas de  $G$  pueden aparecer en más de una trayectoria  $\mu_{ij}$  algunas aristas de  $G^-(M)$  pueden, después de que todas las aristas de las trayectorias  $\mu_{ij}$  han sido añadidas, estar triplicadas, cuadruplicadas, etc.

Podemos enunciar entonces el siguiente teorema:

### Teorema 3.8

Para cualquier circuito recorriendo  $G$ , existe alguna elección de  $M$  para la cual  $G^-(M)$  posee un circuito Euleriano correspondiente al circuito de  $G$ . La correspondencia es tal que si un circuito recorre una arista  $(i, j)$  de  $G$   $J$  veces, existen  $J$  aristas (una real y  $(J-1)$  artificiales) entre  $i$  y  $j$  en  $G^-(M)$ , cada una de las cuales es utilizada exactamente una vez por el circuito Euleriano de  $G^-(M)$  y reciprocamente.

### Demostración

Si un circuito recorre  $G$ , entonces al menos una arista incidente a todo vértice  $i$  de grado impar debe ser recorrida 2 veces, (una arista recorrida dos veces puede ser considerada como dos aristas paralelas, una real y la otra artificial con ambas recorridas una sola vez). Sea  $(i, k)$  esta arista. Si el grado  $g(k)$  en  $G$  es impar entonces la adición anterior de una arista artificial hará  $g(k)$  par y esta arista debe solamente ser recorrida dos veces hasta que  $i$  y  $k$  son alcanzados.

Sin embargo, si  $g(k)$  es par entonces la adición de la arista artificial hará ahora  $g(k)$  impar y una segunda arista surgida de  $k$  debe también ser recorrida dos veces (i.e. otra arista ficticia se agregó). El argumento continúa de  $k$  hasta que un vértice de grado impar es alcanzado, como ya se mencionó. Así para satisfacer la condición de recorrer  $i$ , una trayectoria total de  $i$  a algún otro vértice  $r$  de  $N^-$  debe ser recorrida dos veces. Esto automáticamente satisface la condición de atravesar el vértice  $r$ . Similarmente para todos los otros vértices  $i$  de  $N^-$ ; lo que significa que un conjunto  $M$  de rutas de  $G$ , como se definieron, deben ser recorridas 2 veces, y por lo tanto como toda arista de  $G^-(M)$  debe ser recorrida una vez el teorema se sigue. ■

Basado en este teorema obtenemos un algoritmo muy fácilmente: es necesario encontrar el conjunto de trayectorias  $M^*$  (acopiando vértices de grado impar) que produce el mínimo costo adicional. Por esta razón el costo mínimo de un CCCh en  $G$  debe ser igual a la suma de los costos de las aristas de  $G$  más la suma de los costos de las aristas en las trayectorias de  $M^*$ .

## Algoritmo de solución al problema del cartero chino

### Paso 1

Sea  $w_{ij}$  el peso del arco  $(i, j)$  en  $G$ . Identificar los nodos de grado impar en la gráfica  $G$ . Si no hay ninguno ir al paso 4.1.

### Paso 2

Encontrar las rutas más cortas entre todo par de vértices de grado impar.

### Paso 3

Encontrar el acoplamiento perfecto de peso mínimo  $M^*$  de los vértices en  $N^*$ . Ir a 4.2

### Paso 4

4.1 Construir un circuito Euleriano en  $G$ .

4.2 Construir la multigráfica  $G^*(M^*)$  y un circuito Euleriano en  $G^*$ . la suma de los costos de los arcos en el circuito de  $G^*(M^*)$  es el costo mínimo de un TCCh. (tomando los costos de arcos artificiales como los de arcos reales).

Es claro, entonces que un CCCh óptimo no usa algún arco más de dos veces, puesto que si lo hiciera, entonces en la gráfica Euleriana correspondiente, dos copias de esta arista pueden ser removidas, pero entonces una trayectoria Euleriana en la gráfica resultante debe corresponder a una CCCh más corto en la  $G$  original contradiciendo la optimalidad.

Así el problema del cartero chino es equivalente a lo siguiente: Dada una gráfica  $G$ , encontrar el número mínimo de arcos que al duplicarlos resulta una gráfica Euleriana.

Veamos la aplicación del algoritmo al siguiente problema :

Dada la gráfica de la figura 3.32 encontrar un circuito Euleriano, tal que la distancia total recorrida sea mínima.

Los nodos con grado impar son el conjunto  $N^* = \{1, 5, 6, 7\}$ .

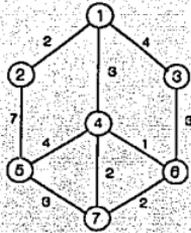


Figura 3.32

Utilizando el algoritmo de Floyd [7] para encontrar las rutas más cortas entre todos los nodos de  $N$ , obtenemos el acoplamiento perfecto de peso mínimo  $M^* = \{ (1,5), (6,7) \}$  que se muestran respectivamente en los diagramas a) y b) de la figura 3.33 .



Figura 3.33

Finalmente al aplicar la regla de Fleury a partir del nodo 1, en la multigráfica  $G(M^*)$  de la figura 3.34, podemos obtener el siguiente circuito Euleriano 1, 4, 6, 7, 4, 5, 4, 1, 3, 6, 7, 2, 1 con un costo mínimo de 40:

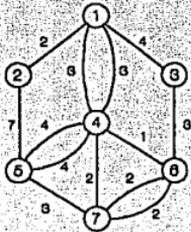


Figura 3.34

## CAPITULO 4

En este capítulo se describen algunos aspectos relacionados con el paquete computacional que forma parte de este trabajo de tesis. En la primera sección se tratan aspectos de la elección de una estructura de datos adecuada para representar las gráficas; en la segunda sección se describen la estructura general del programa y los procedimientos de mayor relevancia dentro del programa, posteriormente se da el manual para el usuario donde se brinda la información necesaria para un aprovechamiento óptimo del paquete. Por último se muestran algunos ejemplos que se resolvieron utilizando este paquete, así como las conclusiones.

### 4.1 ELECCION DE LA ESTRUCTURA DE DATOS PARA EL PROGRAMA.

Una de las cuestiones más importantes para lograr un programa eficiente en cuanto a espacio de memoria y velocidad, es la representación que tendrá, dentro de la computadora, una gráfica.

Como este trabajo, en su mayor parte trata de gráficas no dirigidas, empleamos las dos representaciones más utilizadas para representar gráficas no dirigidas y que son la matriz de adyacencia y las listas de adyacencia.

La matriz de adyacencia para una gráfica  $G = (N, A)$  con  $|N| = n$  y  $|A| = m$  se definió en el primer capítulo como una matriz de dimensión  $n \times n$  con elementos uno en la entrada  $(i, j)$  si en la gráfica se encuentra la arista entre el nodo  $i$  y el nodo  $j$ , y cero en otro caso.

La desventaja en el uso de la matriz de adyacencia es que se desperdicia la mitad del espacio de la matriz, pues por construcción la matriz es simétrica para gráficas no dirigidas. Si es poco densa (i.e. el número de aristas es pequeño), el desperdicio es mayor ya que la matriz contendrá ceros en la mayoría de sus entradas. Por otro lado, entre las ventajas de su uso, es que para gráficas bipartitas aprovechando la proposición 1.2, se puede reducir el espacio perdido al utilizar sólo la submatriz de los nodos en el conjunto  $S$  contra los nodos en el conjunto  $T$  de la bipartición.

Otra ventaja en el uso de la matriz de incidencia para el caso de gráficas bipartitas es que facilita la programación y favorece la rápida ejecución del programa.

Una alternativa para el caso de gráficas poco densas o gráficas generales es la representación por medio de listas de adyacencia, esto es, asignando a cada vértice la lista de los nodos adyacentes a él, pero cuando se trata de una gráfica densa, el volumen de memoria es casi el mismo que utilizando la matriz de incidencia; en este caso la eficiencia del programa dependerá del tiempo de ejecución que es mayor con esta implementación.

Dado que el interés en el desarrollo de esta primera versión del paquete es didáctico, y no era el de construir gráficas demasiado grandes, se decidió utilizar la implementación de la matriz de adyacencia para los problemas de acoplamiento en gráficas bipartitas, limitando el número de nodos a 20, diez en cada subconjunto de la bipartición, para desarrollar además una representación en modo gráfico de la solución del problema del acoplamiento máximo en bigráficas, así como su problema dual: la cubierta de nodos de cardinalidad mínima. También se presenta en modo gráfico la solución al problema del acoplamiento de peso máximo en bigráficas ponderadas.

Para el caso general del problema del acoplamiento máximo se decidió utilizar la representación de listas de adyacencia, para hacer más eficiente la rapidez de ejecución, además de que la presencia de ciclos de cardinalidad impar y el desperdicio de memoria al utilizar la matriz de adyacencia hace ineficiente utilizar esta última representación, para la resolución de este problema.

#### 4.2 ESTRUCTURA DEL PROGRAMA

A lo largo de este trabajo se plantean y desarrollan algoritmos para la solución de varios problemas de la teoría del acoplamiento, se decidió realizar un paquete computacional que resolviera algunos de estos problemas, proporcionando las ideas generales para la construcción de los demás casos.

Para su desarrollo se utilizó el lenguaje de programación estructurado denominado Turbo-Pascal en su versión 6.0.

Enseguida se presenta el diseño del sistema a través de diagramas de bloque que muestran la relación entre los procedimientos principales, así como las condiciones necesarias que se deben cumplir para que sean realizados.

#### Procedimiento LECTURA

El procedimiento Lectura varía según el problema a resolver, así en la opción de PRESENTACION DE UN EJEMPLO GUARDADO EN MEMORIA para cada problema Lectura, lee del disco los datos de una gráfica bipartita, de una gráfica bipartita ponderada, de una gráfica general o las matrices de preferencia para un acoplamiento de Gale-Shapley, según sea el caso.

Cuando la opción es SOLUCION A UN EJEMPLO DADO POR EL USUARIO, pide los datos necesarios en cada caso. En el problema del acoplamiento máximo en bigráficas, pide el número de nodos en cada conjunto de la bipartición, las aristas de la gráfica y si el usuario desea un acoplamiento inicial.

#### Procedimiento VERIFICA

Verifica despliega diferentes mensajes de error en pantalla, que obliga al usuario a proporcionar datos válidos. Este procedimiento es utilizado sólo cuando estamos en la opción *Resolver un problema dado por el usuario*.

#### Procedimiento EJECUTA

Realiza el cálculo del acoplamiento óptimo según sea el caso.

#### Procedimiento RESULTADOS

Presenta la solución a los problemas. Así, por ejemplo en el caso del acoplamiento de peso máximo en bigráficas presenta las matrices de incidencia de nodos en S contra nodos en T de la gráfica utilizada y del acoplamiento obtenido; el valor del acoplamiento óptimo, los valores de las variables de la solución dual y la representación en modo gráfico.

Para el caso del problema del acoplamiento máximo en gráficas no bipartitas se presenta el acoplamiento final obtenido y el número de nodos expuestos al finalizar el algoritmo.

ELECCION

PROBLEMA DEL ACOPLAMIENTO DE CARDINALIDAD MAXIMA EN GRAFICAS BIPARTITAS.

PROBLEMA DEL ACOPLAMIENTO DE PESO MAXIMO EN GRAFICAS BIPARTITAS.

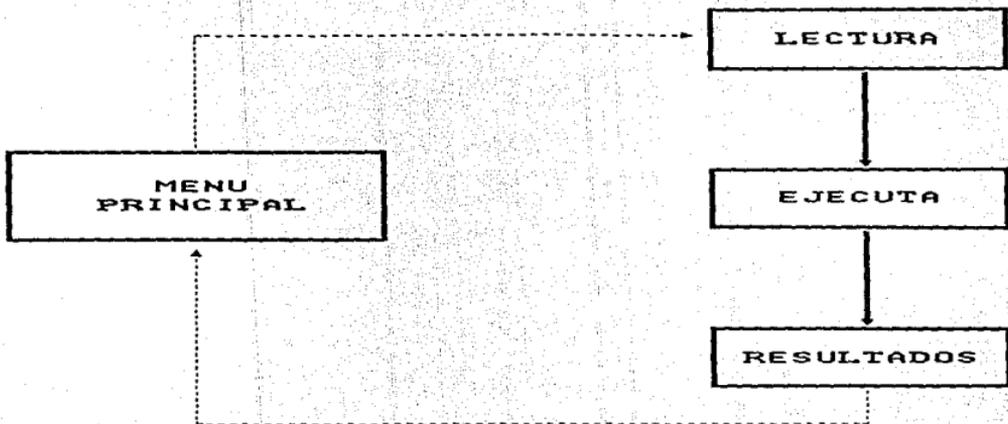
PROBLEMA DEL ACOPLAMIENTO DE MAXIMA CARDINALIDAD EN GRAFICAS GENERALES.

PROBLEMA DEL ACOPLAMIENTO DE GALE-SHAPLEY.

FIN.

SOLUCION A UN PROBLEMA DADO POR EL USUARIO.

EJEMPLO DE SOLUCION A UN PROBLEMA GUARDADO EN MEMORIA



#### 4.2.1 DEFINICION DE CONSTANTES Y VARIABLES

Se mencionan en esta sección las constantes y variables más relevantes, utilizadas en el paquete computacional. Como este paquete resuelve diferentes tipos de problemas de acoplamiento, se mencionarán en las siguientes subsecciones a que parte del programa pertenecen.

##### 4.2.1.1 PRINCIPALES CONSTANTES Y VARIABLES UTILIZADAS PARA RESOLVER LOS PROBLEMAS DE ACOPLAMIENTO EN GRAFICAS BIPARTITAS.

Las principales constantes utilizadas en bigráficas son :

MaxS = 10

MaxT = 2 \* MaxS

Debido a que este programa se elaboró con fines didácticos y en él se presentan en modo gráfico los acoplamientos y las gráficas de los problemas, el número máximo de nodos es de 20, siendo 10 el máximo para cada uno de los conjuntos que forman parte de la bipartición.

Las principales variables utilizadas son:

De ENTRADA

n, p de tipo entero

Guardan la cardinalidad de los conjuntos S y T de la bipartición de la bigráfica, respectivamente.

G, W de tipo Matriz

Donde Matriz es un arreglo de enteros bidimensional de tamaño n x p.

G Guarda la matriz de incidencia de la gráfica del problema.

W Guarda los pesos de los arcos de la gráfica, si es el caso.

De SALIDA

A de tipo Matriz

A guarda el acoplamiento que se va obteniendo en cada proceso del programa.

B del tipo ConjT

Donde ConjT es un conjunto de enteros de 1 hasta n \* p

B contiene a los nodos que forman parte de la cubierta mínima de nodos.

## De TRABAJO

**F** del tipo Arraum

Donde Arraum es un arreglo de enteros de dimensión  $1 * 2n$ .

$F(i) = j$  indica que el nodo  $i$  se puede alcanzar desde  $j$ .

**MA** del tipo Arraco

Donde Arraco es un arreglo de enteros de dimensión  $1 * n$ .

$MA(i) = j$  indica el nodo final  $j$  del arco  $(i, j)$  que pertenece al acoplamiento. Cuando  $MA(i) = 0$ , esto indica que el nodo es expuesto.

**AUX** de tipo matriz

Aux Ayuda en la reconstrucción de una trayectoria aumentante.

**DA, RA, WW, V, TC** del tipo ConjT

Donde ConjT es un conjunto de enteros de 1 hasta  $n+p$

DA es el dominio del acoplamiento.

RA es el rango del acoplamiento.

WW contiene al conjunto de nodos expuestos de S.

V contiene al conjunto de nodos expuestos de T.

TC contiene a los nodos de T.

**SC** del tipo ConjS

Donde ConjS es un conjunto de enteros de 1 hasta  $n$ .

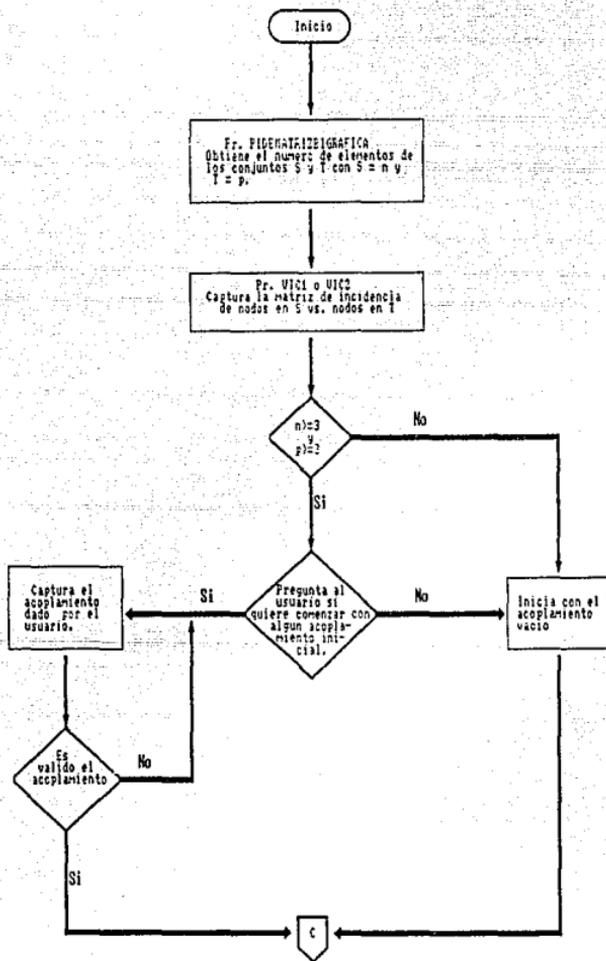
SC contiene a los nodos de S.

**Posición\_encontrada** de tipo lógico.

Se inicializa con falso y permanece en ese estado hasta que una trayectoria aumentante es encontrada.

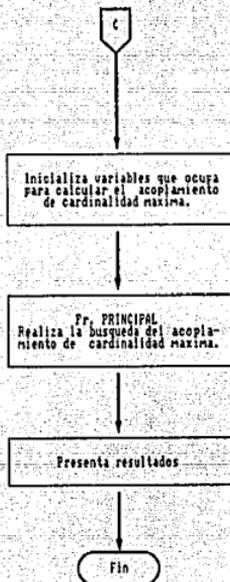
Enseguida se muestran los diagramas de flujo de los programas que resuelven los problemas del acoplamiento de máxima cardinalidad y del acoplamiento de peso máximo en gráficas bipartitas, junto con los diagramas que muestran el caso de la presentación del ejemplo guardado en memoria. Al compararlos vemos que la única modificación se encuentra en como lee los datos de entrada.

Programa que obtiene el acoplamiento de máxima cardinalidad en gráficas bipartitas.



CONTINUA ...

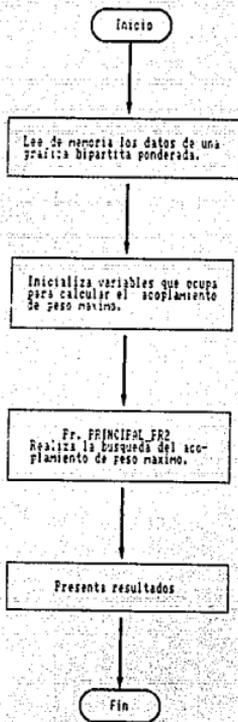
Programa que obtiene el acoplamiento de máxima cardinalidad en gráficas bipartitas.



Programa que obtiene el acoplamiento de peso máximo en gráficas bipartitas.



# Programa Ejemplo del acoplamiento de peso máximo en gráficas bipartitas.



#### 4.2.1.2 PRINCIPALES CONSTANTES Y VARIABLES UTILIZADAS PARA RESOLVER LOS PROBLEMAS DE ACOPLAMIENTO EN GRAFICAS GENERALES.

Las variables utilizadas en la solución del problema del acoplamiento de cardinalidad máxima en gráficas generales son las siguientes:

Las constantes utilizadas son:

$M1 = n+1$ .  $M2 = m+n$ .

Utilizados para la representación tabular de las listas de adyacencia.

Maxnodo = 40.

Máximo número de nodos de la gráfica.

Maxarco = 70.

Máximo número de arcos de la gráfica.

Las variables utilizadas son

De Entrada

$N, M$  de tipo entero

$n$  indica el número de nodos de la gráfica.

$m$  indica el número de arcos de la gráfica.

$Inodo, Jnodo$  vectores de tamaño  $1 \times m$ .

$Inodo(k)$  y  $Jnodo(k)$  son los nodos  $(i, j)$  del  $k$ -ésimo arco, de la gráfica dada.

De Salida

$Par$  Es un vector entero de longitud  $1 \times n$ .

El nodo  $i$  es acoplado con  $Par(i)$ ,  $i = 1, \dots, n$ ;

Si  $Par(i) = 0$ , entonces el nodo  $i$  es nodo expuesto.

$Noacop$  de tipo entero.

Indica el número de nodos no acoplados.

Dentro de los procedimientos tenemos las siguientes variables:

**Fwdarc** Vector entero de longitud  $1 * M2$ .

**Fwdarc(i)** es el nodo final del  $i$ -ésimo arco en la representación de las listas de adyacencia de la gráfica.

**Arcfir** Vector entero de longitud  $1 \times N1$ .

**Arcfir(i)** es el apuntador al número del primer arco con nodo inicial  $i$  en la lista de adyacencia del nodo  $i$  en la gráfica.

**Antes, Cola** Vector entero de tamaño  $1 \times n$ .

**Antes(i)** es el nodo-raíz del árbol al que pertenece el nodo  $i$ .

**Cola** es la cola de nodos.

**Outree** Vector de elementos lógicos.

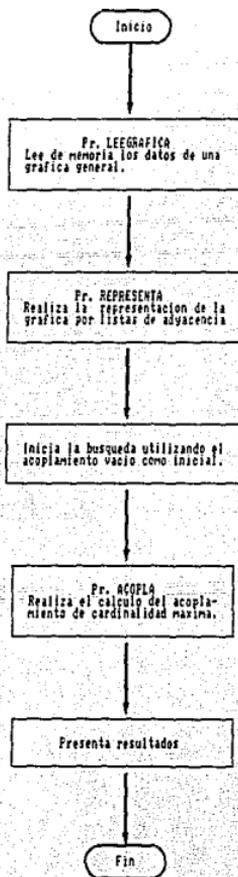
**Outree(i)** indica si el nodo  $i$  está en el árbol.

Enseguida se muestran los diagramas de flujo para los dos alternativas que se pueden resolver con este programa.

Programa que obtiene el acoplamiento de máxima cardinalidad en gráficas generales.



# Programa Ejemplo del acoplamiento de máxima cardinalidad en gráficas generales.



#### 4.2.1.3 PRINCIPALES CONSTANTES Y VARIABLES UTILIZADAS PARA RESOLVER EL PROBLEMA DEL ACOPLAMIENTO DE GALE-SHAPLEY.

Las constantes y variables utilizadas en la solución del acoplamiento de Gale-Shapley son las siguientes:

Las constantes son:

Maxvalor = 10

Restringe el máximo número de casos.

Las variables son las siguientes:

De Entrada

N de tipo entero.

Indica el número de casos y debe ser mayor que 1 y menor que la constante Maxvalor.

MatrizH, MatrizM Arreglos enteros rectangulares de tamaño nxn.

Son las matrices de preferencias de las hombres (mujeres), con respecto a las mujeres (hombres), respectivamente.

De Salida

Prefer Arreglo entero de tamaño 1xn.

Prefer(i) indica la preferencia actual del hombre i.

Rechazado Arreglo lógico de tamaño 1xn.

Rechazado(i) indica si la propuesta del hombre i fue rechazada.

En el procedimiento ACOPLA\_GALE, tenemos las siguientes variables:

Lugar\_O, Acop, Ocup Arreglos enteros de tamaño 1xn.

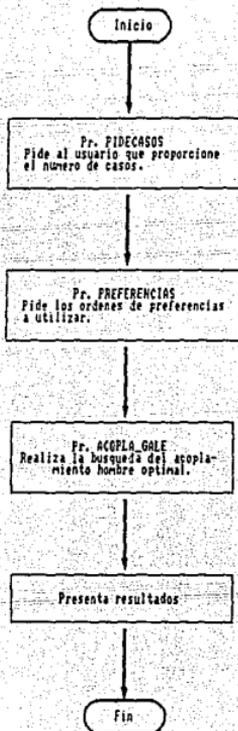
Lugar\_O(i) indica el número de preferencia del hombre i.

Acop(i) indica la pareja de la mujer i.

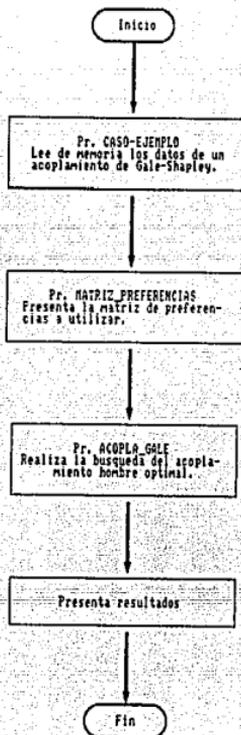
Ocup(i) indica quien esta en la lista de espera de la mujer i.

Enseguida se muestran los diagramas de flujo de los programas que resuelven los problemas del acoplamiento de Gale-Shapley.

# Programa que obtiene el acoplamiento de Gale-Shapley.



# Programa Ejemplo del acoplamiento de Gale-Shapley.



#### 4.2.2 SUBROUTINAS

En esta sección se comentan y analizan los procedimientos más importantes que se utilizan como subrutinas para resolver los problemas que se tratan en este paquete computacional.

Para cada subrutina se ofrece una breve descripción y su correspondiente diagrama de flujo la ilustra.

##### Subrutina PRINCIPAL

Esta subrutina es la parte más importante de los programas que resuelven el problema del acoplamiento de cardinalidad máxima en gráficas bipartitas porque realiza la búsqueda de trayectorias aumentantes relativas al acoplamiento en cada iteración.

Comienza con algún acoplamiento inicial factible (puede ser el vacío). Busca un nodo expuesto en un conjunto de la bipartición, si no lo encuentra el acoplamiento es óptimo y si lo encuentra construye trayectorias alternantes, a partir de ese nodo, hasta encontrar una trayectoria aumentante o tener árboles húngaros.

Si se encuentra una trayectoria aumentante repite el procedimiento, de otra forma el acoplamiento es óptimo.

Cuando se alcanza el óptimo se presentan los resultados obtenidos.

Esta subrutina utiliza a su vez 6 subrutinas y enseguida veremos brevemente que realiza cada una de ellas :

##### Procedimiento PASO2

El dominio de acoplamiento es asignado a la cubierta de nodos.

W contiene a los nodos expuestos del conjunto S.

##### Procedimiento PASO3

¿ Existen nodos expuestos en S ?

##### Procedimiento PASO4

¿ Existe un nodo l expuesto que se alcance desde el nodo k expuesto, con  $k \in S$  ?

##### Procedimiento PASO5

Actualiza la cubierta de nodos, el conjunto de nodos expuestos S (k es considerado revisado), porque desde él no se encontró una trayectoria aumentante.

#### Procedimiento PASOS

Genera la trayectoria aumentante.

Mejora la cardinalidad del acoplamiento.

#### Procedimiento RESULTADOS

Presenta la matriz de incidencia de nodos en S contra nodos en T de la gráfica y del acoplamiento óptimo, los nodos que forman parte de una cubierta mínima de nodos y los representa en modo gráfico.

#### Subrutina ACOPLA

Esta subrutina es la parte medular del programa que obtiene la solución óptima del problema del acoplamiento de cardinalidad máxima en el caso general.

Se invoca después del procedimiento REPRESENTA que realiza la representación de la gráfica por medio de listas de adyacencia. Se inicializan los valores para comenzar con el acoplamiento vacío y en ese momento da inicio el procedimiento ACOPLA, que realiza lo siguiente:

Utilizando las listas de adyacencia acopla el mayor número posible de nodos, si el número de nodos expuestos es mayor o igual a dos busca trayectorias aumentantes, en otro caso la solución es óptima.

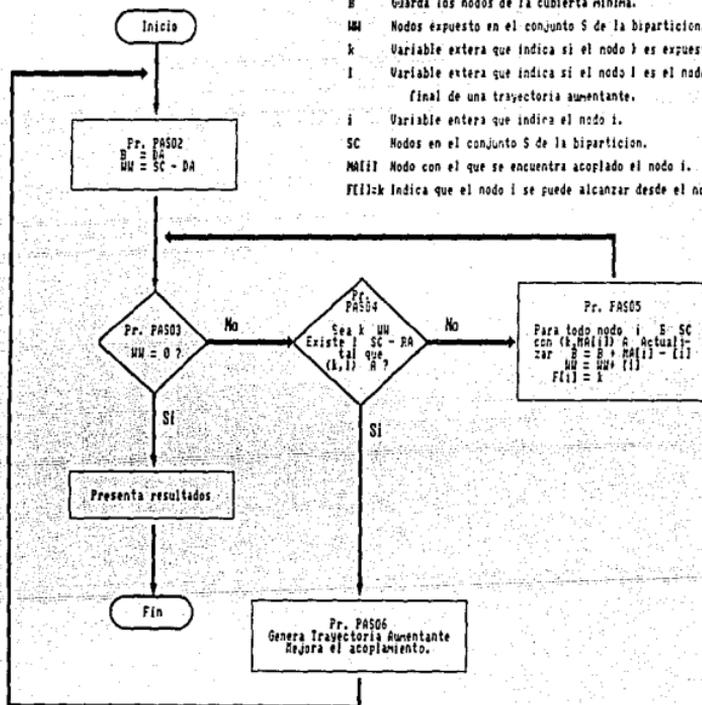
Notemos que al acoplar el mayor número de nodos posibles, la búsqueda de trayectorias aumentantes disminuye considerablemente.

1.- Para cada nodo de la gráfica revisa si es expuesto, y enraiza un árbol alternante en caso afirmativo, expandiéndolo hasta que encuentra una trayectoria aumentante o encuentra que el árbol es húngaro. En el primer caso mejora el acoplamiento y repite el proceso desde 1, en el otro caso considera a ese nodo revisado y busca en otro nodo-raíz, hasta agotar los nodos.

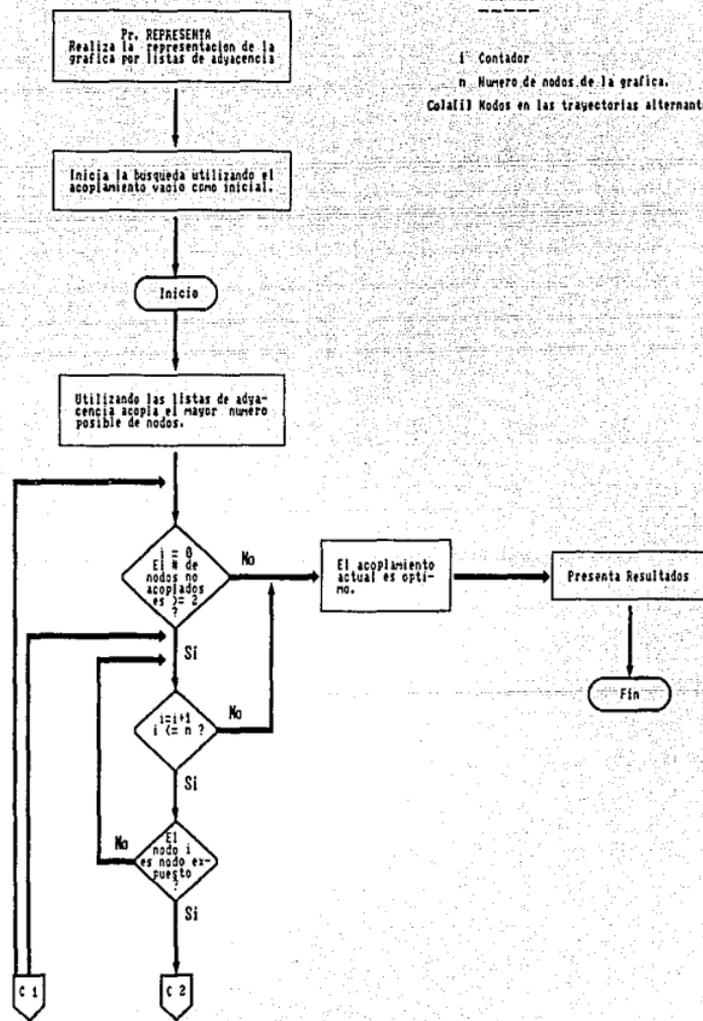
## Procedure Principal

### VARIABLES

- B Guarda los nodos de la cubierta minima.
- MM Nodos expuesto en el conjunto S de la biparticion.
- k Variable externa que indica si el nodo k es expuesto.
- l Variable externa que indica si el nodo l es el nodo final de una trayectoria aumentante.
- i Variable entera que indica el nodo i.
- SC Nodos en el conjunto S de la biparticion.
- MM[i] Nodo con el que se encuentra acoplado el nodo i.
- FF[i]k Indica que el nodo i se puede alcanzar desde el nodo k.



# Procedure ACOPLA



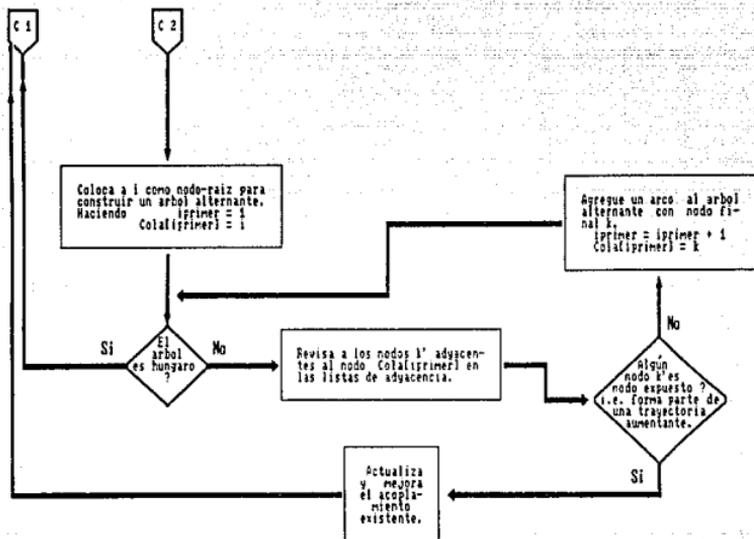
## VARIABLES

i Contador

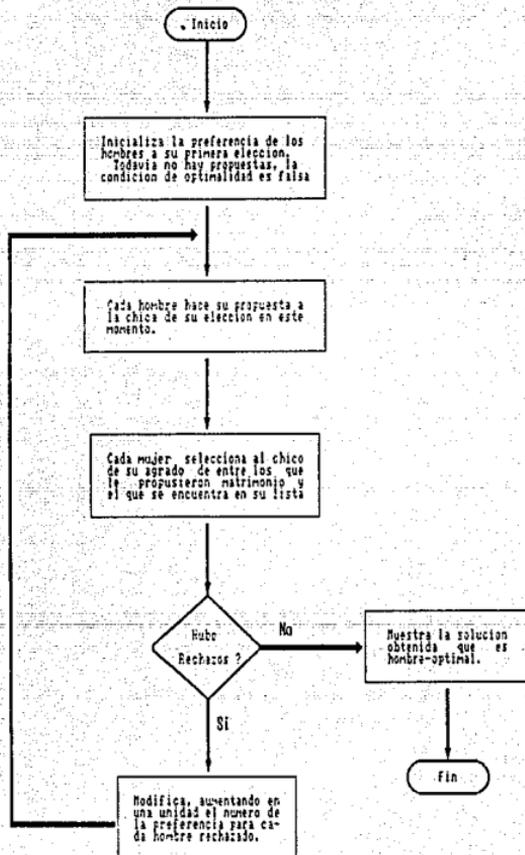
n Numero de nodos de la grafica.

Cola[] Nodos en las trayectorias alternantes.

CONTINUA ...  
**Procedure ACOPLA**



## Procedure ACOPLA\_GALE



En esta sección se presenta la información para el usuario de este paquete; se presentan las opciones que realiza el menú, como capturar la información y la obtención de los resultados:

El nombre del programa ejecutable es ACOPLA.EXE, para su ejecución desde el sistema operativo solamente se teclea ACOPLA. Al entrar al programa se presenta un mensaje de bienvenida, informando el nombre del autor y los problemas que resuelve el sistema. Al llegar al menú principal se presentan 5 opciones:

#### TEORIA DEL ACOPLAMIENTO

- F1 : Problema del acoplamiento de máxima cardinalidad en gráficas bipartitas.
- F2 : Problema del acoplamiento de peso máximo en gráficas bipartitas.
- F3 : Problema del acoplamiento de máxima cardinalidad en gráficas generales.
- F4 : Problema del acoplamiento Gale-Shapley.
- Esc: Salir

Tu opción es :

Selección en el menú principal :

El cursor se posiciona junto a la palabra OPCION, esperando que el usuario oprima la tecla correspondiente a su elección. ( Las teclas [F1] se localizan generalmente a la izquierda del teclado o en la parte superior del mismo ).

Al oprimir la tecla [Esc] se abandona el programa. Al presionar cualquiera de las otras cuatro opciones se llega a menús secundarios (Submenús).

En cualquier submenú se presenta el siguiente mensaje:

S	———>	S <u>o</u> lución a un problema dado por el usuario.
P	———>	P <u>r</u> esentación de un ejemplo guardado en memoria.
Tu opción es : <u>  </u>		

Para elegir una opción se debe presionar la tecla mostrada a la izquierda de cada alternativa.

P Realiza la presentación de un ejemplo del tipo de problema a resolver; se indican mediante distintas pantallas los datos que utiliza.

Así, en el caso del problema del acoplamiento máximo en gráficas bipartitas, se muestra la matriz de adyacencia de la bigráfica a utilizar, la matriz de adyacencia del acoplamiento óptimo y por último se presenta en modo gráfico la representación de la gráfica y el acoplamiento óptimo, también se presenta la solución al problema de la cubierta de nodos que es el problema dual.

S Realiza la solución a un problema dado por el usuario. Mediante una pantalla inicial indica los datos necesarios que debe proporcionar el usuario.

Veamos lo que ocurre en los problemas de bigráficas, la primera pantalla que se muestra es la siguiente:

DAME EL NUMERO DE NODOS DEL CONJUNTO S DE LA BIPARTICION :

DAME EL NUMERO DE NODOS DEL CONJUNTO T DE LA BIPARTICION :

Esperando en cada caso un número entre 1 y 10.

Si los números de los nodos en los conjuntos S y T son ambos mayores o iguales a tres, para una mayor facilidad en la captura de la gráfica una pantalla indicará que se requiere la matriz de incidencia de nodos en S contra nodos en T, para posteriormente presentar una matriz de dimensiones  $|S| \times |T|$  en forma de hoja de cálculo:

CONJUNTO T

		$ S  + 1$	$ S  + 2$	...	$ S  +  T $
C O N J U N T O	1				
	2				
	3				
	.				
S	$ S $				

Dentro de la pantalla . . . Indicará también que para moverse de una celda a otra se deben presionar las flechas ( $\leftarrow$   $\rightarrow$   $\uparrow$   $\downarrow$ ) y la tecla **Esc** cuando la matriz este lista para usarse.

En el otro caso, cuando la cardinalidad de alguno de los subconjuntos de la bipartición es menor a tres, entonces procede a preguntar si el arco  $(i, j)$  pertenece o no a la gráfica.

En el problema del acoplamiento de cardinalidad máxima, si  $|S|$  y  $|T|$  no son ambos mayores o iguales a tres, se despliega un mensaje indicando que comenzará con el acoplamiento vacío, como acoplamiento inicial. En otro caso muestra la siguiente pantalla :

Comenzaras con un acoplamiento inicial (S/N) ?

Si la respuesta es N, entonces utiliza el acoplamiento vacío como inicial.

Si la respuesta es S, entonces presenta la pantalla en forma de hoja de cálculo para capturar ahora la matriz de incidencia de los nodos en S contra los nodos en T del acoplamiento.

		CONJUNTO T			
		$ S  + 1$	$ S  + 2$	...	$ S  +  T $
C O N J U N T O  S	1				
	2				
	3				
	⋮				
	$ S $				

Al oprimir **Esc** una función valida el acoplamiento dado por el usuario, en caso de no ser un acoplamiento válido para la gráfica del problema se muestra el siguiente mensaje de error:

El acoplamiento inicial no es válido  
Vuelve a intentarlo

Oprime <Enter> para continuar

Se brindan al usuario tres oportunidades de proporcionar un acoplamiento inicial válido, si al tercer intento el acoplamiento tampoco es válido se muestra otro mensaje de error:

Ninguno de tus 3 acoplamientos fue válido  
Comenzaremos con el acoplamiento vacío

Oprime <Enter> para continuar

Teniendo un acoplamiento inicial factible ( Por default el acoplamiento vacío) calcula el acoplamiento óptimo y muestra los resultados obtenidos.

En el problema del acoplamiento de peso máximo posterior a la captura de la matriz de incidencia de la bigráfica se muestran las siguientes pantallas para la captura de los pesos de los arcos en la bigráfica :

Cuál es el peso del arco  $(1, j)$  ? \_

Cuál es el peso del arco  $(1, k)$  ? \_

Oprime <Enter> para continuar

El número de pantallas son tantas como la cardinalidad de  $S$  y sólo pregunta por las aristas que estan en la gráfica  $G$ , aceptando sólo valores ENTEROS en los pesos de las aristas. Al leer otro tipo de información (un real o un símbolo) presenta un mensaje de error. Cuando termina de recibir la información, calcula el acoplamiento óptimo, mostrando los resultados obtenidos.

Los resultados que se muestran son los siguientes :

La matriz de incidencia de la gráfica y del acoplamiento óptimo, así como la solución en modo gráfico. Además en el caso del acoplamiento de peso máximo se muestra la matriz de pesos de los arcos, el valor del acoplamiento óptimo , los valores de las variables duales y la solución del problema dual.

Al finalizar el modo gráfica se regresa al usuario al menú principal.

En el caso del problema del acoplamiento de cardinalidad máxima en gráficas generales, cuando la opción seleccionada es P, se presentan los arcos que forman parte de la gráfica guardada en memoria, el acoplamiento óptimo con dos arreglos, uno indicando el número de nodo y el otro el nodo con el que se encuentra acoplado, así como el número de nodos no acoplados.

Si la opción es S se presenta una pantalla indicando que solucionarás un problema del acoplamiento de cardinalidad máxima cualquier gráfica simple, utilizando la representación interna de listas de adyacencia. Enseguida se muestra la siguiente pantalla :

Cuántos nodos tiene tu gráfica ?

Cuántos arcos tiene tu gráfica ?

El número máximo de nodos y arcos permitidos es 40 y 75 respectivamente.

La captura de arcos se presenta de la siguiente forma :

Escribe el nodo inicial del arco i :

Escribe el nodo final del arco i :

presentando diversas pantallas de error, según sea el caso, como las siguientes; Si el dato no es entero :

Error de lectura en el dato  
Espera un momento por favor

Si el nodo inicial o final que proporcionaste es menor que uno o mayor que la cardinalidad del número de nodos en la gráfica :

Número de nodo fuera de rango  
Espera un momento por favor

Si se intenta escribir un bucle

No se permiten escribir arcos de la forma (1,1)

Al terminar de capturar los arcos se calcula el acoplamiento óptimo y se presenta el resultado obtenido.

En el caso del acoplamiento de Gale-Shapley al oprimir P una pantalla indica al usuario que se resolverá un caso del acoplamiento de Gale-Shapley. La segunda pantalla muestra las matrices de preferencia a utilizarse durante la corrida del programa ejemplo. En la solución se muestran las proposiciones y los rechazos, el acoplamiento final estable hombre-óptimo y el número de iteraciones en las que se alcanzó.

Cuando se oprime S se le indica al usuario que necesitará dar el número de elementos en el acoplamiento y las matrices de preferencia, por medio de las siguientes pantallas

Este programa resuelve el acoplamiento de Gale-Shapley que  
proporciona usted, estimado usuario.

Cuál es el número de casos ?

Con esta pantalla se aceptan solamente un número de casos entre 1 y  
9. Enseguida se muestran las siguientes pantallas

Lista de preferencia del hombre  $i$

Escribe la preferencia # 1 :

Escribe la preferencia # 2 :

Escribe la preferencia #  $n$  :

Presiona <Enter> para continuar

Donde para cada  $i = 1, \dots, n$  se muestra una pantalla de captura en  
la cual se pide la lista de preferencias de cada hombre.

Se muestran diferentes mensajes de error, por ejemplo, si la  
preferencia número  $j =$  preferencia número  $k$  de un mismo hombre.

Preferencia no válida

Espera un momento por favor

Al pasar un breve momento el usuario tiene la oportunidad de dar la información correcta. Análogamente para las listas de preferencia de cada mujer, veamos

Lista de preferencia de la mujer 1

Escribe el # de preferencia por el hombre # 1 :

Escribe el # de preferencia por el hombre # 2 :

Escribe el # de preferencia por el hombre # n :

Presiona Enter para continuar

con diferentes mensajes de error que impiden introducir datos no válidos .

Al terminar este proceso de captura, se presentan las propuestas, los rechazos, el acoplamiento hombre-optimal y el número de iteraciones en las que se alcanzó.

## CONCLUSIONES

En este trabajo se desarrollaron algoritmos teóricamente eficientes que resuelven diferentes problemas de acoplamiento.

Se implementó un paquete computacional que resolviera algunos de estos problemas y que reuniera las siguientes características : ser didáctico y eficiente.

En la parte teórica se proporcionan las bases adecuadas para elaborar algoritmos de algunos otros problemas de tipo combinatorio que no fueron tratados en este trabajo, por ejemplo el uso de la teoría del acoplamiento en problemas NP completos, como el de encontrar circuitos Hamiltonianos.

Con respecto al paquete computacional, que resuelve los problemas del acoplamiento máximo, de peso máximo en gráficas bipartitas y de Gale-Shapley se comentó la necesidad de encontrar una estructura de datos adecuada para nuestros requerimientos logrando un programa eficiente; además gracias a un sistema de ventanas y a la presentación en modo gráfico de la solución a algunos problemas, el programa es amigable, es decir personas que tengan poco conocimiento de computación o de teoría de gráficas podrán utilizarlo sin dificultad.

## BIBLIOGRAFIA

- [1] AHO A., HOPCROFT J. AND ULLMAN J.  
1975 *The Design and Analysis of Computer Algorithms*,  
Addison-Wesley, Reading, Massachusetts.
- [2] BALINSKI M. L.  
1969 Labelling to obtain a maximum matching, *Combinatorial  
Mathematics and its Applications* 4, pages 585-602.
- [3] BANACHOWSKI LECH; KRECZMAR A  
1991 *Analysis of Algorithms and Data Structures*,  
Addison-Wesley, Wokingam, England.
- [4] BAZARAA HOKHTAR  
1989 *Programación Lineal y Flujo en Redes*, Limusa, México.
- [5] BERGE CLAUDE  
1985 *Graphs and Hypergraphs*, North Holland, Amsterdam.
- [6] CARRE BERNARD  
1979 *Graphs and Networks*, Clarendon Press, Oxford.
- [7] CHRISTOFIDES NICOS  
1975 *Graph Theory An Algorithmic Approach*, Academic Press,  
London.
- [8] CHVATAL VASEK  
1983 *Linear Programming*, W. H. Freeman, New York.
- [9] EDMONDS J.  
1965 Paths, trees and flowers, *Canadian Journal of  
Mathematics* 17, pages. 449-467.

- EDMONDS J. AND JOHNSON L.
- [10] 1973 Matching, Euler Tours and the Chinese Postman, *Math. prog.* 5, pags. 88-124.
- [11] Matching: A well solved class of integer linear programs, *Combinatorial Structures and their applications*, pags. 89-92.
- [12] EVEN S. AND KARIV O.  
1975 An  $O(n^{5/2})$  algorithm for maximum matching in general graphs, *16th Annual Symposium on Foundations of Computer Science*, pags. 100-112.
- [13] FORD L.R. AND FULKERSON D.R.  
1962 *Flows in networks*. Princeton University Press, Princeton, N. J.
- [14] GIBBONS ALAN  
1985 *Algorithmic Graph Theory*, The MIT Press, Cambridge, Massachusetts.
- [15] GUSFIELD D. AND IRVING R.D.  
1985 *The Stable Marriage Problem: Structure and Algorithms*, The MIT Press, Cambridge, Massachusetts.
- [16] KWAN MEI-KO  
1962 Graphics programming using odd or even points, *Chinese Mathematics*.
- [17] LAWLER E. L.  
1976 *Combinatorial Optimization: Networks and Matroids*, Holt, Rinehart and Wiston, New York.
- [18] LOVASZ LAZLO AND PLUMMER H. D.  
1986 *Matching Theory*, North Holland, Amsterdam.

- [19] MICALI S. AND VIZIRANI V.  
1980 An  $O(V^{1/2}E)$  algorithm for finding maximum matching in general graphs, *21st Annual Symposium on Foundations of Computers Science*, pags. 17-27, Syracuse, N. J.
- [20] MINIEKA EDWARD.  
1978 *Optimization Algorithms for Networks and Graphs*, Marcel Dekker Inc. New York.
- [21] NEMHAUSER G. AND WASLEY L.  
1988 *Integer and Combinatorial Optimization*, John Wiley, N. Y.
- [22] PAPANIMITRIOU C. AND STEIGLIZ K.  
1982 *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall, New Jersey.