UNIVERSIDAD DE SONORA

DEPARTAMENTO DE MATEMÁTICAS

"EL PROBLEMA DE ACOPLAMIENTO EN GRÁFICAS BIPARTITAS"

TESIS

Que para obtener el Título de

LICENCIADO EN MATEMÁTICAS

Presenta

Albertina Morales Osuna

Hermosillo, Sonora

Mayo 1997

I N D I C E

INTRODUCCION

La palabra acoplamiento significa literalmente formar parejas. Si se tiene un conjunto finito de elementos con los que se pueden formar parejas, se representa la situación con una gráfica que tenga un vértice por cada elemento, donde una arista entre dos vértices significa que los elementos representados por tales vértices pueden forman pareja, pero además las parejas no deben tener un vértice en común.

En los problemas de acoplamiento en gráficas, la idea más general consiste en encontrar subconjuntos de aristas que cumplen ciertos criterios de optimización, donde los parámetros involucrados pueden ser el número de aristas en la solución, el número de vértices que son extremos del total de aristas o el peso asociado a las aristas. En éste trabajo sólo se estudiarán dos tipos de problemas de acoplamiento los cuales son: Acoplamiento de Cardinalidad Máxima y Acoplamiento con Peso (problema de asignación), ambos en gráficas bipartitas.

Los problemas de acoplamiento se estudiaron por mucho tiempo tanto en Análisis Combinatorio como en Investigación de Operaciones, llamando la atención de muchos investigadores interesados por sus múltiples aplicaciones tales como: sistema de representantes distintos, problema de cubrimiento, descomposición en cadenas, cuadrados latinos, problema del matrimonio, calendarización de vuelos, enrole de locomotoras, etc. y sus diversas variantes. Sin embargo, en 1957 Claude Berge fué el primero en plantear estos problemas usando el concepto de trayectoria aumentante. El demostró que un acoplamiento es máximo si y sólo sí no hay trayectoria aumentante, éste es un resultado clásico de Optimización Combinatoria el cual es la base de los algoritmos más eficientes para resolver los problemas de acoplamiento.

El método Húngaro para la solución del problema de acoplamiento con peso (problema de asignación) está fundamentado en el trabajo de dos Matemáticos Húngaros, D. Köning y J. Egerváry, y particularmente en el teorema dado por Köning, este teorema fué establecido y demostrado en términos de teoría de gráficas. Una prueba de éste teorema usando teoría de gráficas fué dada por Kuhn en 1955 quien desarrolló un algoritmo para obtener n ceros independientes (en distinto renglón y distinta columna) de una matriz obteniendo un acoplamiento de cardinalidad máxima en una gráfica bipartita asociada. Este algoritmo está basado en el trabajo de los matemáticos Húngaros Köning y Egerváry. En honor a ellos le llamó a éste Método Húngaro o Algoritmo Húngaro. Este trabajo condujo al siguiente año, al método primal-dual para obtener el acoplamiento de peso máximo en gráficas generales.

Este trabajo tiene por objeto el análisis y desarrollo de los fundamentos teóricos de los principales métodos de solución de problemas de acoplamientos. Así como su implementación para la solución de problemas de aplicación.

Para lograr los objetivos anteriores se proporciona una presentación lo más sencilla posible, cuidando principalmente un enfoque algorítmico que permita una fácil implementación en computadora de los algoritmos más eficientes. Por lo cual los conceptos y resultados teóricos presentados en este trabajo son sólo los necesarios, sin embargo, se incluye un apéndice y una amplia bibliográfia.

Debido aque al modelar problemas prácticos por lo regular resultan de tamaño considerables, para su solución es imprescindible la implementación de algoritmos eficientes. Los algoritmos fueron implementados en Turbo pascal utilizando estructura de datos para la representación y manipulación de una gráfica.

El contenido de este trabajo se desarrolló de la siguiente manera:

El capítulo I consta de una visión intuitiva de algunos tipos de problemas de acoplamiento, su aplicación a distintos problemas reales, formulación del problema y análisis comparativo de los principales métodos de solución.

En el capítulo II se dan los conceptos básicos de teoría de gráficas y acoplamiento, para enunciar y demostrar los teoremas fundamentales de Berge, Hall y Köning, tales teoremas proporcionan las bases de los algoritmos que se utilizan en este trabajo para resolver el problema de acoplamiento. También se da la implementación del algoritmo y se concluye este capítulo resolviendo un ejemplo.

En el capítulo III se desarrolla el método Húngaro, así como la justificación de sus pasos y su convergencia. Debido a que el método Húngaro es tipo primal-dual se requerirá de resultados de dualidad lineal. Además se da la transformación de un problema de minimización a uno de maximización y también se da la implementación computacional de éste método y se resuelve un ejemplo.

Finalmente se dan las concluciones, bibliografía y apéndice que contiene los teoremas básicos de dualidad lineal y la solución de los problemas planteados en el capítulo I.

CAPITULO I

En este capítulo se da una visión intuitiva de algunos tipos de problemas de acoplamiento, haciendo énfasis especial en sus múltiples aplicaciones en distintos problemas reales por ejemplo: Sistema de distintos representantes, Descomposición en cadenas, Calendarización, etc. Así como la formulación de los problemas de acoplamiento en términos de programación lineal y análisis comparativo de los principales métodos de solución.

1.- DESCRIPCION DEL PROBLEMA

La palabra acoplamiento significa literalmente formar parejas. Si se tiene un conjunto finito de elementos con los que se pueden ordenar parejas, se representa la situación con una gráfica que tenga un vértice por cada elemento, donde una arista entre dos vértices significa que los elementos representados por tales vértices forman pareja, pero además las parejas no deben tener un vértice en común.

En los problemas de acoplamiento en gráficas, la idea más general consiste en encontrar subconjuntos de aristas que cumplen ciertos criterios de optimización, donde los parámetros involucrados pueden ser el número de aristas en la solución, el número de vértices que son extremos del total de aristas o el peso asociado a los aristas.

Para enunciar de modo formal lo que es un acoplamiento, primero se definirá lo que es una gráfica.

Definición: Una gráfica $G(V,A)$ consta de:

i) un conjunto finito no vacío $V=V(G)$ cuyos elementos se llaman vértices (nodos ó puntos) de G.

ii) $A=A(G)$ es el conjunto finito de pares no ordenados de elementos de $V(G)$ llamadas aristas (arcos, líneas o lados).

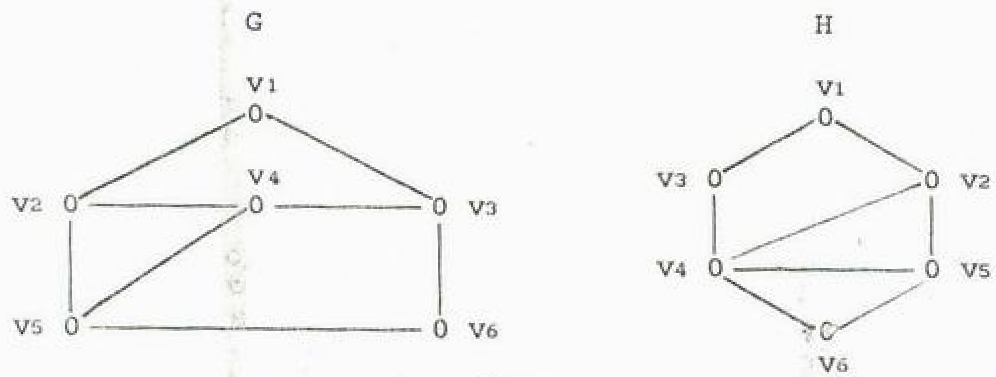


figura 1.1

Por ejemplo para las gráficas G y H los conjuntos de vértices son iguales, es decir, $V(G) = V(H) = \{v_1,v_2,v_3,v_4,v_5,v_6\}$, y los conjuntos de aristas son: $A(G) = \{(v_1,v_2),(v_5,v_4),(v_6,v_3),\ldots\}$, $A(H) = \{(v_4,v_2), v_4,v_5),(v_2,v_5),\ldots\}$.

Definición: Dada una gráfica G=(V,A), un acoplamiento es un subconjunto M de aristas ($M \subseteq A$) en donde no hay dos de ellas que tengan un vértice común.

En la figura 1.2 las aristas con una línea gruesa representan un acoplamiento, en G el acoplamiento no es el de cardinalidad máxima, en H si lo es.

figura 1.2

Definición: Una gráfica G(V,A), es bipartita si existen $V_1, V_2 \subseteq V(G)$ tal que $V_1 \cap V_2 = \emptyset$, $V(G) = V_1 \cup V_2$ y toda arista de A(G) tiene un extremo en V_1 y otro en V_2. Esta gráfica bipartita se denota $G(V_1, V_2, A)$.

En la figura 1.3 se muestran gráficas bipartitas donde para la primera figura $V_1=\{x_i/i=1,2,3\}$ y $V_2=\{y_i/ \ i=1,2,3\}$, para la segunda $V_1=\{x_i/ \ i=1,2\}$ y $V_2=\{y_i/ \ i=1,2\}$.

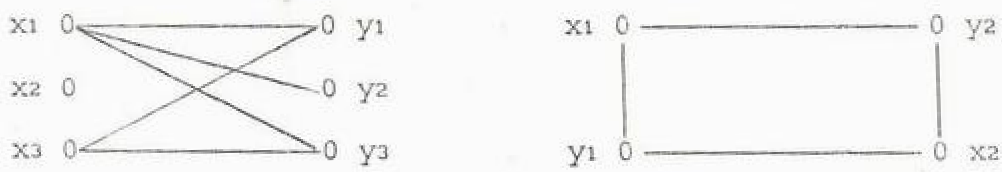


Figura 1.3

Las gráficas de la figura 1.1 no son bipartitas.

Existen varios problemas de acoplamiento pero, en este trabajo sólo nos interesan dos, los cuales son: **Acoplamiento de cardinalidad máxima y Acoplamiento con peso.**

El problema de acoplamiento fué planteado primero por Claude Berge en 1957. Desde entonces llamó la atención de muchos investigadores interesados por sus múltiples aplicaciones y sus diversas variaciones.

A continuación se describen ambos problemas de acoplamiento en gráficas bipartitas.

2.- ACOPLAMIENTO DE CARDINALIDAD MAXIMA

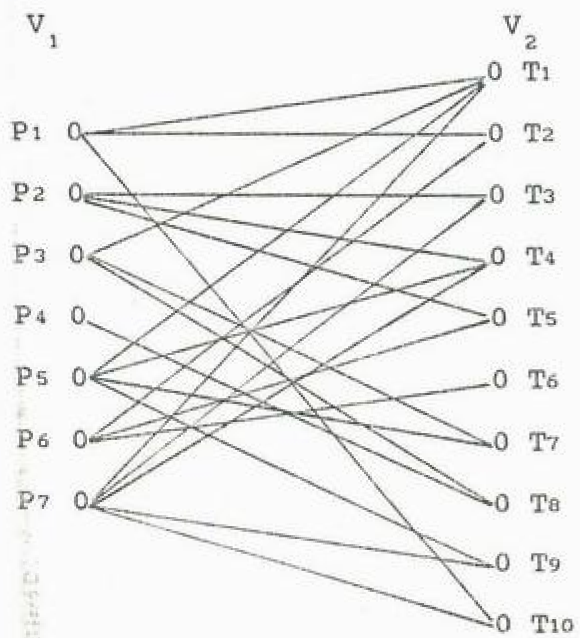
Dada una gráfica G con un número finito de vértices y aristas, se desea determinar el acoplamiento de mayor cardinalidad (número de aristas) posible. A este problema también se le conoce como acoplamiento sin peso o acoplamiento máximo.

Ejemplo : El Departamento de Idiomas de la Universidad de Sonora tiene 7 profesores para cubrir 10 trabajos de traducción. En la siguiente tabla se puede determinar si el profesor tiene o no habilidades para realizar un trabajo de traducción.

	Trabajos									
Prof	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}
P_1	si	si	no	no	no	no	no	no	no	si
P_2	no	no	si	si	si	no	no	no	no	no
P_3	si	no	no	no	no	no	si	si	no	no
P_4	no	no	no	no	no	no	no	si	no	no
P_5	si	no	no	si	no	no	si	no	si	no
P_6	no	si	no	no	si	si	no	no	no	no
P_7	si	no	si	si	no	no	no	no	si	si

El problema es determinar el máximo número de trabajos que puede realizar el Departamento de Idiomas, de tal forma que a cada profesor no debe asignarse más de un trabajo.

La representación gráfica del problema anterior es la siguiente:



Donde los conjuntos V_1 y V_2 son los profesores y trabajos respectivamente, la arista (P_i, T_j) indica que el profesor P_i si puede realizar el trabajo T_j. El problema anterior es equivalente a determinar el acoplamiento de cardinalidad máxima de esta gráfica bipartita.

Un acoplamiento de cardinalidad máxima es M = { (P_1,T_1), $(P_2,T_3,)$, (P_3,T_7), (P_4,T_8), (P_5,T_4), (P_6,T_2), (P_7,T_9) } por lo que el número máximo de trabajos que puede realizar el Departamento de Idiomas es 7.

Si al profesor P_3 se acopla con la tarea T_8, ¿ Se tendrá un acoplamiento máximo? ¿Se acoplarán a todos los profesores?.

Existen diversos problemas de Optimización Combinatoria que pueden ser planteados de manera equivalente al problema de acoplamiento de cardinalidad máxima. Un par de ejemplos se citan enseguida.

2.1.- Sistema de representantes distintos

Definición: Dado un conjunto finito R, y $\Gamma=\{R_1,R_2,..,R_K\}$ una colección de subconjuntos de R, entonces un sistema de representantes distintos de Γ es un conjunto de k elementos tales que $r_i \neq r_j$ si $i \neq j$ y $r_i \in R_i$, para i=1,2,...,k.

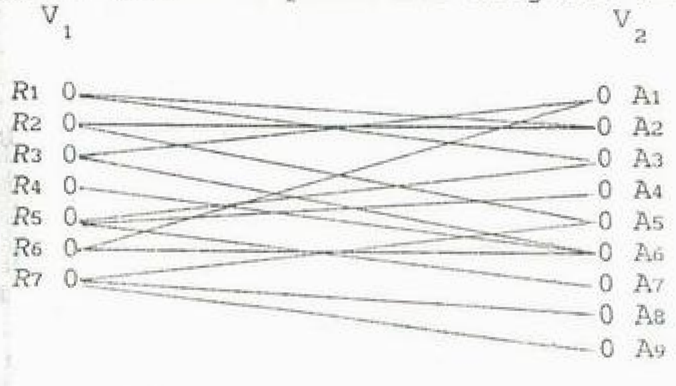
En el siguiente ejemplo se requiere encontrar un sistema de representantes distintos:

Ejemplo : Un despacho de nueve asesores industriales atiende a 7 empresas distintas. Considerando el conjunto de asesores $\{A_j \ / \ j = 1,2..,9\}$ la forma en que están asignados a las empresas es como sigue:

Empresa # 1	(A_2,A_3)
Empresa # 2	(A_2,A_5)
Empresa # 3	(A_1,A_6)
Empresa # 4	(A_6)
Empresa # 5	(A_3,A_4,A_7)
Empresa # 6	(A_1,A_6)
Empresa # 7	(A_5,A_8,A_9)

Para la reunión anual de evaluación del despacho, se desea formar un grupo de 7 asesores, representando cada uno de ellos a una empresa diferente ¿Cómo se integra el grupo deseado?.

En este problema $R=\{A_j \ / \ j=1,2,..,9\}$ es el conjunto de asesores, $R_i=\{A_j \ / \ A_j$ puede asesorar a la empresa $i\}$ y se modela en una gráfica bipartita escogiendo $V_1=\{R_i \ / \ i=1,2,..,7\}$, $V_2=R$ donde la arista (R_i,A_j) significa que el asesor A_j puede ser asignado a la empresa R_i.



A partir de esta gráfica, el problema se puede resolver encontrado el acoplamiento de cardinalidad máxima, este acoplamiento es fácil de encontrarlo en la gráfica anterior. Si el acoplamiento máximo contiene menos de 7 aristas, 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.

En el capítulo 2 se darán las condiciones necesarias y suficientes para la existencia de un sistema de representantes distintos de R_i.

2.2.- Descomposición en cadenas

En algunos problemas que se manejan con gráficas, los elementos representados por los vértices pueden ser ordenados de modo estricto ($a<b$) de manera que se refleje un orden de precedencia entre ellos. Por supuesto, existe la posibilidad de que dos elementos en particular no puedan compararse.

La formalización de la idea anterior es como sigue:

Definición: Para un conjunto finito N, un orden parcial estricto es una relación definida en el producto cartesiano NxN que es transitiva y antisimétrica.

Por ejemplo, sea S un conjunto finito y P(S) el conjunto potencia. La relación de inclusión propia ($\subset$) establece un orden parcial estricto en P(S).

Definición: Dos elementos i.j de N se llaman comparables si ocurre $i<j$ ó $j<i$, en caso contrario se llaman no-comparables.

Definición: Una cadena en N es un subconjunto de elementos de N tal que toda pareja es comparable.

Por la transitividad la cadena puede escribirse en forma de orden ascendente como: $n_1 < n_2 < \ldots < n_k$. Se consideran cadenas a los elementos aislados.

Definición: La longitud de una cadena es igual al número de elementos que contiene.

Definición: Una descomposición en cadenas de N es una colección de cadenas disjuntas cuya unión forma N.

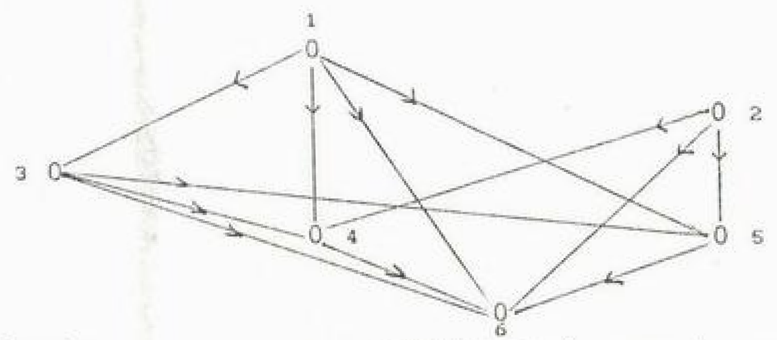
Uno de los problemas de Optimización Combinatoria es encontrar la descomposición que tenga el menor número de cadenas, a este problema se le llama descomposición en cadenas.

Ejemplo : Un periódico planea cubrir las conferencias de la Reunión Internacional de Legisladores a celebrar en cierta fecha. Para tal día se ofrecen 6 conferencias con tiempo de inicio y duración como se muestra a continuación:

Conferencia	1	2	3	4	5	6
Inicio	8:00	8:30	9:00	10:00	10:00	11:00
Duración (horas)	1	1	1	1	1	1

¿Cuál es el mínimo número de reporteros que pueden cubrir el total de las conferencias?

Observe que las conferencias se pueden ordenar de manera que a<b significa que la conferencia 'a' termina antes de que la conferencia 'b' comience. La siguiente gráfica modela el orden parcial estricto que existe entre las conferencias:

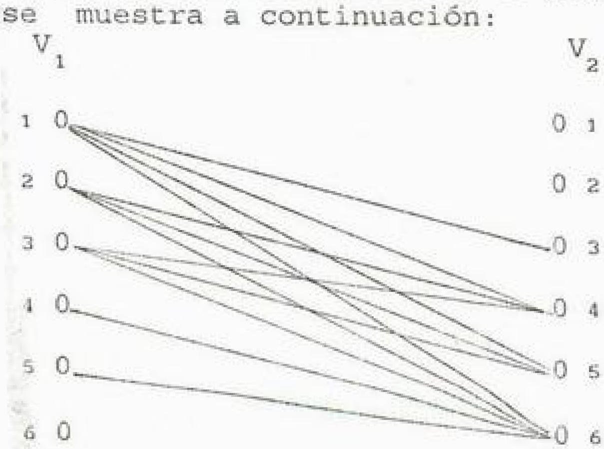


Cada vértice representa la conferencia con la que está numerado, y si se puede alcanzar el vértice 'b' desde el vértice 'a' considerando la dirección de las aristas, se tiene la condición a<b.

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

El problema se puede reducir a un caso de acoplamiento de cardinalidad máxima como sigue: Sea N={1,2,3,4,5,6} el conjunto de las conferencias, y G una gráfica bipartita donde $N=V_1=V_2$, y la arista (i,j) representa la condición (conferencia i) < (conferencia j).

La gráfica G se muestra a continuación:



Encontrar un acoplamiento máximo en la gráfica anterior, equivale a encontrar un subconjunto de pares de elementos de N que son comparables y 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, se unen los pares y se forman cadenas disjuntas.

El acoplamiento máximo de la gráfica anterior es: (1,3), (2,4), (3,5), (4,6) con estas aristas se pueden construir dos cadenas $1 < 3 < 5$ y $2 < 4 < 6$, las cuales contiene a todos los elementos de N, el total de reporteros que se necesitan son 2. En caso de que existan elementos que no sean cubiertos por las aristas del acoplamiento máximo (ni en V_1 y ni en V_2) se considerarán como cadenas de longitud uno.

Otro ejemplo de descomposición en cadenas es: Sea $N=\{n_1,n_2,..,n_k\}$ un conjunto de tazones de tamaños variados, donde la relación '$n_i<n_j$' significa que se puede acomodar el tazón n_i dentro del n_j. Esta relación resulta ser un orden parcial estricto en N, y una cadena es un subconjunto de tazones que pueden 'anidarse'. El problema es como acomodar los tazones unos dentro de otros tal que forme el menor número posible de 'nidos'.

En el segundo capítulo se justifica la reducción del problema de descomposición en cadenas a uno de acoplamiento máximo.

2.3.- Rectángulos latinos

Definición: Un rectángulo latino de hxn basado en los enteros 1,2,3...,n, es un arreglo de h renglones y n columnas formadas con los enteros 1,2,3,..,n, de tal manera que los enteros en cada renglón y cada columna son diferentes.

Algunos ejemplos son los siguientes:

Rectángulo latino de 3x5

1	2	3	4	5
3	4	5	1	2
2	3	4	5	1

Rectángulo latino de 4x3

4	3	1
1	2	3
3	4	2
2	1	4

Definición: Un cuadrado latino nxn es un arreglo de n renglones y n columnas de los enteros 1,2,3,..,n, de tal manera que ningún entero aparece más de una vez en cada renglón y en cada columna.

Los siguientes son algunos ejemplos:

Cuadrado latino de 3x3

1	2	3
2	3	1
3	1	2

Cuadrado latino de 5x5

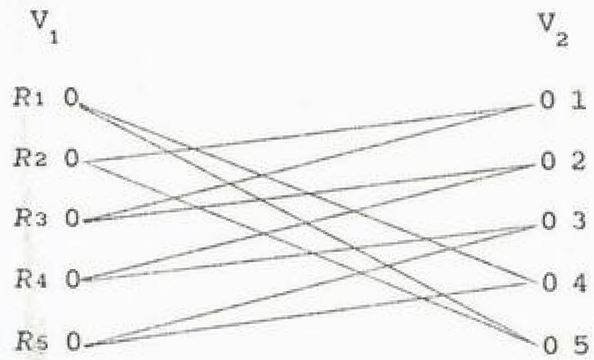
1	2	3	4	5
2	3	4	5	1
5	1	2	3	4
4	5	1	2	3
3	4	5	1	2

A continuación se muestra como extender un rectángulo latino a un cuadrado latino obteniendo un sistema de representantes distintos.

Dado un rectángulo latino de hxn (h < n) formado con los enteros 1,2,3,..,n , se escoge R = { 1,2,3,..,n } y una colección de subconjuntos de R, $\Gamma=\{R-C_i$ / C_i es el conjunto de elementos en cada columna i}. Por lo tanto $R_i=(R-C_i)$ es el conjunto de elementos de R que no estan en la columna i. Añadiendo como renglón el sistema de representantes distintos de Γ se obtiene un nuevo rectángulo (o cuadrado) latino. Para convertir un rectángulo de hxn (n > h) a un cuadrado latino, lo que se hace es repetir (n-h)-1 veces el sistema de representantes distintos.

Por ejemplo para el rectángulo latino de 3x5 dado anteriormente $R=\{1,2,3,4,5\}$ y la colección de subconjuntos Γ es: $R_1=\{4,5\}$, $R_2=\{1,5\}$, $R_3=\{1,2\}$, $R_4=\{2,3\}$ y $R_5=\{3,4\}$.

Para representarlo en una gráfica se escoge $V_1=\{R_i/\ i=1,2,..,5\}$ y $V_2= R$ quedando como sigue



donde la arista (R_i,j) significa que j no está en la columna C_i.

El sistema de representantes distintos de Γ es: $r_1=5$, $r_2=1$, $r_3=2$, $r_4=3$, $r_5=4$. Por lo tanto agregando el renglón (5,2,1,3,4) se obtiene el nuevo rectángulo latino de 4x5

Rectángulo latino de 4x5

1	2	3	4	5
3	4	5	1	2
2	3	4	5	1
5	1	2	3	4

Observe que a éste rectángulo le hace falta añadir un renglón para convertirlo a un cuadrado latino, pero esto es fácil de completar sólo hay que ver el número que hace falta.

Así, el rectángulo de 3x5 dado anteriormente se transforma en el siguiente cuadrado latino de 5x5:

Cuadrado latino de 5x5

1	2	3	4	5
3	4	5	1	2
2	3	4	5	1
5	1	2	3	4
4	5	1	2	3

Un procedimiento similar se realiza para aumentar columnas al extender rectángulos latinos de nxh cuando n >h basados en los enteros 1,2,..,n.

3.- ACOPLAMIENTO CON PESO

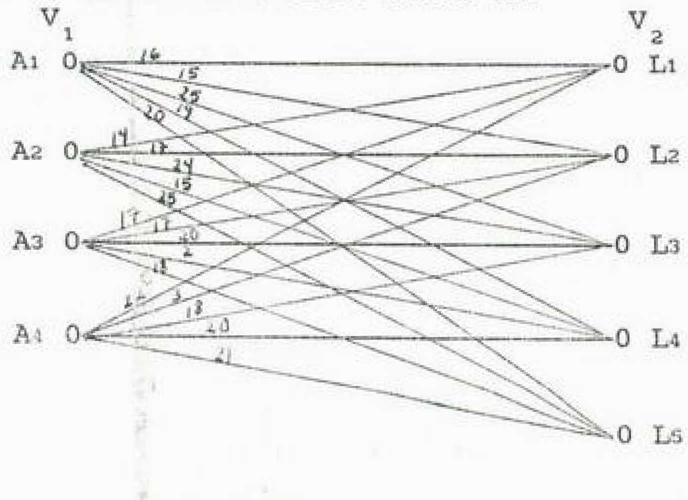
En este problema se tiene una gráfica G donde cada arista tiene asociada un peso que puede ser positivo, negativo o cero. Lo que se quiere es hallar un acoplamiento en donde la suma de los pesos de sus aristas sea máxima. Este problema es una generalización del problema de acoplamiento de cardinalidad máxima.

Al problema de acoplamiento con peso se le conoce como un problema de asignación pues se aplica de modo natural a la búsqueda de la asignación óptima de aspirantes (elementos de V_1) a tareas (elementos de V_2) cuando se conoce la eficiencia del aspirante 'i' en la tarea 'j' que se denota por el peso C_{ij} .

Ejemplo : Un corredor de bienes raíces planea la venta de cuatro lotes de terreno y ha recibido ofertas individuales de cuatro clientes. Debido a la cantidad de capital que se requiere, estas ofertas se han hecho en el entendimiento de que ninguno de los cuatro clientes comprará más de un lote. El corredor de bienes raíces quiere maximizar su ingreso total a partir de las siguientes ofertas (dadas en miles de pesos):

	lote				
comprador	L_1	L_2	L_3	L_4	L_5
A_1	16	15	25	19	20
A_2	14	17	24	15	25
A_3	17	17	20	2	18
A_4	22	3	18	20	21

La gráfica asociada a esta tabla es:



Donde los conjuntos V_1 y V_2 representan a los lotes y compradores respectivamente, además la arista (A_i, L_j) indica que el comprador A_i se interesa por el lote L_j y c_{ij} es lo que ofrece el comprador A_i por el lote L_j.

¿Cómo deben elegirse las ofertas de tal manera que se maximize el ingreso total?.

Para que se maximize el ingreso total el acoplamiento de peso máximo debe ser:

$A_1 \xrightarrow{25} L_3$, $A_2 \xrightarrow{25} L_5$, $A_3 \xrightarrow{17} L_2$ y $A_4 \xrightarrow{22} L_1$, cuyo valor es 89.

Puede observarse que un acoplamiento (asignación) en la matriz de pesos $[C_{ij}]_{mxn}$ (considerando $m < n$) es escoger un elemento de cada renglón en distinta columna. Como se ilustra en la siguiente matriz de peso con (*):

	lote				
comprador	L_1	L_2	L_3	L_4	L_5
C_1	16	15	25 *	19	20
C_2	14	17	24	15	25 *
C_3	17	17 *	20	2	18
C_4	22 *	3	18	20	21

Existen problemas que requieren ser transformados para poder ser modelados como problemas de acoplamiento con peso, los ejemplos que se presentan a continuación son de éste estilo.

3.1.- Calendarización de vuelos

Supóngase que Aero-México tiene el siguiente horario de vuelos diarios México - Río de Janeiro:

salida México	vuelo	llegada Río de Janeiro
6:00	A	12:00
7:30	B	13:30
11:30	C	17:30
19:30	D	1:00
0:30	E	6:30

llegada México	vuelo	salida Río de Janeiro
11:30	1	5:30
15:00	2	9:00
21:00	3	15:00
0:30	4	18:30
6:00	5	00:00

El problema que tiene Aero-México, es la calendarización de la tripulación en estos vuelos. Resulta que una tripulación (piloto, copiloto, aeromosa,etc.) que sale de México un lunes a las 7:30, llega a Río de Janeiro ese mismo lunes a las 13:30, sale el martes a las 9:00 de Río de Janeiro y llega a México a las 15:00. El tiempo transcurrido desde las 13:30 del lunes hasta el martes a las 9:00, es un tiempo muerto. En este caso se trata de reducir los tiempos muertos de las tripulaciones en estos vuelos, sujeto a la siguiente condición, cada tripulación debe descansar al menos 8 horas pero no más de 24 horas, si esta condición no se cumple el tiempo muerto es un valor muy grande (M).

El problema se puede enunciar de la siguiente manera: Dónde deben vivir las tripulaciones y qué tripulaciones deben asignarse a los vuelos, tal que los tiempos muertos totales se minimizen y al mismo tiempo se respeten las condiciones de descanso de las tripulaciones.

Supóngase una tripulación que vive en la ciudad de México, que trabaja en el vuelo C y se regresa en el vuelo 2 de Río de Janeiro. De acuerdo con los tiempos de vuelo, esa tripulación llega a las 17:30 y sale a las 9:00 de la mañana rumbo a México, transcurriendo 15 horas 30 minutos de tiempo muerto. En cambio, una tripulación que vive en Río de Janeiro y sale en el vuelo 4 hacia México, y regresa en el vuelo A a Río, tiene un tiempo muerto de 5 horas y 30 minutos, pero éste tiempo no cumple con la condición anterior así que el tiempo muerto es M.

Como se puede observar una asignación será un vuelo redondo.

Así se pueden construir dos matrices de tiempos muertos, a saber:

MATRIZ I

	1	2	3	4	5
A	17.5	21	M	M	12
B	16	19.5	M	M	10.5
C	12	15.5	21.5	M	M
D	M	8	14	17.5	23
E	23	M	8.5	12	17.5

Tiempos muertos para tripulaciones con sede en México (horas)

MATRIZ II

	1	2	3	4	5
A	18.5	15	9	M	M
B	20	16.5	10.5	M	M
C	M	20.5	14.5	11	M
D	M	M	22	18.5	13
E	13	9.5	M	M	18.5

Tiempos muertos para tripulaciones con sede Río de Janeiro (horas)

La matriz I se interpreta como sigue: Los vuelos de salida (de México) van a ser los renglones y las columnas serán los vuelos de regreso (desde Río de Janeiro). De la misma manera en la matriz II las columnas representan los vuelos de salida (de Río de Janeiro) y los renglones los vuelos de regreso (desde México).

Dadas estas dos matrices, se construye una nueva, donde las entradas t_{ij} serán:

$$t_{ij} = \min\ (t^{I}_{ij}\ ,\ t^{II}_{ij}\ )$$

donde:

t_{ij} = Es el tiempo muerto de la tripulación que sale en el vuelo i y se regresa en el vuelo j.

t^{I}_{ij} = Es el tiempo muerto en la posición (i,j) de la matriz I

t^{II}_{ij} = Es el tiempo muerto en la posición (i,j) de la matriz II

La nueva matriz es:

MATRIZ III

	1	2	3	4	5
A	17.5	15	9	M	12
B	16	16.5	10.5	M	10.5
C	12	15.5	14.5	11	M
D	M	8	14	17.5	13
E	13	9.5	8.5	12	18.5

Observe que en esta matriz ya se pierde información sobre que tripulación sale (de México o Río de Janeiro) y que tripulación se regresa (a México o Río de Janeiro), por ejemplo si en la matriz III se escoge $T_{ij}=14$ (i=D,j=3) no se sabe de que matriz proviene, entonces este número se busca en la entrada (D,3) de las matrices I y II, encontrándose este en la matriz I. Por lo tanto se conservarán las matrices I y II para el resultado posterior.

De aquí la matriz III representa los pesos t_{ij} (en este caso tiempos muertos) y las restricciones son cada tripulación podrá salir únicamente en un vuelo y podrá regresarse en un sólo vuelo, por lo tanto es un problema de asignación.

La gráfica bipartita asociada $G(V_1,V_2,A)$ se forma con $V_1=\{A,B,C,D,E\}$, $V_2=\{1,2,3,4,5\}$ y la arista (i,j) con peso t_{ij}.

3.2.- El enrole de locomotoras para un itinerario de trenes

El enrole de locomotoras consiste en la planificación de la asignación de locomotoras a los trenes fijados por un itinerario (con llegadas y salidas pre-establecidas) de manera que la circulación de estos quede asegurada.

La calidad del enrole se mide por la suma de los tiempos muertos de las locomotoras, el cual se desea minimizar.

Para el planteamiento formal de este problema se definirán las siguientes variables:

S_{ro} = hora de salida del tren r de la estación o.
L_{pd} = hora de llegada del tren p a la estación d.
M_d, M_o = son los tiempos de estancia de las locomotoras en las estaciones d y o respectivamente.
t_{od} = tiempo de recorrido (locomotora sola) entre las estaciones o y d.

Observe que la locomotora que deja el tren p en la estación d para que pueda enrolarse al tren r que sale de la estación o es necesario:

$$(S_{ro} - L_{pd}) \geq M_o + M_d + t_{od}.$$

En caso contrario, si la locomotora no alcanza a enrolarse con el tren r en el mismo día es necesario sumar el menor múltiplo de 24 horas al lado izquierdo para que lo pueda alcanzar, es decir:

$$(S_{ro} - L_{pd}) + 24x \geq M_o + M_d + t_{od}.$$

Por lo tanto el tiempo muerto se obtiene por:

$$(S_{ro} - L_{pd}) + 24x$$

donde x es el menor entero que hace cumplir la desigualdad

$$(S_{ro} - L_{pd}) + 24x \geq M_o + M_d + t_{od}$$

Para evitar que hayan recorridos de locomotoras solas se introduce un coeficiente de peso PC ≥ 1 (o una penalización) que al multiplicarse por el tiempo muerto refleje la inconveniencia de recorrido vacío de la locomotora.

Para resolver el problema de enrole de locomotoras se forma la matriz de tiempos muertos y se obtiene el acoplamiento de peso máximo que determinará el enrole óptimo.

Este resultado del enrole nos dice además cuantas locomotoras son necesarias para una determinada cantidad de trenes y en qué lugares debe cambiarse la locomotora de un tren a otro.

A continuación se aplicará la fórmula de tiempo muerto dada anteriormente al siguiente problema partícular, pero además se considera que los tiempos de estancia en las estaciones es cero ($M_o=M_d=0$), así que la única variable en consideración es la del tiempo de traslado.

Supóngase que Ferrocarriles Nacionales del Pacífico tiene el siguiente horario de 8 trenes que circulan por 5 estaciones, los trenes que se estan considerando son el BURRO y la BALA ambos con una ida de Nogales-Navojoa (N-NV) y un regreso de Navojoa-Nogales (NV-N), además el tren que sale de Mexicali se conecta en Benjamín Hill.

Los trenes se enumeran del 1 al 8, donde los números del 1 al 4 representan al tren bala, de los cuales el 1 y 2 indican el regreso (Nv-N) y los otros dos la ida (N-Nv), de igual manera para los otros cuatro, los trenes 5 y 6 indican el regreso (Nv-N) y los otros dos la ida. Para ambos trenes se dan sus respectivos horarios de llegada y salida.

La tabla 1 indica los horarios de ida N-NV, la cual será leída de arriba hacia abajo, y la tabla 2 indica el regreso NV-N, se leerá de abajo hacia arriba.

TABLA 1

ESTACION		TREN		HORARIO	
		Burro	Bala	Burro	Bala
Mexicali(M)	Salida	T_8	T_4	21:50	10:00
Nogales(N)	Salida	T_7	T_3	7:00	15:30
Benjamin(BH) Hill	LLega	T_7	T_3	9:30	17:44
		T_8	T_4	7:30	17:10
	Salida	T_7	T_3	10:10	18:05
Empalme(E)	LLega	T_7	T_3	14:30	21:40
	Salida	T_7	T_3	15:00	21:55
Navojoa(NV)	LLega	T_7	T_3	18:05	0:23
	Salida	T_7	T_3	18:20	0:28

TABLA 2

ESTACION		TREN		HORARIO	
		Burro	Bala	Burro	Bala
Mexicali(M)	LLega	T_6	T_2	6:25	17:55
Nogales(N)	LLega	T_5	T_1	21:00	11:50
Benjamin(BH) Hill	Salida	T_6	T_2	21:20	10:45
		T_5	T_1	18:30	9:40
	LLega	T_5	T_1	17:56	9:25
Empalme(E)	Salida	T_5	T_1	13:30	5:55
	LLega	T_5	T_1	12:55	5:40
Navojoa(NV)	Salida	T_5	T_1	9:50	3:03
	LLega	T_5	T_1	9:38	2:58

Pero además se dan los tiempos de traslado (en horas) entre una estación y otra las cuales se muestran en la siguiente tabla:

	Estacion				
Estacion	M	N	BH	E	NV
M	0	9.54	7.26	10.76	13.37
N	9.54	0	2.28	5.78	8.38
BH	7.26	2.28	0	3.50	6.11
E	10.76	5.78	3.50	0	2.61
NV	13.37	8.38	6.11	2.61	0

A continuación se procede a formar la matriz de tiempos muertos (dados en hora) de locomotoras que se desocupan en los trenes que llegan y se ocupan en los trenes que estan de salida en las distintas estaciones respectivas.

Por ejemplo, T_1 llega a Navojoa a las 2.96 horas y T_5 sale de Benjamín Hill a 18.5 horas, en este caso el tiempo muerto es 15.54 horas y el tiempo de recorrido entre estas estaciones es 6.11 horas, por tanto se cumple la fórmula de tiempo muerto (no hay necesidad de sumar un múltiplo de 24 horas).

En cambio, T_5 llega a Navojoa a las 9.63 horas y T_1 ocupa una locomotora a las 3.05 horas, de aquí se observa que no se cumple la fórmula del tiempo muerto por lo que hay que sumar un múltiplo de 24 horas, este múltiplo es 24 por lo tanto el tiempo muerto es 17.42 horas.

Utilizando la fórmula de tiempo muerto como en los dos casos anteriores se construye la siguiente matriz de tiempos muertos:

salidas

llegadas

	nv T_1	E T_1	B T_1	B T_2	N T_3	B T_3	E T_3	nv T_3	M T_4	nv T_5	E T_5	B T_5	B T_6	N T_7	B T_7	E T_7	nv T_7	M T_8
nv T_1	0. 09	2.95	30. 7	7.7 9	12. 54	15. 12	18. 95	21. 5	31. 04	6.8 7	10 .5	15 .5	18 .3	28	7. 2	12	15 .3	18 .8
E T_1	21 .3	0.25	4	5. 09	9. 84	12. 42	16. 25	18. 8	28. 34	4. 17	7. 84	12 .8	15 .6	25 .3	4. 50	9. 34	12 .6	16 .1
B T_1	17 .6	20.5	0. 25	1. 34	6. 09	8. 67	12. 50	15. 05	24. 59	24. 42	4. 09	9. 09	11 .9	21 .5	0. 75	5. 59	8. 92	12 .4
N T_1	15 .2	18. 08	21. 83	22. 92	3. 67	6. 25	10. 08	12. 63	22. 17	22	25 .6	6. 67	9. 50	19 .1	22 .3	27 .1	30 .5	10
M T_2	33 .1	12	15. 75	16. 84	21. 59	24. 17	28	30. 55	16. 09	15. 92	19 .5	24 .5	27 .4	13	16 .2	21	24 .4	3. 92
B T_3	9. 32	12. 18	15. 93	17. 02	21. 77	0. 35	4. 18	6. 73	16. 27	16 1	19 .7	0. 77	3. 6	13 .2	16 .4	21 .2	24 .6	28 .1
E T_3	5. 39	8.25	12	13. 09	17. 84	20. 42	0. 25	2.8	12. 34	12. 17	15 .8	20 .8	23 .6	9. 34	12 .5	17 .3	20 .6	27 .1
nv T_3	2. 67	5.53	9. 28	10. 37	15. 12	17. 7	21. 53	0. 08	33. 62	9. 45	13 .1	18 .1	20 .9	30 .6	9. 78	14 .6	17 .9	21 .4
B T_4	9. 89	12. 75	16. 50	17. 59	22. 34	0. 92	4. 75	7.3	16. 84	16. 67	20 .3	1. 34	4. 17	13 .8	17	21 .8	25 .1	28 .6
nv T_5	17 .4	20. 28	24. 03	25. 12	29. 87	8. 45	12. 28	14. 83	24. 37	0.2	3. 87	8. 87	11 .7	21 .3	24 .5	5. 37	8. 7	36 .2
E T_5	14 .1	17	20. 75	21. 84	26. 59	5. 17	9	11. 55	21. 09	20. 92	0. 59	5. 59	8. 42	18	21 .2	2. 09	5. 42	32 .9
B T_5	9. 12	11. 98	15. 73	16. 82	21. 57	0. 15	3. 98	6. 53	16. 07	15. 9	19 .5	0. 57	3. 4	13	16 .2	21	24 .4	27 .9
N T_5	30	8.91	12. 66	13. 75	18. 5	21. 08	24. 91	27. 46	13	12. 83	16 .5	21 .5	24 .3	10	13 .1	18	21 .3	24 .8
M T_6	20 .6	23.5	27. 25	28. 34	33. 09	11. 67	15. 5	18. 05	3. 59	27. 42	31	12	14 .9	24 .5	27 .7	32 .5	35 .9	15 .4
B T_7	17 .5	20.4	0. 15	1. 24	5. 99	8. 57	12. 4	14. 95	24. 49	24. 32	3. 99	8. 99	11 .8	21 .4	0. 65	5. 49	8. 82	12 .3
E T_7	12 .5	15. 41	19. 16	20. 25	25	3. 58	7. 41	9. 96	19. 5	19. 33	23	4	6. 83	16 .5	19 .6	0. 5	3. 83	31 .3
nv T_7	8. 97	11. 83	15. 58	16. 67	21. 42	24	3. 83	6. 38	15. 92	15. 78	19 .4	24 .4	27 .2	12 .9	16	20 .9	0. 25	27 .7
B T_8	19 .5	22. 41	2. 16	3. 25	8	10. 58	14. 41	16. 96	26. 5	26. 33	6	11	13 .8	23 .5	2. 66	7. 5	10 .8	14 .3

A ésta matriz se le aplicará el método Húngaro que se verá en el capítulo 3, para obtener la asignación de locomotoras a trenes con tiempo muerto total mínimo.

4.- METODOS DE SOLUCION DEL PROBLEMA DE ACOPLAMIENTO

Se formularán los problemas de acoplamiento en términos de programación lineal y se discutirán sus métodos de solución, así como la relación con otros problemas.

Sea C_{ij} el peso de la arista (i,j) y X_{ij} la variable de desición, donde

$$X_{ij} = \begin{cases} 1 \text{ si la arista (i,j) es incluida en el acoplamiento} \\ 0 \text{ en caso contrario} \end{cases}$$

Por lo tanto el problema queda formulado como sigue:

$$\text{Min } Z = \sum_{i=1}^{m} \sum_{j=1}^{n} C_{ij} X_{ij}$$

$$\text{s.a} \quad \sum_{i=1}^{m} X_{ij} \leq 1 \qquad \text{para } j=1,2,..,n \qquad (1)$$

$$\sum_{j=1}^{n} X_{ij} \leq 1 \qquad \text{para } i=1,2,..,m \qquad (2)$$

$$X_{ij} = 0 \text{ ó } 1 \quad \forall\, i,j. \qquad (3)$$

El conjunto de restricciones (3) es para que el acoplamiento no sea fraccionado. Los conjuntos de restricciones (1) y (3) exigen que de todas las aristas que llegan a j a lo más una puede estar en el acoplamiento, en cambio, (2) y (3) sólo permiten que de todas las aristas que salen de i a lo más una puede estar en el acoplamiento. Se puede observar que hay mn variables, m+n restricciones y cada variable aparece únicamente en dos restricciones.

El planteamiento es el mismo para el problema de acoplamiento de cardinalidad máxima, excepto que la función objetivo se maximiza y si la arista (i,j) no aparece en la gráfica bipartita el coeficiente de X_{ij} es cero tanto en la función objetivo como en las restricciones, en caso contrario el coeficiente es 1.

Ambos problemas de acoplamiento son problemas de programación lineal 0-1, pero por las características de la matriz de restricciones (totalmente unimodular) se garantiza que el óptimo es entero aún considerando todas las variables continuas $0 \leq X_{ij} \leq 1$, sin embargo resulta muy ineficiente resolver este tipo de problemas por medio del método simplex, ya que su formulación tiene muchas variables y restricciones, además de que la matriz de las restricciones tiene muchos ceros. Por ejemplo si se tiene una matriz de peso de 10x15 la matriz de las restricción sería de tamaño 25x150, con un total de entradas de 3750 de las cuales 300 son 1's y 3450 son 0's con 150 variables, más 25 variables de holgura y 25 restricciones.

Por otro lado los problemas de acoplamiento son un caso particular del problema de transporte con m origenes y n destinos. La oferta $a_i = 1$ para $i = 1,2,..,m$ y la demanda $b_j = 1$ para $j = 1,..,n$. Introduciendo origenes o destinos necesarios para transformarlo al caso cuadrado rxr con restricciones de igualdad (como se indica en el capítulo 3) y resolviendo por el algoritmo de transporte resulta que las restricciones del problema de acoplamiento admiten exactamente r variables positivas en cada solución básica factible y como el número de variables básicas es 2r-1, se tendran r-1 variables básicas en el nivel cero, lo cual conduce a un problema altamente degenerado.

Como se hizo ver anteriormente un acoplamiento en la matriz de pesos $C_{m \times n}$ (con $m < n$) es escoger un elemento de cada renglón en distinta columna. Pudiera pensarse que es fácil obtener el acoplamiento máximo a prueba y error, sin embargo hay $\frac{n!}{(n-m)!}$ acoplamientos (ya que es equivalente a escoger m de las n columnas importando el orden) el cual es un número considerablemente grande. Por ejemplo para m=n=5 existen 120 acoplamientos a los cuales hay que obtener la suma de sus pesos y escoger el de valor máximo.

Esto pone de manifiesto la conveniencia de otros algoritmos que aproveche la estructura de este problema.

Los algoritmos que se utilicen para encontrar un acoplamiento de cardinalidad máxima nos van a servir en el problema de Acoplamiento con peso, y estos algoritmos estan basados en el concepto de trayectoria aumentante, que se verán en el capítulo dos.

CAPITULO II

Un enfoque algorítmico que ha resultado ser útil para resolver el problema de acoplamiento es el de teoría de gráficas ya que un acoplamiento se puede representar mediante una gráfica donde los elementos a formar parejas son los vértices y la posibilidad a formar parejas se representa por aristas.

En este segundo capítulo sólo se estudiarán los resultados necesarios de teoría de gráficas para obtener un método de solución al problema de acoplamiento de cardinalidad máxima. Los principales resultados son los teoremas de Berge, Hall y Köning, tales teoremas proporcionan las bases de los algoritmos que se utilizan en éste trabajo para resolver el problema de acoplamiento. También se da la implementación del algoritmo y se concluye este capítulo resolviendo un ejemplo. Este capítulo es de suma importancia ya que justifica el segundo paso del método Húngaro que se dará en el capítulo 3.

1.- CONCEPTOS BASICOS DE TEORIA DE GRAFICAS

Se darán algunas definiciones sobre teoría de gráficas, ya que es fundamental para plantear y definir importantes resultados sobre acoplamiento.

Definición: Una gráfica G(V,A,f) consta de:
i) Un conjunto finito no vacío $V = V(G)$ cuyos elementos se llaman vértices (nodos o puntos).
ii) Un conjunto $A = A(G)$ cuyos elementos se llaman arista (o líneas).
iii) Una relación $f: A(G) \longrightarrow V(G) x V(G)$, llamada relación de incidencia, ya que a cada arista se le asignan dos vértices.

Ejemplo: La figura 2.1 muestra una gráfica

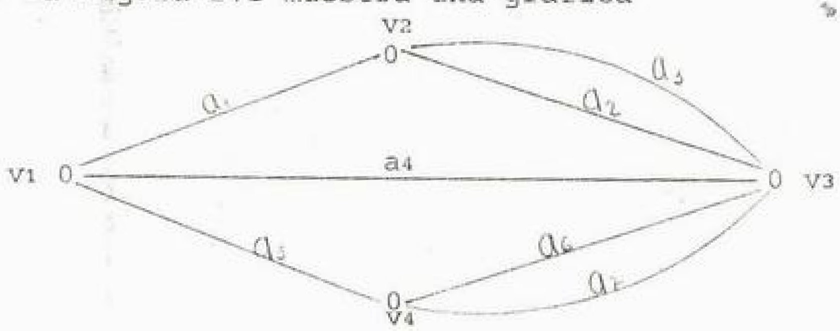


figura 2.1

Donde $V(G) = \{v_1, v_2, v_3, v_4\}$, $A(G) = \{a_1, a_2, a_3, a_4, a_5, a_6, a_7\}$ y $f(a_1) = (v_1, v_2)$, $f(a_2) = (v_2, v_3)$, etc.

Definición: Sea G una gráfica, los extremos de una arista son dos elementos del conjunto de vértices que la determinan. Cuando estos son iguales, se obtiene un lazo. A dos o más arista con los mismos extremos se les llamará arista múltiples (o arista paralelas).

Ejemplo: La gráficas de la figura 2.2 y 2.3 muestran lazos y arista múltiples, respectivamente

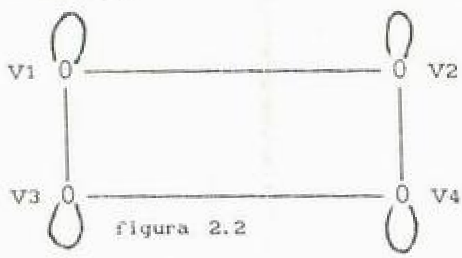


figura 2.2

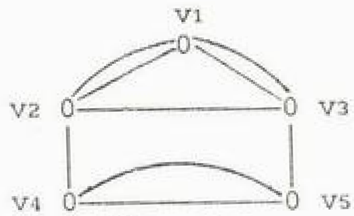


figura 2.3

Definición: Una gráfica simple es aquella donde A(G) es finita pero además no contiene lazos y arista múltiples.

En la figura 2.4 se muestra una gráfica simple

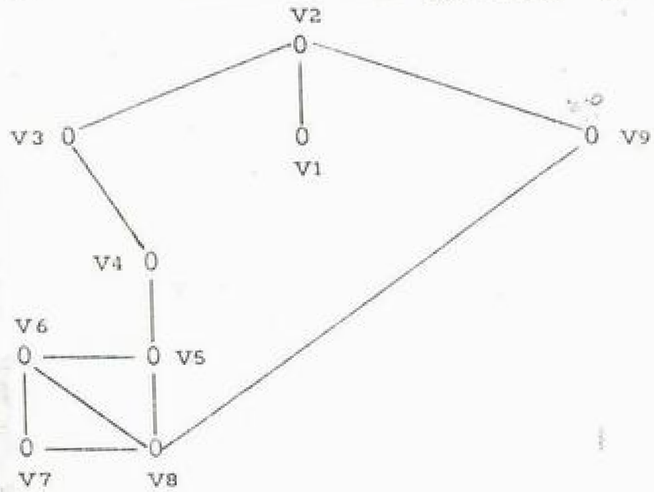


figura 2.4

¡ En este trabajo el término gráfica significará gráfica simple, a menos que se indique lo contrario !

Definición: Sea G(V,A,f) y G'=(V',A',f') dos gráficas. Se dice que G' es subgráfica de G (G'⊂ G) si V'⊂ V, A'⊂ A y f(a)=f'(a) para toda arista a de A'.

Ejemplo: En la figura 2.5 se muestra una gráfica y una subgráfica de ésta en la figura 2.6

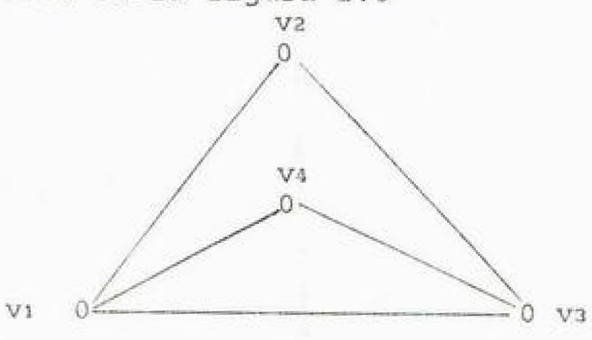


figura 2.5

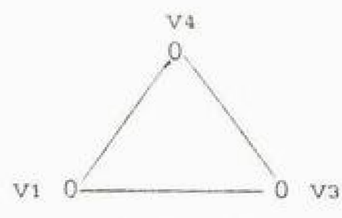


figura 2.6

Definición: Una gráfica $G(V,A)$, es bipartita si existen V_1, $V_2 \subseteq V(G)$ tal que $V_1 \cap V_2 = \emptyset$ y $V(G) = V_1 \cup V_2$ y toda arista de $A(G)$ tiene un extremo en V_1 y otro en V_2. Esta gráfica bipartita se denota $G(V_1, V_2, A)$.

Definición: Dos vértices son adyascentes si existe una arista que los tiene por extremos. Se denota u ady_G v. Dos aristas son adyascentes si tienen un extremo en común y, una arista y un vértice inciden si este último es un extremo de la arista.

Definición: $G(V_1, V_2, A)$ es bipartita completa si cada vértice de V_1 es adyascente a todos los de V_2, la cual se denotará por $K_{n,m}$, donde $n=|V_1|$ y $m=|V_2|$.

Un ejemplo de gráfica bipartita completa es el corredor de bienes raíces visto en el capítulo uno.

Definición: El grado (o valencia) de un vértice $v \in V(G)$, es el número de aristas incidentes a v. Se denota $gr_G(v)$.

Para la gráfica de la figura 2.5 el grado de v_4 es 2, es decir, $gr(v_4)=2$, y el grado de v_3 es 3, es decir $gr(v_3)=3$.

Definición: A una sucesión de vértices $v_1, v_2, \ldots, v_k$, $k \geq 1$, se le llama trayectoria si $(v_{i-1}, v_i) \in A(G)$ y además no se repiten vértices intermedios. La cual se denotará por $T: v_1 \longrightarrow v_2 \longrightarrow \ldots \longrightarrow v_k$.

De la figura 2.4 se obtienen algunas trayectorias tales como: $T_1: v_3 \longrightarrow v_4 \longrightarrow v_5 \longrightarrow v_8 \longrightarrow v_6$ y $T_2: v_3 \longrightarrow v_4 \longrightarrow v_5 \longrightarrow v_6 \longrightarrow v_7 \longrightarrow v_8 \longrightarrow v_9 \longrightarrow v_2 \longrightarrow v_1$.

Definición: Un ciclo es una trayectoria con el mismo vértice inicial y final, es decir, $C: v_o \longrightarrow v_1 \ldots \longrightarrow v_n = v_o$.

Ejemplo: En la gráfica de la figura 2.7, se tiene que

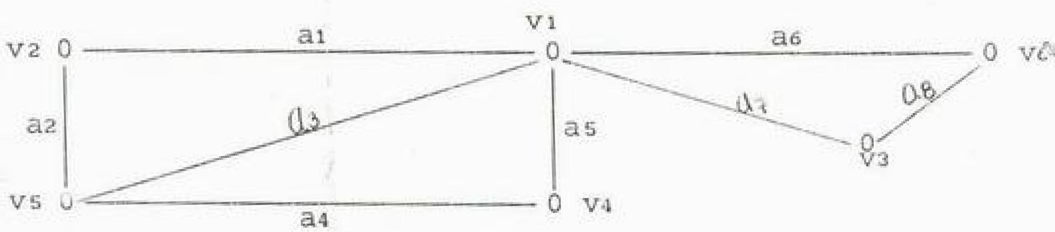


figura 2.7

Son vértices adyascentes v_2 y v_1, v_3 y v_6 etc., a_2 y a_4, a_6 y a_8, son aristas adyascentes, a_2 y v_2 inciden, $C_1: v_1 \longrightarrow v_2 \longrightarrow v_5 \longrightarrow v_4 \longrightarrow v_1$ es un ciclo.

2.- ACOPLAMIENTO

En esta sección se desarrolla la teoría fundamental para la implemantación de algoritmos que resuelven los problemas de acoplamiento (de cardinalidad máxima y con peso). Existen algunos algoritmos que resuelven estos problemas de una manera eficiente. Estos algoritmos estan basados en el concepto de trayectoria aumentante. Ambos problemas se tratarán en el caso bipartito, porque este caso tiene una estructura muy simple. Se ilustrarán las ideas básicas en gráficas generales.

Definición: Dada una gráfica $G(V,A)$, un acoplamiento es un subconjunto M de aristas ($M \subseteq A$) en donde no hay dos de ellas que tengan un vértice en común. Un acoplamiento también puede ser vacío $M=\emptyset$.

Definición: Sea M un acoplamiento en G. Si $u,v \in V(G)$ y $(u,v) \in M$, se dice que u y v estan acoplados. Además si $u \in V(G)$ y u incide en alguna arista de M, se dice que u esta M-saturado, en otro caso u es expuesto.

En la gráfica 2.8, por ejemplo, $M_1=\{(v_2,v_3),(v_4,v_5),(v_6,v_8),(v_7,v_{10})\}$ (mostrada con líneas onduladas) y $M_2=\{(v_1,v_2),(v_3,v_5),(v_4,v_7),(v_6,v_8),(v_9,v_{10})\}$ (mostrada con líneas gruesas) son acoplamientos.

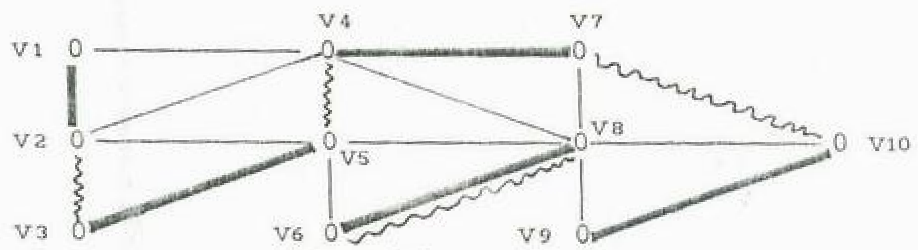


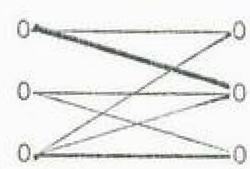
figura 2.8

En la gráfica anterior observe que: v_2,v_3 estan acoplados, v_1 y v_9 son expuestos y v_2,v_4 son saturados con respecto al acoplamiento M_1.

Definición: M es un acoplamiento perfecto, si para todo vértice $u \in V(G)$ u está M-saturado.

Las siguientes gráficas muestran acoplamientos perfectos y no perfectos (ambos se indican con líneas gruesas). Note que en la figura 2.8 M_2 es un acoplamiento perfecto.

Acoplamiento perfecto



Acoplamiento no perfecto

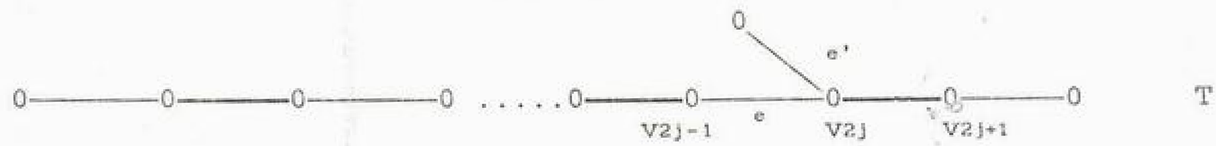
Se puede observar que la cardinalidad de un acoplamiento en $G(V,A)$ está acotado por $|V|/2$, en caso de que exista un acoplamiento con cardinalidad $|V|/2$ es un acoplamiento perfecto, pero sólo basta

En este caso hay dos aristas en M que tienen el mismo vértice en común, esto contradice a que M es un acoplamiento, por lo tanto e, e' no inciden en un mismo vértice y por consiguiente M-T es un acoplamiento. Como se ilustra en la siguiente figura:

2) e,e' $\in$ T-M (en la trayectoria hay que quitar aquellas aristas que están en el acoplamiento).

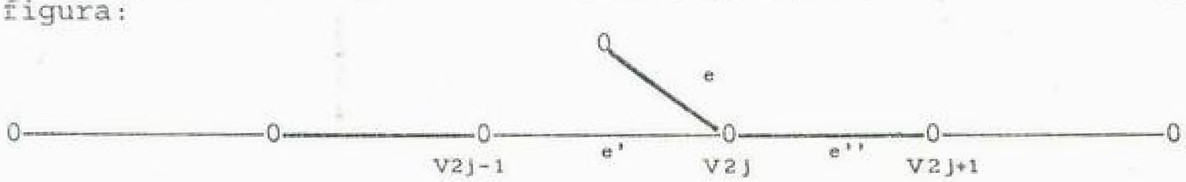
En este caso las aristas en T-M son de la forma (v_{2j-1}, v_{2j}) para $j = 1,2,..,k$ de aquí que dos de ellas no pueden ser incidentes en el mismo vértice porque en una trayectoria alternante no deben existir dos aristas consecutivas que no esten en el acoplamiento, por lo tanto e, e' forman un acoplamiento, debido a esto T-M es un acoplamiento.

Este caso se ilustra a continuación:



3) e $\in$ M-T, e'$\in$ T-M.

En este caso suponga que una arista $e' = (v_{2j-1}, v_{2j})$ en T-M tiene un vértice en común con otra arista e $\in$ M-T. Sin pérdida de generalidad, supóngase que este vértice es v_{2j}, pero v_{2j} es un vértice de la arista $e'' = (v_{2j}, v_{2j+1}) \in M$, entonces dos aristas e'' y e de M tienen un vértice en común, lo que contradice que M es un acoplamiento, de aquí que e y e'' no inciden en un vértice. Por lo tanto M' es un acoplamiento. Este caso se ilustra con la siguiente figura:



Falta por demostrar que $|M'| = |M| + 1$.

Como (M-T), (T-M) son ajenos, entonces $|M'| = |M-T| + |T-M|$, pero además una trayectoria aumentante tiene un número impar de aristas $(2k-1)$ de las cuales $k-1$ estan en M y k no estan en M. De esto se sigue que:

$$|M'| = (|M| - (k-1)) + k$$

$$|M'| = |M| - k + 1 + k = |M| + 1$$

Por lo tanto M' es un acoplamiento con cardinalidad $|M| + 1$. ∎

Por ejemplo, la trayectoria aumentante $T: v_1 \longrightarrow v_4 \longrightarrow v_5 \longrightarrow v_6 \longrightarrow v_8 \longrightarrow v_7 \longrightarrow v_{10} \longrightarrow v_9$ con respecto al acoplamiento M_1 de la figura 2.8, a su conjunto de aristas $T = \{ (v_1,v_4),(v_4,v_5),(v_5,v_6),(v_6,v_8),(v_8,v_7),(v_7,v_{10}),(v_{10},v_9) \}$ y a M_1 se le aplica el lema 1 y se obtiene $M_3=\{ (v_1,v_4),(v_2,v_3),(v_5,v_6),(v_7,v_8),(v_9,v_{10}) \}$. El acoplamiento M_3 es máximo (ya que tiene cardinalidad $|V|/2=5$, es decir, es perfecto) así que no tiene sentido buscar más trayectorias aumentantes con respecto a M_3.

El siguiente teorema establece condiciones necesarias y suficientes para que un acoplamiento M en G, sea de cardinalidad máxima. Este teorema fué demostrado por Berge en 1957.

Teorema (Berge): Un acoplamiento M en G, no es de cardinalidad máxima si y sólo si, G contiene trayectoria M-aumentante.

Demostración: En el lema anterior se demostró en un sentido ($\Leftarrow$), falta sólamente la otra dirección ($\Rightarrow$).

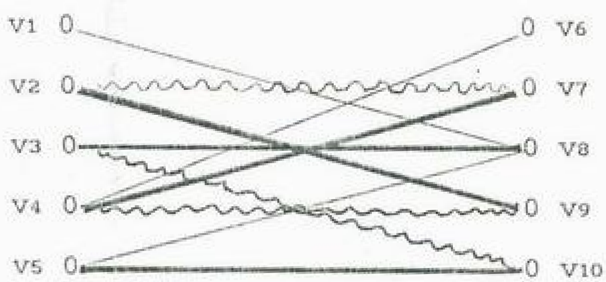
Si M no es de cardinalidad máxima, entonces existe un acoplamiento M' con $|M'|>|M|$. Sea $H=(M-M') \cup (M'-M)$ la diferencia simétrica de M y M'.

Considere la subgráfica $H(V(G),A(H))$, entonces cada $v \in V(G)$ en H tiene $g_r(v) \leq 2$, si el $g_r(v)=2$, entonces una arista incidente a v es de M y la otra de M'. De aquí cada componente de H es un vértice aislado, un ciclo alternante relativo a M y M' que contiene un número par de aristas ó una trayectoria alternante relativa a M y M'.

Por otra parte $|M'|>|M|$ así que existe en H al menos una trayectoria alternante que tiene más elementos en M' que en M por lo cual debe ser aumentante con respecto a M. ∎

Para ilustrar la demostración anterior considere la siguiente gráfica

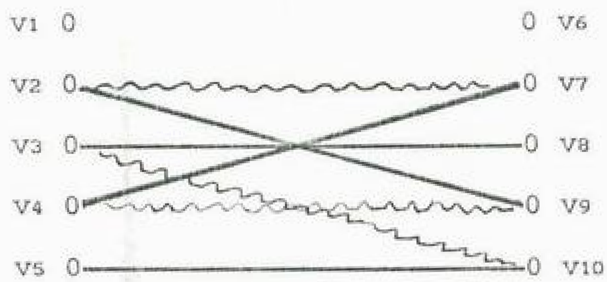
G



De la gráfica G, seleccione los acoplamientos M' (con líneas onduladas) y M (con líneas gruesas) cuya diferencia simétrica es: $H = \{ (v_2,v_7), (v_3,v_{10}),(v_4,v_9),(v_2,v_9),(v_3,v_8),(v_4,v_7),(v_5,v_{10}) \}$.

Una subgráfica de G es $H(V(G),A(H))$, ésta se muestra a continuación:

H



En la gráfica H se aprecian vértices aislados, ciclos de longitud par y trayectorias alternantes con respecto a M', tal trayectoria es aumentante con el acoplamiento M.

El teorema de Berge es fundamental para el desarrollo del algoritmo que resuelve el problema de acoplamiento de cardinalidad máxima.

3.- TEOREMA DE HALL Y ACOPLAMIENTO PERFECTO

En muchas ocasiones se desea hallar acoplamientos que sature a todo vértice de un subconjunto de la gráfica, tal es el caso del problema de asignación, el problema del matrimonio, etc. , donde las gráficas correspondientes son bipartitas, sin embargo, éste acoplamiento no siempre existe. En esta sección se determinarán las condiciones necesarias y suficientes para la existencia de tales acoplamientos, dados por Hall en 1935. Antes de establecerlas se necesita la siguiente definición.

Definición: Sea S un subconjunto de vértices de V(G), se define el conjunto de vecinos de S en G, como el conjunto de todos los vértices adyascentes a algún vértice de S (excepto los de S). El cual se denota $V(S)$.

Ejemplo: Si de la figura 2.7 se escoge el conjunto $S = \{v_2, v_4, v_3\}$ el conjunto de vecinos es $V(S) = \{v_5, v_1, v_6\}$.

Teorema (Hall): Sea $G(V_1, V_2, A)$ una gráfica bipartita, entonces G contiene un acoplamiento que satura a todo vértice de V_1 si y sólo si $|V(S)| \geq |S| \;\; \forall \;\; S \subseteq V_1$.

Demostración: Supóngase que G contiene un acoplamiento que satura a todo vértice de V_1 y sea $S \subseteq V_1$. Por demostrar que $|V(S)| \geq |S| \;\; \forall$ $S \subseteq V_1$.

Ya que M es un acoplamiento que satura a todos los vértices de V_1, entonces todos los vértices de S están acoplados bajo M, con vértices distintos de $V(S)$, así $|V(S)| \geq |S| \;\; \forall \; S \subseteq V_1$.

La suficiencia se hará por contradicción. Sea G una gráfica bipartita con partición (V_1, V_2), supóngase que G no contiene un acoplamiento que satura a todo vértice de V_1 y $|V(S)| \geq |S|$ $\forall$ $S \subseteq V_1$.

Sea M un acoplamiento máximo en G, por hipótesis M no satura a todos los vértices de V_1. Sea $u \in V_1$ un vértice expuesto y Z el conjunto de todos los vértices conectados a u por trayectorias M-alternantes. Como M es un acoplamiento de cardinalidad máxima, por el teorema de Berge, G no contiene trayectorias M-aumentante, de donde se puede inferir que u es el único vértice expuesto en Z.

Sea $S = Z \cap V_1$ y $T = Z \cap V_2$ (ver figura 2.13, donde el acoplamiento M se muestra con líneas gruesas). Donde los vértices en $S-\{u\}$ están acoplados bajo M con los vértices de T, por lo tanto $|T| = |S| - 1$.

Como todo vértice en $V(S)$ está conectado a u por trayectorias M-alternantes, se tiene que $V(S) = T$, esto implica $|V(S)| = |S| - 1 < |S|$ de aquí $|V(S)| < |S|$, lo que contradice la hipótesis, por consiguiente M satura a todo los vértices de V_1. ■

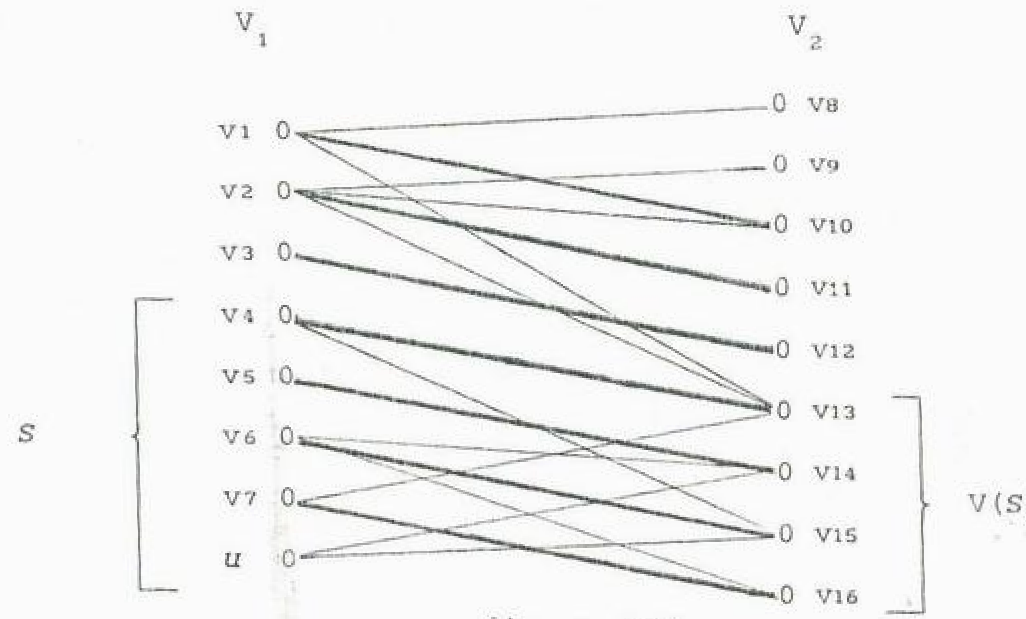


figura 2.13

Ejemplo: En la siguiente gráfica, si $S=\{v2, v3\}$ no se cumple las condiciones del teorema de Hall, ya que el conjunto de vecinos es $V(S)=\{v4\}$ y por tanto $|S| > |V(S)|$.

figura 2.14

Del teorema de Hall se deduce el siguiente corolario que proporciona las condiciones necesarias y suficientes par la existencia de un sistema de representantes distintos de R_i $i=1,2,..,k$ (definido en el capítulo 1) muy conocido en análisis combinatorio.

Corolario: La colección de subconjuntos de R, $\Gamma=\{R_1,R_2,..,R_k\}$ tiene un sistema de representantes distintos si y sólo si

$$|Q| \leq \left| \bigcup_{i\in Q} R_i \right| \quad \forall\ Q \subseteq \{\ 1,2,..,k\ \}.$$

Demostración: Para ponerlo en terminos del teorema de Hall considere lo siguiente: Sea $R = \{r_1, r_2, ..., r_n\}$, $R_i \subseteq R$, la gráfica $G(V_1,V_2,A)$ con $V_1=\Gamma$, $V_2=R$ y las aristas $(R_i,r_j)\in A$ si $r_j \in R_i$. Además haciendo $Q \subseteq \{1,2,3,...,k\}$ (conjunto de subíndices de la colección de R_i). Entonces para $S=\{R_i\ /\ i\in Q\}$ se tiene que

$$V(S) = \bigcup_{i\in Q} R_i$$

Por el teorema de Hall para que exista un sistema de representantes distintos es necesario y suficiente que

$$|\ S\ | \leq |V(S)|$$

es decir

$$|Q| \leq \left| \bigcup_{i\in Q} R_i \right| \qquad \blacksquare$$

Definición: Una gráfica es p-regular si todos sus vértices son de grado p.

Un ejemplo de gráfica p-regular es la calendarización de vuelos visto en el capítulo 1, donde $g_r(v_i) = 5$ para $i=1,2,...,10$.

Corolario: Si G es una gráfica bipartita p-regular con $p > 0$, entonces G tiene un acoplamiento perfecto.

Demostración: Sea $G(V_1,V_2,A)$ p-regular. Como G es p-regular, se tiene que $p|V_1|=A(G)=p|V_2|$ pero $p > 0$ así que $|V_1|=|V_2|$. Sea $S \subseteq V_1$ y $A_1(G)$, $A_2(G)$ los conjuntos de aristas incidentes en S y $V(S)$ respectivamente. Por definición de vecinos $A_1(G) \subseteq A_2(G)$ se tiene:

$$p|S|\leq|A_1(G)|\leq|A_2(G)|\leq p|V(S)|$$

De aqui se sigue $|S|\leq|V(S)|$ y por el teorema de Hall, G tiene un acoplamiento M que satura a cada vártice en V_1, pero $|V_1|=|V_2|$ así que M es un acoplamiento perfecto. $\blacksquare$

Como puede observarse la demostración del teorema de Hall son las bases de un buen algoritmo para encontrar un acoplamiento máximo en gráficas bipartitas. Este algoritmo se dará más adelante.

4.- PROBLEMA DE CUBIERTA MINIMA

Un problema asociado al problema de acoplamiento máximo es el problema de cubierta mínima, el cual establece la relación entre el número de aristas de un acoplamiento máximo y el conjunto mínimo de vértices que son extremos de todas las aristas. Para su formulación se define lo siguiente:

Definición: Sea $K \leq V(G)$. K es una cubierta de G, si toda arista de G tiene al menos un extremo en K.

Definición: K es una cubierta mínima de G, si no existe una cubierta K', tal que $|K'| < |K|$.

La figura 2.15, muestra una cubierta y una cubierta mínima de una gráfica:

(a) cubierta K=(V1,V2,V3,V4,V5)

(b) cubierta mínima K=(V2,V4,V5,V6)

figura 2.15

Observe que si K es una cubierta y M es un acoplamiento ambos de G, entonces K contiene al menos uno de los extremos de la arista de M. Así, para cualquier acoplamiento M y cualquier cubierta K, se tiene que $|M| \leq |K|$. Particularmente, si M'es un acoplamiento máximo y K' es una cubierta mínima entonces:

$$|M'| \leq |K'| \tag{2.16}$$

Lema: Sea M un acoplamiento y K una cubierta tal que $|M|=|K|$. Entonces M es un acoplamiento máximo y K es una cubierta mínima.

Demostración: Si M' es un acoplamiento máximo y K' es una cubierta mínima, entonces por 2.16 se tiene

$$|M| \leq |M'| \leq |K'| \leq |K|$$

por hipótesis $|M|=|K|$, así que $|M|=|M'|$ y $|K|=|K'|$ por lo tanto M es un acoplamiento máximo y K una cubierta mínima. ∎

En general, la igualdad no se cumple en 2.16, por ejemplo en la figura 2.15, el acoplamiento máximo es de cardinalidad 3 ($M=\{(v_1,v_5),(v_2,v_6),(v_3,v_4)\}$), mientras que la cubierta mínima es 4. Sin embargo, si G es bipartita se tiene $|M'|=|K'|$. Este resultado, debido a Köning y a Egervary quienes lo demostrarón en 1931, está estrechamente relacionado con el teorema de Hall.

Teorema (Köning): En una gráfica bipartita $G(V_1, V_2, A)$ si M es un acoplamiento máximo y K una cubierta mínima entonces $|M| = |K|$.

Demostración:
i) Si M es un acoplamiento máximo que satura a todo V_1 entonces $|M| = |V_1|$, considere la cubierta $K=V_1$ de aquí se tiene que $|M| = |K|$ por el lema anterior K en mínima, el teorema se cumple.

ii) Si M es un acoplamiento máximo que no satura a todo V_1 entonces por el teorema Hall existe $S \subseteq V_1$ tal que $|V(S)| < |S|$.

Por ser M acoplamiento máximo no existe tarayectoria M-aumentante, es decir, la trayectoria termina en el conjunto V_1.

Sea $S=V_1 \cap Z$, donde Z es la unión de todos los vértices de las trayectorias M-alternantes conectadas a cada vértice expuesto en V_1, como en la demostración del teorema Hall. Se tiene que $V(S)$ esta M-saturado porque de no ser así se encontraría una trayectoria M-aumentante y esto contradice la hipótesis.

También se tiene que los vértices de S^c estan M-saturados, pero la arista que lo satura tiene su extremo en $V(S)^c$.

De aquí se tiene que:

$|V(S)| < |S|$, $|S^c| \leq |V(S)^c|$ y además $|V(S)| + |S^c| = |M|$

Una cubierta es $K = V(S) \cup S^c$ con $|K| = |M|$ y por el lema anterior, K en mínima. ∎

La cubierta mínima de este teorema se utiliza en el segundo paso del método Húngaro, el cual se dará en el capítulo 3.

5.- DESCOMPOSICION EN CADENAS

El problema de la descomposición en cadenas (descrito en el capítulo 1) se puede reducir a uno de acoplamiento máximo de la siguiente manera:

Sean V_1 y V_2 dos copias del conjunto N ($V_1=V_2=N$). A la copia en V_1 de un elemento $t \in N$ se le denotará como t', y a la copia en V_2 como t''. Se forma la gráfica bipartita $G(V_1, V_2, A)$, de modo que una arista (i,j) pertenezca a A siempre y cuando se cumpla $t_i < t_j$.

Encontrar un acoplamiento máximo en la gráfica G así construida es lo mismo que hallar el subconjunto más grande de parejas de elementos comparables en N, donde no hay dos elementos que se repitan como primer elemento o como segundo elemento en las parejas.

Las parejas del acoplamiento máximo se pueden "encadenar" identificando los elementos comunes a ellas. Los elementos restantes de N se pueden considerar como cadenas de longitud uno, y así se obtiene la descomposición en cadenas de N que se buscaba.

Cada descomposición en cadenas D del conjunto N le corresponde un único acoplamiento M en la gráfica bipartita asociada. Más aún, bajo esta correspondencia cada cadena de longitud m en D se asocia con m-1 parejas en M.

El siguente teorema muestra lo mencionado anteriormente:

Teorema: Si D es una descomposición en cadenas en N y M un acoplamiento en G entonces $|M|$ es máximo si $|D|$ es mínima.

Demostración: Sean $C_1, C_2, \dots, C_k$ las cadenas en D, $\bigcup_{i=1}^{k} C_i = N$ y $|D| = k$ de aquí se tiene que:

$$|M| = (|C_1|-1) + (|C_2|-1) + \dots + (|C_k|-1)$$

$$|M| = |C_1| + |C_2| + \dots + |C_k| - k$$

$$|M| = |N| - |D|$$

Como $|N|$ es constante entonces el valor $|M|$ es máximo a costa de hacer al valor $|D|$ mínimo. ■

Esto justifica la reducción del problema de descomposición en cadenas a uno de acoplamiento máximo.

Teorema (Teorema de Dilworth): Si N es un conjunto finito con un orden parcial estricto, entonces: min $(|D|$: D es descomposición en cadenas de N$)$=max$(|A|$: A es una anticadena en N $)$.

Demostración: Sean D una descomposición en cadenas y A una anticadena en N. Entonces se cumple $|D| \geq |A|$, puesto que ninguna cadena en D contiene más de un elemento en A.

Ahora, supóngase que D es una descomposición en cadenas de cardinalidad mínima, y sea M el acoplamiento asociado tal como se describio anteriormente, el cual cumple: $|M| = |N| - |D|$.

De acuerdo al teorema de Köning, existe una cubierta mínima de vértices K que satisface $|K| = |M| = |N| - |D|$. Si n_1, n_2 son elementos de N que cumplen $n_1 < n_2$, al ser K cubierta de vértices de N debe ocurrir (por la forma en que se contruyó M) que K contiene: la primera copia n'_1 de n_1, o la segunda copia n''_2 de n_2.

Llamando A al conjunto de elementos n en N que no tienen ni la copia n' ni la n'' en K, se puede ver que A es una anticadena, y que se cumple:

$|N|-|A|=$ número de elementos n en N con n' o n'' en K.

Para continuar, se mostrará que no hay elementos n en N que tengan simultáneamente n' y n'' en K, de modo que el valor $|N|-|D|$ es igual a $|K|$ exactamente.

Supóngase que hay un elemento n en N con n' y n'' en K. Dado que K es de cardinalidad mínima, no se puede eliminar de K ninguno de esos elementos sin alterar su condición de óptimo. Por tanto debe tenerse un elemento n en N tal que $n_1 < n$ (para tener la arista (n', n''_1)) y n' no pertenezca a K. Análogamente, debe existir n_2 en N tal que $n < n_2$ y n''_2 no esté en K. Entonces, por transitividad del orden en N debe ocurrir $n_1 < n_2$, lo que implica tener la arista (n'_1, n''_2) sin incidir en ningún vértice de la cubierta K, lo que es contrdictorio $|N|-|A| = |K| = |N|-|D|$, con lo que A es una anticadena que cumple $|A|=|D|$. Esto completa la demostración. ■

Teorema: Sea N un conjunto finito con un orden parcial estricto, y supóngase que la cadena más larga en N es de longitud *l*. Entonces se puede hacer una partición de N en m anticadenas disjuntas.

Demostracion: La prueba es por inducción sobre la longitud '*l*' de la cadena.

Cuando $l=1$, N contiene un sólo vértice, y claramente se tiene una anticadena.

Supóngase ahora que la proposición vale para $l=k$ (hipótesis de inducción). Sea N conjunto con orden parcial estricto y con la longitud de sus cadenas más largas igual a k+1. Sea *A* el conjunto de elementos maximales en N, es decir, si $m \in A$, entonces para todo $x \in N$ ocurre que $x < m$. Es claro que *A* es una anticadena no vacía. Considérese entonces al conjunto parcialmente ordenado: $N-A$, con el orden heredado de N. Puesto que en $N-A$ no existen cadenas de longitud k+1, la longitud de las cadenas más largas en $N-A$ es a lo más k. Por otro lado, si la longitud de las cadenas más largas en $N-A$ es menor de k, *A* debe contener dos o más elementos maximales pertenecientes a una misma cadena, lo que contradice la construcción de *A*. Por tanto, se concluye que las cadenas más largas de $N-A$ son de longitud k, y por la hipótesis de inducción, $N-A$ puede particionarse en k cadenas disjuntas si a estas k cadenas disjuntas se añade *A*, se obtiene una partición de N en k+1 cadenas disjuntas. ■

La figura (a) muestra un conjunto N con un orden parçial igual a la descomposición en cadenas del capítulo I, donde la cadena de mayor longitud $C=\{1,3,4,6\}$ tiene por cardinalidad 4; en la figura (b) las líneas punteadas indican la partición en anticadenas ajenas.

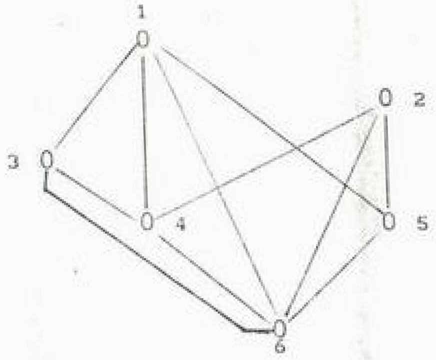


figura (a)

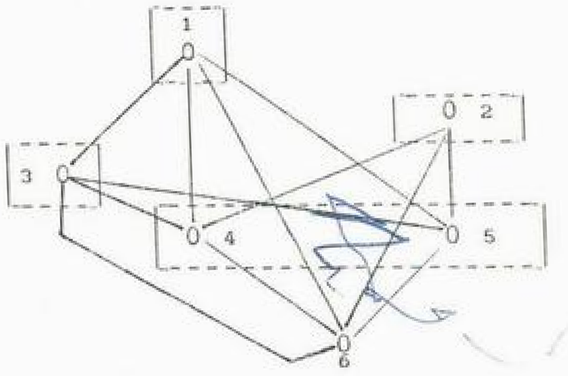


figura (b)

6.- IMPLEMENTACION COMPUTACIONAL

Los algoritmos de acoplamiento comienzan con un acoplamiento (por ejemplo, el acoplamiento vacío), y repetidamente encuentran trayectorias aumentantes. Todos los algoritmos conocidos de acoplamiento estan basados en esta idea.

6.1.- Representación de una gráfica en la computadora

Uno de los aspectos más importantes en la manipulación de una gráfica es su representación en computadora. Existen diferentes estructuras de datos para gráficas, pero en este trabajo sólo se utilizarán la representación de la **lista ligada hacia adelante** (para el problema de acoplamiento de cardinalidad máxima) y la **matriz con peso** (para el problema de acoplamiento con peso) que se explicará en el capítulo 3.

Si en el proceso sobre la gráfica no se requiere agregar ni quitar aristas, entonces simplemente se guardan las aristas secuencialmente. (Todas las aristas que salen del vértice k se guardan inmediatamente después de todas las aristas que salen de la arista k-1). Por ejemplo, la lista ligada hacia adelante para la gráfica de la figura 2.14 se muestra a continuación usando tres arreglos:

i	Apuntador	Vértice_fin
1	1	4
2	4	5
3	5	6
4	6	4
		4

Observe que todas las aristas que salen del vértice i son: (i, vértice_fin[apuntador[i]), (i, vértice_fin[apuntador[i]+1) ,..., (i, vértice_fin[apuntador[i+1]-1). Note que el caso i=n se toma cuidadosamente agregando una celda artificial al arreglo *apuntador* que contiene el valor m+1.

Así es como se va almacenar los datos de una gráfica bipartita en la computadora.

6.2.- Árbol alternante

Para encontrar una trayectoria aumentante con respecto a un acoplamiento M, comience con cualquier vértice expuesto *r*. Si existe un vértice expuesto *s* adyascente a *r*, se ha encontrado una trayectoria aumentante y se agrega la arista (*r*,*s*) al acoplamiento actual M, si no existe ningún vértice expuesto adyascente a *r*, entonces todos los vértices adyascentes a r estan acoplados, se hará crecer un árbol enraizado en *r* (ver la gráfica de la figura 2.17). Sea *r* la raíz situada en un nivel 0 y todos los vértices adyascentes $a_1,a_2,...,a_q$, en un nivel 1 (los vértices $a_1,a_2,...,a_q$, son saturados). Ahora incluya en este árbol todas las aristas acopladas incidentes en $a_1,a_2,...,a_q$, y sitúe sus respectivos vértices terminales $b_1,b_2,...,b_q$, en el nivel 2. A continuación, todos los vértices adyascentes a los b_i (estos vértices no están contenidos en el árbol) se colocan a un nivel 3. Se continúa en cada uno de los b_i como al principio en *r*. Un árbol construído en la forma anterior se le llama árbol alternante.

En la figura 2.17 se muestra lo descrito anteriormente

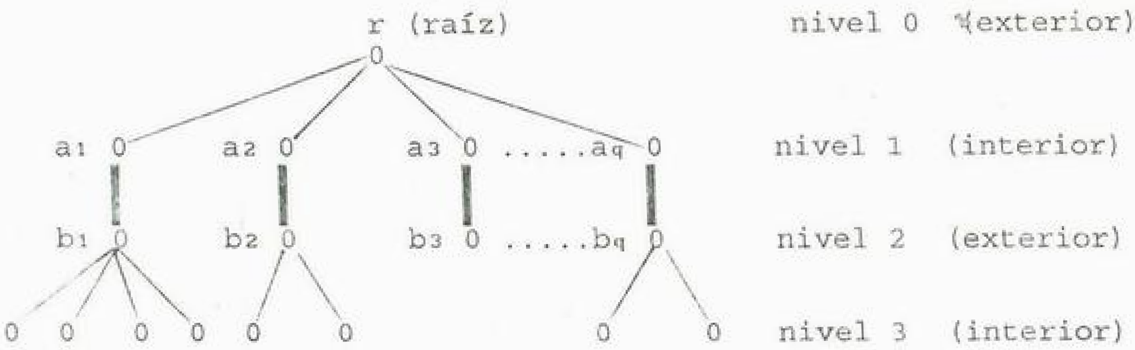


figura 2.17 árbol alternante

Más formalmente se da la definición de árbol alternante

Defenición: Un árbol alternante con respecto a un acoplamiento M, es un árbol enraizado en un vértice expuesto que tiene la propiedad que todas las trayectorias que principian en la raíz son trayectorias alternantes (existe sólamente una trayectoria entre cada par de vértices en el árbol).

En el árbol alternante, se llamarán vértices interiores a los vértices situados en los niveles impares y vértices exteriores a los vértices situados en los niveles pares. Conforme el árbol alternante crece, si se encuentra en un vértice interior, entonces se agrega al árbol la correspondiente arista acoplada. Por lo tanto todo vértice interior es de grado 2. Pero en los vértices exteriores *x* pueden surgir diferentes casos. Si se examina una arista (*x*,*y*) incidente en *x*, si *y* es un vértice expuesto y no está contenido en el árbol, se habrá encontrado una trayectoria aumentante. De otra manera, si y es un vértice saturado con vértice acoplado z (ninguno de los dos todavía en el árbol), se agregan ambas aristas (x,y) y (y,z) al árbol, haciendo y un vértice interior y z un vértice exterior. Si y está en el árbol como un vértice interior (las dos formas en que esto puede pasar se muestra en la figura 2.18(a)), el ciclo formado agregando la arista (x,y) al árbol es un ciclo de longitud par (contiene un número par de aristas). Puede mostrarse que la arista (x,y) puede ignorarse ya que no crea una trayectoria aumentante adicional. Finalmente, si y está en el árbol como vértice exterior, se habrá formado un ciclo de longitud impar (vea figura 2.18 (b)). (Esto no puede ocurrir en las gráficas bipartitas).

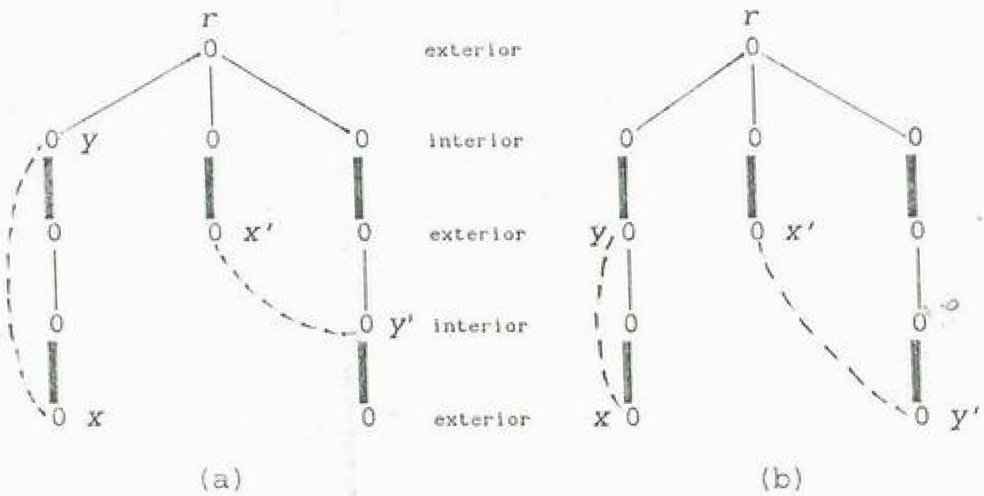


figura 2.18 ciclos formados en el árbol alternante
(a) ciclos pares (b) ciclos impares

6.3.- Aumentando el árbol alternante

El árbol alternante debe crecer a lo ancho comenzando en un vértice expuesto r arbitrario. Como ya se mensionó al principio, el crecimiento comienza en un vértice exterior. (En todos los vértices interiores se agregan las correspondientes aristas acopladas). Para llevar a cabo la búsqueda a lo ancho se define una lista Q que contiene todos los vértices exteriores en el árbol que no han sido explorados. Se toma un vértice x de la cabeza de la lista, se elimina de la lista y se explora, examinando todos los vértices adyascentes a x (vea figura 2.17).

Para registrar los vértices del árbol que estan como raíz o como vértices interiores, se guardan en un arreglo booleano denominado *noarbol* de tamaño n (n es el número de vértices en la gráfica), al inicio cada una de sus componentes son verdaderas excepto la raíz r. Esto es:

noarbol$(r) \longleftarrow$ falso;
para todo $j \in N$, $j \neq r$ has *noarbol* $\longleftarrow$ verdadero

Cada vez que un nuevo vértice j en el árbol se hace interior se registra como *noarbol*(j) $\longleftarrow$ falso. Recuerde que siempre que entra al árbol un vértice interior, su vértice adyascente acoplado se hace automáticamente vértice exterior.

Otro arreglo, llamado ***abuelo***, de tamaño n se utiliza para obtener la trayectoria hacia atrás hasta la raíz cuando se encuentre una trayectoria aumentante y el acoplamiento se agranda. Para un vértice exterior w, *abuelo*(w)=u si hay una trayectoria u $\longrightarrow$ j $\longrightarrow$ w en el árbol alternante de u a w, y (j,w) en una arista acoplada.

6.4.- Acoplamiento inicial

Aunque se puede iniciar el algoritmo con cualquier acoplamiento inicial, incluso puede ser vacío, es más eficiente comenzar con un acoplamiento tan grande como sea posible. Un acoplamiento inicial "razonablemente grande", puede obtenerse rápidamente, considerando cada vértice expuesto v en V_1 de la gráfica y acoplándolo con el primer vértice expuesto adyascente a él.

La variable *expo* denota el número de vértice expuestos en V_1 de la gráfica y el arreglo *acopla* de tamaño n indica el acoplamiento en la gráfica. Esto es, si (i,j) es una arista en el acoplamiento, entonces *acopla*(i)=j y *acopla*(j)=i. Para un vértice expuesto x, *acopla*(x)=0.

El siguiente algoritmo da este acoplamiento inicial:

```
                    Acoplamiento inicial
Comienza
     para toda v∈V₁ has acopla (v) ←— 0;
     expo ←— n; (*número de vértices expuestos en V₁*)
     para todo v∈V₁ has
     comienza
          si acopla (u)=0 entonces
          comienza
               v ←— un vértice expuesto adyascente a u;
               acopla(u) ←— v;
               acopla(v) ←— u;
               expo ←— expo-1
          fin
     fin
fin
```

6.5.- Acoplamiento máximo

Comience con un acoplamiento inicial obtenido de la forma descrita anteriormente y contenido en el arreglo *acopla*; también se tendrá en la variable *expo* el número de vértices expuestos. Se continúa en la búsqueda de un acoplamiento máximo sólamente si *expo*≥1. En este caso, se toma el primer vértice expuesto *r* encontrado y comienze el crecimiento del árbol alternante desde *r* por el método de búsqueda a lo ancho.

Cuando se trate de expander desde un vértice exterior x (tomado de la cabeza de la lista Q) vea su vértice adyascente y, aquí existen varios casos:

Si *noarbol*(y)=falso, ignore la arista (x,y) (y proceda con el siguiente vértice adyascente a x), esto se debe a que si noarbol(y)=falso implica que y está como vértice interior en el árbol y la arista (x,y) esta en el acoplamiento actual o forma un ciclo par con el árbol, en ambos casos se ignora (x,y). Así, sólo se consideran aquellos vértices adyascentes a x cuyo noarbol tenga valor verdadero.

Si *noarbol*(y)=verdadero, el vértice y puede ser acoplado o expuesto:

Si y es expuesto (*acopla*(y)=0), se ha encontrado una trayectoria aumentante, se procede hacer el acoplamiento de cardinalidad mayor, trazando la trayectoria desde éste vértice a la raíz e intercambiando las aristas que estan en el acoplamiento con aristas que no lo estan. Enseguida se abandona el árbol alternante actual pero continúa con el ciclo principal haciendo crecer un árbol alternante desde el siguiente vértice expuesto.

Si y no es expuesto, primero se debe estar seguro que y no es un ancestro de x en el árbol; por que si no, puede encontrarse una trayectoria aumentante falsa. Esto se logra checando desde la trayectoria de x a la raíz usando el arreglo abuelo. En el caso que y sea un ancestro de x, no se hace nada. Si y no es un ancestro de x, y y esta acoplado con el vértice z, se deben incorporar al árbol (es decir, insertar en la lista al vértice exterior z, y hacer *abuelo*(z) $\longleftarrow x$; *noarbol*(y) $\longleftarrow$ falso).

Se termina el ciclo de construcción del árbol cuando se encuentra una trayectoria alternante (la variable booleana *encontrar*=verdadera) o cuando no hay mas vértices exteriores que expander (cuando la lista Q está vacía).

El algoritmo de acoplamiento finaliza (el acopamiento obtenido es máximo) cuando el número de vértices expuestos es menor que 1 o cuando todo vértice expuesto existente ha pasado a formar parte de la raíz de algún árbol alternante sin haber obtenido una trayectoria aumentante.

Este algoritmo de acoplamiento máximo puede describirse más formalmente y de manera compacta como sigue:

Acoplamiento máximo

```
comienza
  empieza con un acopalmiento inicial;
  para todo r∈V₁ has
    si(acopla(r)=0) y (expo≥1) entonces
    comienza (* crece árbol alternante enraizado en r *)
       para toda v∈V has noarbol(v) ←—— verdadero;
       noarbol(r) ←—— falso;
       inicializa Q conteniendo unicamente a r;
       encontrar ←—— falso; (* trayectoria aumentante no encontrada *)
```

```
repite (* hasta que Q=ø o encontrar=verdadero *)
  borra vértice x de Q; (* x encabeza la lista en Q *)
  mientras no encontrar has
  comienza
     para todo y adyascente x has
        si noarbol(y)=verdadero entonces
        comienza
           agrandar acoplamiento;
           expo ←— expo-1;
           encontrar ←— verdadero
        fin
        sino si acopla(y)≠x entonces
        comienza
          z ←— acopla(y);
          si(x≠r) o (z no es ancestro de x) entonces
          comienza (* agrega aristas (x,y) y (y,z) al arbol *)
             noarbol(y) ←— falso;
             abuelo(z) ←— x;
             insertar z en la cola de Q
          fin
   fin (* mientras *)
 hasta encontrar o (Q=ø)
fin
fin
```

6.6.- Acoplamiento Aumentante

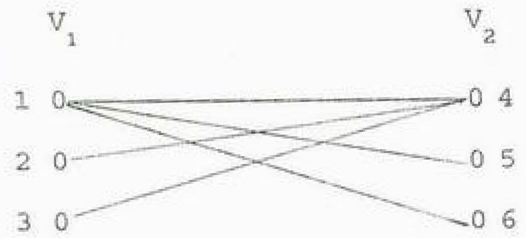
El acoplamiento es el más grande cuando se ha encontrado un vértice expuesto y adyascente al vértice actual. El procedimiento para aumentar la trayectoria alternante de *y* a *r* y convertir cada arista no acoplada a una arista acoplada y viceverza, es como sigue:

Acoplamiento aumentante

```
acopla(y) ←— x;
repite
   next ←— acopla(x)  (* next es el acoplado más viejo de x *)
   acopla(x) ←— y  (* y es el nuevo acoplado de x *)
   si next ≠ 0 entonces (* si x no es la raíz *)
   comienza
      x ←— abuelo(x);
      acopla(next) ←— x
      y ←— next
   fin
hasta next=0.
```

6.7.- EJEMPLO

A la siguiente gráfica se le aplicaran los procedimientos anteriores para encontrar un acoplamiento de cardinalidad máxima.



Antes de comenzar a resolverlo, se dará el formato del archivo de entrada:

1) La primera línea debe contener algún título de preferente relacionado con el problema.

2) La sugunda línea esta compuesta por: número de vértices, número de aristas y número de vértices de V_1, separdos por un espacio.

3) Cada una de las siguientes líneas corresponde a las aristas escribiendo primero el vértice de V_1 después su vértice adyascente en V_2.

4) El archivo de entrada puede tomar cualquier nombre de preferencia que sea el nombre del problema con que se este trabajando con cualquier extensión pero de preferencia que sea .dat (en éste ejemplo, gráfica.dat).

Formato del archivo de salida :

1) El archivo de salida puede tomar cualquier nombre de preferencia que sea el nombre del archivo de entrada con cualquier extensión, pero de preferencia que sea .sal (en este caso gráfica.sal), el archivo de salida contiene la siguiente información:

a) un letrero que dice: "problema de acoplamiento máximo".
b) título del problema.
c) número de vértices expuestos.
d) acoplamiento máximo.
e) cubierta mínima.

De lo anterior se tienen algunas observaciones:

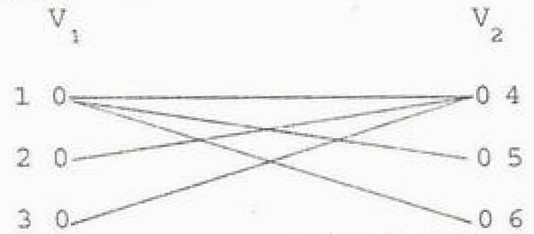
1).- No hay pérdida de generalidad si los algoritmos de acoplamiento máximo comienzan con el conjunto V_1 o V_2 pero, por la forma en que se programaron en necesario que V_1 sea el conjunto con $|V_1| \le |V_2|$, para que los algoritmos tengan sentido.

2.- Cuando se de la entrada de cualquier archivo es conveniente escoger los vértices de V_2 consecutivos a los de V_1.

Por tanto el archivo de entrada es:

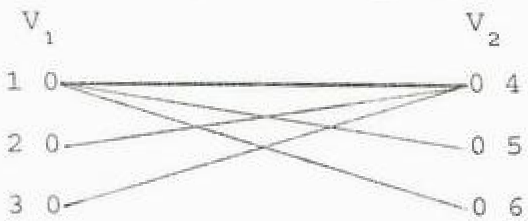
```
gráfica
6 5 3
1 4
1 5
1 6
2 4
3 4
```

donde la gráfica bipartita es:



En la cual se desea encontrar un acoplamiento de cardinalidad máxima.

Primero se encuentra un acoplamiento inicial, el cual se obtiene por el algoritmo 6.4, el acoplamiento inicial se muestra en la siguiente gráfica con línea gruesa, pero además en el arreglo *acopla* se da dicho acoplamiento y el total de vértices expuestos en V_1.



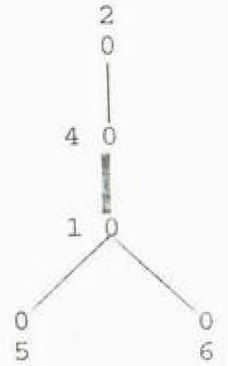
	1	2	3
acopla =	4	0	0

expo = 2

Observe que hay dos vértices expuestos en V_1 los cuales son 2 y 5, entonces se comienza el aumento del árbol alternante enraízado en el primer vértice expuesto 2, de acuerdo con el procedimiento 6.3. La arista (2,4) se añade al árbol con su respectiva arista acoplada (con línea gruesa), a continuación se muestra el árbol construido hasta aquí con los arreglos *noarbol*, *abuelo* y *Q* (lista):

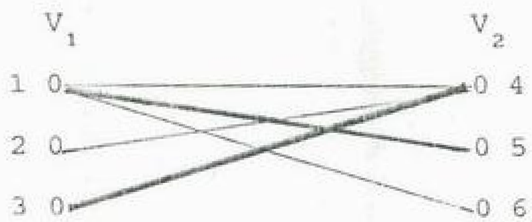
	1	2	3	4	5	6
noarbol	v	f	v	f	v	v
abuelo	2					
Q	2	1				

Siguiendo, el vértice 2 se borra de la cabeza de Q y ahora se analiza el 1, los vecinos de 1 son $\{5,6\}$ observe que *noarbol*(5)=*noarbol*(6)=verdadero así que se añaden al árbol las aristas (1,5) y (1,6), se muestra el aumento del árbol hasta aquí:



	1	2	3	4	5	6
noarbol	v	f	v	f	f	f
abuelo	2				4	4
Q	2	1	5	6		

Como puede observarse *noarbol*(5) es verdadero y *acopla*(5) es cero entonces se ha encontrado un vértice expuesto, por lo tanto una trayectoria aumentante y por el procedimiento 6.6 se tiene un acoplamiento de cardinalidad uno más que el anterior .El acoplamiento obtenido (indicado con líneas gruesas) se ilustra a continuación con su respectivo arreglo *acopla* y el total de vértices expuestos (*expo*):



	1	2	3
acopla =	5	4	0

expo = 1

Otra vez se vuelve hacer lo anterior pero ahora con éste acoplamiento y el árbol anterior se borra, es decir, se comienza con un nuevo árbol.

Observe que el primer y único vértice expuesto en V_1 es 3, entonces se construye el árbol alternante enraízado en 3, de acuerdo con el procedimiento 6.3. La arista (3,4) se añade al árbol con su respectiva arista acoplada (línea gruesa), a continuación se muestra el árbol construido hasta aqui con los arreglos *noarbol*, *abuelo* y *Q*:

	1	2	3	4	5	6
noarbol	v	v	f	f	v	v
abuelo		3				
Q	3	2				

Siguiedo, el vértice 3 se borra de la cabeza de Q y ahora se analiza el vértice 2, como puede observarse éste vértice no tiene vecinos así que se termina la construcción del árbol. Por lo tanto el acoplamiento obtenido es máximo, el cual se da a continuación en el arreglo *acopla* asi como también el número de vértices expuestos y la cubierta mínima.

$$acopla = \begin{array}{ccc} 1 & 2 & 3 \\ \boxed{5} & \boxed{4} & \boxed{0} \end{array}$$

$$expo = 1$$

$$K = \{1,4\}$$

El procedimiento de acoplamiento máximo se utilizará como subrutina en el segundo paso del método Húngaro el cual se dará en el suguiente capítulo.

CAPITULO III

1.- INTRODUCCION

Al problema de acoplamiento con peso se le llama problema de asignación como se hizo saber en el capítulo uno. Debido a que el algoritmo que resuelve éste problema sólo utiliza acoplamiento en uno de sus pasos y trabaja más sobre la matriz de pesos (los pesos pueden ser beneficio, tiempo, etc.), se prefiere referirse a éste problema como problema de asignación.

En este capítulo se desarrolla el método Húngaro, así como la justificación de sus pasos y su convergencia. El método Húngaro es tipo primal-dual, ya que comienza con una solución dual factible, sigue iterando manteniendo factibilidad dual y holgura complementaria, obteniendo el óptimo hasta alcanzar factibilidad primal. Además se da la transformación de un problema de minimización a uno de maximización y también se da la implementación computacional de este método y se resuelve un ejemplo.

2.- PARADOJA AL PROBLEMA DE ASIGNACION

A continuación se resuelve en forma empírica una serie de problemas, conduciendo con este razonamiento a una paradoja. Lo anterior es con el objeto de hacer notar que el sentido común suele fallar aunque la instancia del problema a resolver sea pequeña.

Una compañía tiene 3 tareas a realizar y hay 3 aspirantes disponibles. El problema de la compañía es el de decidir como asignar cada aspirante a una tarea y a cada tarea un aspirante de modo que maximice el beneficio, para ello se considera la siguiente matriz de beneficio:

	R_1	R_2	R_3
A_1	10	8	7
A_2	4	8	3
A_3	6	4	9

(3.1)

Observe que A_1 es mejor en R_1, A_2 es mejor en R_2 y A_3 es mejor en R_3, se muestra en la siguiente matriz con (*):

	R_1	R_2	R_3
A_1	10 *	8	7
A_2	4	8 *	3
A_3	6	4	9 *

(3.2)

Esta es una asignación ya que se ha escogido un elemento de cada renglón en distinta columna (como se expresó en el capítulo uno). Puede observarse que no se puede obtener otra de valor más grande por lo cual esta asignación es óptima. En este caso se dice que la matriz tiene una asignación por renglón máximo.

Para ésta matriz fué casual que cada aspirante fuese mejor en una tarea diferente. En general uno no puede esperar que esto suceda, como lo ilustra la siguiente matriz:

	R_1	R_2	R_3
A_1	10 *	8	6
A_2	8	4	7 *
A_3	6	9 *	4

(3.3)

Aquí no es posible lograr asignar a cada persona el trabajo que él hace mejor porque A_1 y A_2 son mejores en R_1 y esto no puede ser asignación. En cambio, se probará escoger al mejor hombre para cada tarea. Así, el mejor hombre para R_1 es A_1, para R_2 es A_3 y para R_3 es A_3 (mostrada con *), la cual es una asignación y es óptima. En este caso se dice que la matriz tiene asignación de columna máxima.

Generalmente una matriz de asignación no necesariamente tiene un renglón o columna máxima, por ejemplo la siguiente matriz:

	R_1	R_2	R_3
A_1	10	8	7 *
A_2	8 *	4	3
A_3	9	6 *	4

(3.4)

Si cada persona es asignado a la tarea que mejor hace, entonces cada uno fué asignado a R_1, y si cada tarea es dado al mejor hombre, entonces A_1 fué asignado a las tres tareas. En ambos casos ninguna es asignación, la asignación óptima para este caso es señalada con * en (3.4) cuyo valor es 7 + 8 + 6 = 21.

Ahora supóngase que se le da oportunidad a un cuarto aspirante con los siguientes beneficios:

	R_1	R_2	R_3
A_4	2	3	5

(3.5)

La pregunta que surge es ¿El cuarto aspirante sustituirá a alguno de los tres de la asignación óptima?.

Observe que en la asignación óptima actual en R_1 se obtiene un beneficio de 8 mientras que A_4 ofrece 2, en R_2 se obtiene un beneficio de 6 mientras que A_4 tiene 3, y en R_3 se obtiene 7 mientras que A_4 tiene 5. En todas las tareas A_4 ofrece menos beneficio que la asignación óptima, por lo que a simple vista se puede concluir que A_4 no debe sustituir a nadie.

Pero este tipo de regla esta equivocada, ya que si A_4 es asignada a R_3, A_1 a R_2 y A_3 a R_1, el valor de la asignación es 8+0+9+5=22 (mostrada con * en la tabla 3.6), es mejor que la asignación óptima anterior ya que el valor es 21 (mostrado con ■ en 3.6). En la siguiente tabla se muestran los cuatro aspirantes con sus respectivos beneficios:

	R_1	R_2	R_3
A_1	10	8 *	7■
A_2	8■	4	3
A_3	9 *	6■	4
A_4	2	3	5 *

(3.6)

Con esto se contesta la pregunta hecha anteriormente, por lo tanto es mejor sustutuir A_4 por A_2 aunque A_4 sólo tiene mayor beneficio que A_2 en R_3.

De aquí se ve que el sentido común no siempre es útil al intentar resolver estos problemas. Parece viable contratar a las personas con beneficios muy altos, pero como se vio, hasta la persona menos capacitada puede elevar el valor de la asignación óptima.

Por lo que es necesario el desarrollo de métodos analíticos para resolver este problema. Uno de los métodos que da muy buenos resultados es el Algoritmo Húngaro que está basado en resultados de acoplamiento y los que a continuación se desarrollan.

3.- CONCEPTOS BASICOS

En esta sección se dan algunas definiciones necesarias para el desarrollo del método.

Definición: Una asignación "a" es una permutación $[i_1, i_2, i_3, .., i_n]$ de los enteros 1 , 2,.., n. Esto se lee como sigue: en la asignación "a" el aspirante A_1 es asignado a la tarea R_{i_1}, el aspirante A_2 es asignado a la tarea R_{i_2}, el aspirante A_j es asignado a la tarea R_{i_j}, etc.

El entero en la i-ésima posición de la permutación es la columna asignada al i-ésimo renglón.

Definición: Una matriz de asignación es la matriz de pesos asociada al problema de asignación $C = [C_{ij}]$.

Definición: Sea C una matriz nxn de un problema de asignación. El valor de una asignación $a=[i_1, i_2, i_3, .., i_n]$ denotada por V(a) es:

$$V(a) = C_{1,i_1} + C_{2,i_2} + \ldots + C_{n,i_n}.$$

Definición: Se dice que una asignación es de valor máximo (mínimo) si entre todas las posibilidades no hay otra de valor más grande (más chico). Se dice que una asignación es óptima si es de valor máximo (o de valor mínimo).

Definición: Dos matrices de asignación C y C' son equivalentes si ambas tienen la misma asignación óptima.

Teorema: Si una constante se suma a todas las entradas de cualquier renglón o columna de una matriz de asignación, la matriz obtenida es equivalente a la original.

Demostración: Sean Z y Z' funciones objetivos de un problema de asignación, donde Z es la función objetivo original y Z' es la función que resulta de sumar una constante α_i a todas las entradas de cada renglón i y sumar β_j a todas las entradas de cada columna j.

Por demostrar que C y C' son equivalentes.

$$Z' = \sum_{i=1}^{n} \sum_{j=1}^{n} C'_{ij} X_{ij}$$

$$Z' = \sum_{i=1}^{n} \sum_{j=1}^{n} (C_{ij} - \alpha_i - \beta_j) X_{ij}$$

$$Z' = \sum_{i=1}^{n} \sum_{j=1}^{n} C_{ij} X_{ij} - \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i X_{ij} - \sum_{i=1}^{n} \sum_{j=1}^{n} \beta_j X_{ij}$$

$$Z' = \sum_{i=1}^{n} \sum_{j=1}^{n} C_{ij} X_{ij} - \sum_{i=1}^{n} \alpha_i \sum_{j=1}^{n} X_{ij} - \sum_{j=1}^{n} \beta_j \sum_{i=1}^{n} X_{ij}$$

Como $\sum_{j=1}^{n} X_{ij} = \sum_{i=1}^{n} X_{ij} = 1$ son las restricciones de un problema de asignación se tiene que

$$Z' = \sum_{i=1}^{n} \sum_{j=1}^{n} C_{ij} X_{ij} - \sum_{i=1}^{n} \alpha_i - \sum_{=1}^{n} \beta_j$$

$Z' = Z$ + Constante.

De aquí el mínimo (ó máximo) para Z' se obtiene con la solución X que da el mínimo (ó máximo) para Z, por lo tanto C y C' son equivalentes. ∎

Se ilustrara lo anterior con las siguientes matrices de costos:

	R_1	R_2	R_3
A_1	0	2	3
A_2	0	4	5
A_3	0	3	5

(3.7)

	R_1	R_2	R_3
A_1	0	0	0
A_2	0	2	2
A_3	0	1	2

(3.8)

Donde (3.8) se obtiene de (3.7) restandole 2 a la segunda columna y 3 a la tercera. Las posibles asignaciones para ambas matrices son las siguientes: $a_1=[1,2,3]$, $a_2=[1,3,2]$, $a_3=[2,1,3]$, $a_4=[2,3,1]$, $a_5=[3,2,1]$ y $a_6=[3,1,2]$, los valores de éstas asignaciones se muestran a continuación donde la primera columna representa los valores de (3.7) y la segunda de (3.8):

$V(a_1) = 9$	$V(a_1) = 4$
$V(a_2) = 8$	$V(a_2) = 3$
$V(a_3) = 7$	$V(a_3) = 2$
$V(a_4) = 7$	$V(a_4) = 2$
$V(a_5) = 7$	$V(a_5) = 2$
$V(a_6) = 6$	$V(a_6) = 1$

Observe que las dos matrices tienen la misma asignación óptima la cual es a_6 si se esta minimizando (a_1 si es maximización) por lo tanto (3.7) y (3.8) son equivalentes, además el valor de esta asignación difieren por el valor total que se le resta.

Debido a que en la justificación de ciertos pasos del algoritmo Húngaro es necesario algunos resultados de programación lineal, en la siguiente sección se formula el problema de asignación en éstos terminos.

4.- PROBLEMA DE ASIGNACION Y SU DUAL (CASO CUADRADO)

En general en un problema de asignación con n tareas y m aspirantes, se tiene que n y m no necesariamente son iguales. Puede haber más aspirantes que tareas (m>n), en este caso la compañía solo empleará a los aspirantes que generen un beneficio total máximo. También pueden existir más tareas que aspirantes (n>m) en cuyo caso la compañía solo asignará aspirantes a las m-tareas que den un beneficio total máximo. Existe una manera, para que todo problema de asignación sea reducido al caso "cuadrado", es decir, el número de tareas y aspirantes sean iguales.

Así, si hay más aspirantes que tareas (m>n) a la matriz se le añaden m-n columnas, cuyas entradas son beneficios cero si se está maximizando (o son muy grandes si se está minimizando) obteniéndose una matriz de mxm con m-n tareas ficticias adicionales que pueden considerarse como tareas existentes improductivas. En otro caso , si hay más tareas que aspirantes (n>m) a la matriz se añaden n-m renglones, cuyas entradas son cero cuando se está maximizando (o son muy grandes si se está minimizando) obteniéndose así una matriz de nxn con n-m aspirantes ficticios adicionales.

La formulación del problema de asignación (considerado en general en el capítulo 1) para el caso cuadrado queda de la siguiente manera·

$$\text{Min } Z = \sum_{i=1}^{n} \sum_{j=1}^{n} C_{ij} X_{ij}$$

$$\text{sujeto a:} \quad \sum_{i=1}^{n} X_{ij} = 1 \text{ para } j=1,2,..,n \qquad (1)$$

$$\sum_{j=1}^{n} X_{ij} = 1 \text{ para } i=1,2,..,n \qquad (2)$$

$$X_{ij} = 1 \text{ ó } 0 \text{ para } i,j=1,2,..,n. \qquad (3)$$

Ahora las resticciones (1) y (3) significan que a cada aspirante se le asigne una tarea diferente, en la matriz de pesos se refiere a que cada renglón se le asigne un peso de diferente columna. Las restricciones (2) y (3) significan que cada tarea debe ser realizada por diferentes aspirantes, lo cual en la matriz de pesos se refiere a que cada columna se le asigne un peso de diferente renglón.

Dedido a que la matriz de restricciones es totálmente unimodular, se garantiza que el óptimo es entero aún considerando todas las variables continuas $0 \le X_{ij} \le 1$, además por las restricciones (1) y (2) sólo basta considerar que las variables sean no negativas. Por lo tanto al cambiar las restricciones (3) por $X_{ij} \ge 0$ para $i,j = 1,2,..,n$ resulta un problema lineal continúo con el mismo óptimo.

Su correspondiente dual considerando las restricciones de no negatividad resulta ser como sigue:

$$\text{Max } W = \sum_{i=1}^{n} u_i + \sum_{j=1}^{n} v_j$$

$$\text{sujeto a:} \quad u_i + v_j \le C_{ij} \quad \text{para } i,j = 1,2,..,n$$

$$u_i, v_j \text{ no restringidas para } i,j = 1,2,..,n$$

Si se encuentra la solución del caso cuadrado con variables no negativas, también se obtiene la solución del problema original. Apartir de ahora, a menos que se especifíque lo contrario se restringirá al caso cuadrado no negativo, debido a que el método que se va a utilizar trabaja con éste caso.

5.- FUNDAMENTOS TEORICOS

En esta sección se justificará que el método Húngaro es tipo primal-dual, ya que comienza con una solución dual factible, sigue iterando manteniendo factibilidad dual y holgura complementaria, obteniendo el óptimo hasta lograr factibilidad primal.

5.1.- Solución inicial dual factible (paso 1 del método Húngaro)

Considere un problema de asignación cuya matriz de peso es $C=[C_{ij}]_{n \times n}$ a la cual se le aplicará el método Húngaro.

Sea $\hat{u}_i$ el mínimo de los C_{ij} en el renglón i, y $\hat{v}_j$ el mínimo de los $\{C_{ij} - \hat{u}_i\}$ en la columna j como lo especifíca el paso 1. Es decir

$$\hat{u}_i = \underset{1 \le j \le n}{\text{mínimo}} \{C_{ij}\} \quad \text{para } i = 1,2,..,n$$

$$\hat{v}_j = \underset{1 \le i \le n}{\text{mínimo}} \{C_{ij} - \hat{u}_i\} \quad \text{para } j = 1,2,..,n$$

Como puede observarse $\hat{u}_i$, $\hat{v}_j$ satisfacen las restricciones duales.

La matriz reducida obtenida en el paso uno $\hat{C}_{ij} = C_{ij} - \hat{u}_i - \hat{v}_j$ tiene un cero en todos los renglones y en todas las columnas y todos sus elementos serán no negativos. De hecho, esta matriz reducida es la matriz de las variables de holgura duales.

Supóngase que se puede encontrar un conjunto factible de las x_{ij} tales que cada x_{ij} con valor 1 está asociado con una celda cero de la matriz reducida. Entonces por la holgura complementaria se concluye que se tiene una solución óptima, ésta solución es óptima porque se han encontrado n ceros, es decir, se tiene un 1 por renglón pero en distintas columnas y un 1 por columna pero en distinto renglón. Revisando las restricciones del problema de asignación, es claro que debe tenerse exactamente una x_{ij} igual a 1 en cada renglón, y exactamente una x_{ij} igual a 1 en cada columna, por lo tanto, en una solución factible hay exactamente n de las x_{ij} iguales a 1 y las restantes serán cero.

Se ilustrará con un ejemplo las ideas anteriores. Considérese la siguiente matriz de coeficientes de costo para un problema de asignación

				renglón mínimo
3	2	5	4	2
0	1	2	3	0
4	1	-1	3	-1
2	5	3	4	2

Restando cada elemento de cada renglón el mínimo en ese renglón, se obtiene la siguiente matriz

1	0	3	2	
0	1	2	3	
5	2	0	4	
0	3	1	2	
0	0	0	2	columna mínima

Restando el mínimo de cada columna en la nueva matriz a cada elemento en la columna se obtiene la matriz reducida siguiente.

1	0 *	3	0
0 *	1	2	1
5	2	0 *	2
0	3	1	0 *

Ahora, si se toma $x'_{12} = x'_{21} = x'_{33} = x'_{44} = 1$ (mostrados con *) y si se toman todos los x'_{ij} restantes iguales a cero, se tiene entonces una solución factible con los x_{ij} positivos asociados a celdas cero en la matriz reducida, produciendo así una solución óptima.

No siempre es fácil encontrar una solución óptima. Por ejemplo, la matriz 3.7 se reduce a 3.8. Aquí no es posible tomar tres x_{ij} iguales a 1 de tal forma que todos los tres x_{ij} positivos ocurran en celdas cero y que no ocurran dos x_{ij} positivos en el mismo renglón o columna.

5.2.- Cubierta mínima de ceros (paso 2)

Nótese que para la matriz reducida anterior, el número máximo de las x_{ij} que ocurren en la celda cero que se pueden igualar a 1 sin que dos de los x_{ij} positivos ocurran en el mismo renglón o columna es 2. Por ejemplo, podría tomarse $x_{12} = x_{21} = 1$, $x_{12} = x_{13} = 1$, $x_{13} = x_{21} = 1$, ó $x_{13} = x_{31} = 1$. En este caso, el máximo número de celdas con C_{ij} cero tales que dos de estas celdas no están en el mismo renglón o columna es 2. Las celdas correspondientes se llaman independientes. Nótese también que si se trazara un conjunto de líneas sobres los renglones y columnas por cubrir los cero de tal manera que hay al menos una línea sobre cada cero, entonces el número mínimo de tales líneas para esta matriz es 2, a saber, una línea sobre la columna 1 y una línea sobre el renglón 1.

	R1		R2	R3
A1	0		0	0
A2	0		2	2
A3	0		1	2

$x_{ij} = 1$

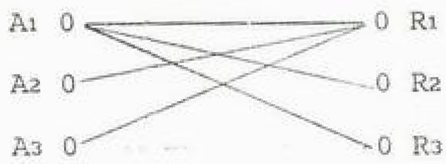
$(x_{ij} = 0)$

$x_{ij} = 0$

Se ve en este ejemplo que el número máximo de celdas cero independientes y el número mínimos de líneas requeridas para cubrir los cero son iguales. En general, lo anterior es cierto para cualquier matriz, esto se puede observar obteniendo la cubierta mínima (teorema de Köning) de la gráfica bipartita $G(V_1, V_2, A)$ asociada a cada nueva matriz de peso $C=[C_{ij}]$ de la siguiente manera: V_1 es el conjunto de renglones, V_2 conjunto de columnas y conjunto de aristas $(i,j) \in A$, $i \in V_1$, $j \in V_2$ sólo si $C_{ij}=0$.

Ejemplo: Se ilustra la siguiente matriz con su correspondiente gráfica bipartita:

	R_1	R_2	R_3
A_1	0	0	0
A_2	0	2	2
A_3	0	1	2



Como puede observarse en caso de que $|V_1|=|V_2|$ un acoplamiento perfecto es una asignación.

Lo anterior es necesario en el paso dos del método Húngaro.

5.3.- Nueva solución dual factible (paso 3)

Supóngase que aún no se ha obtenido la solución óptima, es decir, no se puede encontrar un conjunto factible de las x_{ij} positivos de entre las celdas cero de la matriz reducida. Considérese la matriz cubierta, es decir, la matriz reducida con los ceros cubiertos por el menor número de líneas. Sea K el número de líneas requeridas. Así mismo, sea $F = \{i_1, i_2, .., i_p\}$ el conjunto de renglones cubiertos y $H = \{j_1, j_2, .., j_q\}$ el conjunto de columnas cubiertas. Defínase $F^c = M - F$ y $H^c = M - H$ en donde $M = \{1,2,..,n\}$. Finalmente, sea p el número de renglones en F, y q el número de columnas en H. Entonces $K = p+q$.

Sea C_0 el mínimo elemento no cubierto, es decir,

$$C_0 = \underset{i\in F^c,\, j\in H^c}{\text{mínimo}} \{\hat{C}_{ij}\} > 0$$

Se puede demostrar fácilmente que una nueva solución dual factible $(\overline{u_i}, \overline{v_j})$ está dada por:

$$\begin{aligned}
\overline{u_i} &= \hat{u}_i - C_0 && i\in F^c\\
\overline{u_i} &= \hat{u}_i && i\in F\\
\overline{v_j} &= \hat{v}_j && j\in H^c\\
\overline{v_j} &= \hat{v}_j + C_0 && j\in H
\end{aligned} \tag{4}$$

En la matriz reducida esto es equivalente a restar C_0 de cada renglón no cubierto y sumar C_0 a cada columna cubierta. Otra forma de ver esto es que C_0 se resta de cada elemento no cubierto y se suma a cada elemento cubierto por dos líneas. La nueva matriz reducida de coeficientes de costos tienen elementos no negativos y un cero en cada renglón y columna.

Para la matriz de 3x3 anterior se tiene C_0 = mínimo {2,2,2,1} = 1 y la nueva matriz reducida de pesos está dada por

	R_1	R_2	R_3
A_1	1	0	0^*
A_2	0^*	1	1
A_3	0	0^*	1

Obsérvese que ahora existe un conjunto factible de varias x_{ij} con los x_{ij} positivos asociados con las celdas cero (variables de holgura duales cero).

Nótese que se alcanza factibilidad primal, la factibilidad dual se mantiene (pues los elementos de la matriz reducida de costos son no negativos), y finalmente, se cumple la holgura complementaria (pues x_{ij} = 1 sólo si la holgura dual correspondiente es cero). Por lo tanto, se cumplen las condiciones de holgura y se cuenta con la solución óptima $x'_{13} = x'_{21} = x'_{32} = 1$ (mostradas con *) todas las otras x'_{ij} son iguales a cero.

Teorema: Si $\overline{u}_i$, $\overline{v}_j$ se definen como (4), entonces las nuevas $\overline{u}_i$, $\overline{v}_j$ son una solución dual factible ($\overline{u}_i + \overline{v}_j \leq C_{ij}$ $\forall$ i,j).

Demostración: Sea $\hat{C}_{ij} = C_{ij} - \hat{u}_i - \hat{v}_j$ donde

$$\hat{u}_i + \hat{v}_j \leq C_{ij} \ \forall \ i,j \tag{5}$$

por lo tanto $\hat{C}_{ij} \geq 0$. Como $C_0 = \min\{ \hat{C}_{ij} \} \geq 0$, $i \in F^c, j \in H^c$ entonces

$$0 \leq C_0 \leq \hat{C}_{ij}, \ i \in F^c, j \in H^c \tag{6}$$

Por demostrar que las nuevas $\overline{u}_i$, $\overline{v}_j$ también satisfecen las restricciones del dual.

Los subíndices i,j=1,2,...,n se particionan en $\{i \in F, \ j \in H^c\}$, $\{i \in F^c, \ j \in H\}$, $\{i \in F^c, \ j \in H^c\}$, $\{i \in F, \ j \in H\}$.

a) Para $i \in F$ y $j \in H^c$ se tiene que $\overline{u}_i = \hat{u}_i$ y $\overline{v}_j = \hat{v}_j$ entonces

$\overline{u}_i + \overline{v}_j = \hat{u}_i + \hat{v}_j \leq C_{ij}$ por (5) $\Rightarrow \overline{u}_i + \overline{v}_j \leq C_{ij}\ \ \forall\ i,j.$

b) Para $i \in F^c$ y $j \in H$ se tiene que $\overline{u}_i = \hat{u}_i - C_0$ y $\overline{v}_j = \hat{v}_j + C_0$ entonces

$\overline{u}_i + \overline{v}_j = \hat{u}_i - C_0 + \hat{v}_j + C_0 = \overline{u}_i + \overline{v}_j \leq C_{ij}$ por (5) $\Rightarrow \overline{u}_i + \overline{v}_j \leq C_{ij}$ $\forall\ i,j.$

c) Para $i \in F^c$ y $j \in H^c$ se tiene que $\overline{u}_i = \hat{u}_i - C_0$ y $\overline{v}_j = \hat{v}_j$ entonces

$\overline{u}_i + \overline{v}_j = \hat{u}_i - C_0 + \hat{v}_j = \hat{u}_i + \hat{v}_j - C_0 \leq C_{ij}$ por (5) y (6) $\Rightarrow \overline{u}_i + \overline{v}_j \leq$ C_{ij} $\forall\ i,j.$

d) Para $i \in F$ y $j \in H$ se tiene que $\overline{u}_i = \hat{u}_i$ y $\overline{v}_j = \hat{v}_j + C_0$ entonces

$\overline{u}_i + \overline{v}_j = \hat{u}_i + \hat{v}_j + C_0 \leq \hat{u}_i + \hat{v}_j + \hat{C}_{ij} = C_{ij}$ por (6) $\Rightarrow \overline{u}_i + \overline{v}_j \leq$ C_{ij} $\forall\ i,j.$

Por tanto $\overline{u}_i$, $\overline{v}_j$ son soluciones factibles del problema dual.

Si se puede encontrar un conjunto de las U, V y X factibles a sus respectivos problemas, que satisfagan la holgura complementaria, estas serán óptimas.

De manera equivalente se demuestra lo anterior pero ahora con el problema primal.

Corolario: Si a una matriz de peso $C = [C_{ij}]$ se le aplica el paso 3 del método Húngaro resulta una matriz equivalente.

Demostración: Sea $C=[C_{ij}]$ la matriz equivalente que se obtiene del paso 1. Sea $F=\{i_1,i_2,..,i_p\}$ y $H=\{j_1,j_2,..,j_q\}$ los conjuntos de índices de renglones y columnas cubiertos en el paso 2, $p,q < n$.

En el paso 3 en forma desglosada se realizan los siguientes pasos:

i) Se obtiene $C_0 = \min\ \{c_{ij}\ /\ i \in F^c$ y $j \in H^c\ \}$

ii) Se resta C_0 a todos los elementos de C obteniéndose la matriz equivalente C'. C' tiene negativos en donde C tiene ceros (renglón o columna cubierta).

iii) Para obtener la matriz equivalente positiva C'', en C' se suma C_0 a los renglones y columnas cubiertas.

Como puede observarse estos tres pasos son equivalentes al paso 3 del algoritmo. Por lo tanto el paso 3 obtiene la matriz equivalente C''.

$$C''_{ij} = \begin{cases} C_{ij} & \text{si} \quad i\in F \ , \ j\in H^c \\ C_{ij} & \text{si} \quad i\in F^c \ , \ j\in H \\ C_{ij} + C_0 & \text{si} \quad i\in F \ , \ j\in H \\ C_{ij} - C_0 & \text{si} \quad i\in F^c \ , \ j\in H^c \end{cases} \tag{7}$$

La cual por el teorema anterior es equivalente a C. ∎

6.- METODO HUNGARO

El método Húngaro tiene como propósito resolver el problema de asignación, el nombre se debe a que fué desarrollado por los matemáticos Húngaros Köning y Egervary en 1931. Este método es sumamente fácil ya que mediante suma o resta de constantes a renglones o columnas se obtiene una nueva matriz equivalente que tiene asignación óptima y con menor valor de asignación. El método termina cuando se ha obtenido la matriz equivalente con asignación de valor cero.

Este método es únicamente para el caso de minimización, más adelante se verá cómo un problema de maximización se puede resolver por medio de éste.

Paso 1: Obtención de ceros

En cada renglón, obtener el menor elemento y réstelo a todos los elementos del mismo renglón. Hacer lo mismo por columna.

Paso 2: Búsqueda de una solución óptima

Obtener la cubierta mínima de la gráfica bipartita asociada a la nueva matriz. Si la cardinalidad de la cubierta es igual a la dimensión de la matriz, terminar el acoplamiento determina la asignación óptima, en caso contrario ir al paso 3.

Paso 3: Obtención y desplazamiento de ceros

Cubrir los renglones y columnas asociadas a la cubierta mínima. Obtener el menor elemento que este en renglón y columna no cubiertos. Restéselo a los elementos del renglón y columna no cubiertos y suméselo a los elementos del renglón y columna cubiertos. Continuar en el paso 2.

Como puede observarse el método termina en el paso 2. En el paso 1, se obtiene por lo menos un cero por renglón y otro por columna y en el paso 3, los elementos cubiertos una vez quedan igual.

7.- ASIGNACION MAXIMA

En algunos problemas de asignación, se desea encontrar la asignación que sea un máximo de la función objetivo. En este caso se opera de la siguiente manera:

Sea $C = [C_{ij}]$ la matriz de pesos del problema de maximización, para aplicar el método Húngaro a ésta matriz se procede como sigue:

1).- A todas las entradas de la matriz C se le cambia de signo, obteniéndose la matriz C', es decir:

$$C'_{ij} = -C_{ij} \qquad \text{para } i,j = 1,2,..,n$$

2).- A la matriz C' se le aplican los tres pasos del método Húngaro, obteniéndose la asignación óptima de C'.

3).- Utilizando el hecho de que el máximo(f) = - mínimo(-f), se ontiene la asignación máxima de C.

Ejemplo: Supóngase que se quiere maximizar el beneficio de la siguiente matriz:

	R_1	R_2	R_3
A_1	10	8	7
A_2	8	4	3
A_3	9	6	4

(3.9)

Paso 1: Se cambia de signo a todas las entradas de 3.9 se obtiene

	R_1	R_2	R_3
A_1	-10	-8	-7
A_2	-8	-4	-3
A_3	-9	-6	-4

(3.10)

Paso 2: Aplicando el paso 1 del método húngaro, se tiene:

	R_1	R_2	R_3
A_1	0	0	0
A_2	0	2	2
A_3	0	1	2

(3.11)

Observe que 3.11 es igual a 3.13, donde la asignación óptima que se obtendra en el ejemplo al final de este capítulo es [3,1,2] y cuyo valor es 21.

8.- CONVERGENCIA DEL METODO HUNGARO

En la práctica por lo regular no se dan costos o beneficios irracionales, pero si pueden existir costos racionales, éstos costos fácilmente se pueden convertir en enteros basta con multiplicarlos por el mínimo común múltiplo (positivo) de sus denominadores, obteniéndose así una matriz de costos equivalente. A continuación se demuestra la convergencia de este algoritmo con numeros enteros.

Teorema: Si C es la matriz de pesos enteros de un problema de asignación, entonces el método Húngaro converge.

Demostración: Sea $\hat{C}=[\hat{C}_{ij}]$ la matriz actual equivalente a C en alguna iteración del método Húngaro, y sea $\overline{C} = [\overline{C}_{ij}]$ la nueva matriz equivalente que se obtiene del paso 3 (se muestra en (4)). Como se hizo saber en la sección 5.1 y 5.3, $\hat{C}$ y $\overline{C}$ tienen todas sus entradas no negativas.

Como $\hat{C}_{ij} = C_{ij} - \hat{u}_i - \hat{v}_j$ y además $\hat{C}_{ij} \geq 0$ haciendo $Z_{ij} = \hat{u}_i + \hat{v}_j$ se tiene que $\hat{C}_{ij} = C_{ij} - Z_{ij}$ es el negativo de los costos reducidos del problema primal (es decir, los costos reducidos son no positivos). La convergencia se demostrará en base a que la suma del negativo de los costos reducidos en cada iteración va disminuyendo en una cantidad entera.

Unicamente hay que establecer que:

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\hat{C}_{ij} - \sum_{i=1}^{n}\sum_{j=1}^{n}\overline{C}_{ij} = \text{número entero positivo, en efecto:}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\hat{C}_{ij} - \sum_{i=1}^{n}\sum_{j=1}^{n}\overline{C}_{ij} = \sum_{i=1}^{n}\sum_{j=1}^{n}(\hat{C}_{ij} - \overline{C}_{ij})$$

$$= \sum_{i\in F}\sum_{j\in H^c}(\hat{c}_{ij}-\overline{c}_{ij}) + \sum_{i\in F^c}\sum_{j\in H}(\hat{c}_{ij}-\overline{c}_{ij}) +$$

$$\sum_{i\in F}\sum_{j\in H}(\hat{c}_{ij}-\overline{c}_{ij}) + \sum_{i\in F^c}\sum_{j\in H^c}(\hat{c}_{ij}-\overline{c}_{ij})$$

$$= 0+0 + \sum_{i\in F}\sum_{j\in H}(\hat{c}_{ij}-\overline{c}_{ij}) + \sum_{i\in F^c}\sum_{j\in H^c}(\hat{c}_{ij}-\overline{c}_{ij})$$

$$= \sum_{i\in F}\sum_{j\in H}(\hat{c}_{ij}-(\hat{c}_{ij}+C_0)) + \sum_{i\in F^c}\sum_{j\in H^c}(\hat{c}_{ij}-(\hat{c}_{ij}-C_0))$$

$$= \sum_{i\in F^c}\sum_{j\in H^c}C_0 - \sum_{i\in F}\sum_{j\in H}C_0$$

$= (n-p)(n-q)C_0 - pqC_0$

$= (n^2 - np - nq + pq)C_0 - pqC_0$

$= (n^2 - np - nq)C_0 = n(n - (p+q))C_0$

pero hay p renglones y q columnas así que K=p+q se sigue que:

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \hat{C}_{ij} - \sum_{i=1}^{n} \sum_{j=1}^{n} \overline{C}_{ij} = n(n - K)C_0$$

como $n > K$, se sigue que $n(n - k)c_0$ es un entero positivo . ■

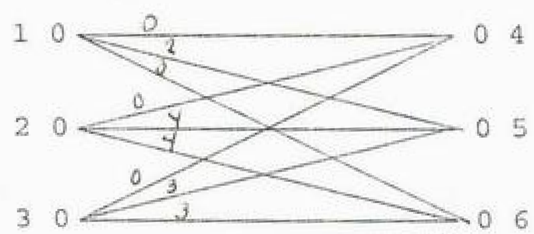
9.- IMPLEMENTACION COMPUTACIONAL

9.1.- Representación de una gráfica en la computadora

Uno de los aspectos más importantes en la manipulación de una gráfica es su representación en la computadora. Para el caso del problema de asignación se representará en la computadora como una **matriz de peso**, la cual se descibe a continuación:

La matriz de peso o la matriz vértice-vértice de una gráfica bipartita $G(V_1,V_2,A)$, es la matriz $C = [C_{ij}]_{nxn}$, donde c_{ij} es el peso de la arista $(i,j) \in A$. Si no existe arista del vértice $i \in V_1$ al vértice $j \in V_2$ en G entonces el elemento correspondiente c_{ij} se hace ∞ (en la práctica un valor muy grande), puede observarse que los vértices de V_1 corresponden a los renglones mientras que los de V_2 a las columnas de la matriz.

Por ejemplo dada la siguiente gráfica



su correspondiente matriz de peso es:

0	2	3
0	4	4
0	3	3

9.2.- Tipo de problema (minimizar o maximizar)

El procedimiento tipo de problema lo que hace es análizar que tipo de problema se va a resolver ya sea un problema de minimización o maximización, si es minimizar la matriz de peso original queda igual, si es maximizar todas las entradas de la matriz de peso original se convierten en negativos. A continuación se da el pseudocodigo de éste procedimiento.

```
procedimiento tipo_ problema;
comienza
  si tipo=1 entonces   (* el problema es de maximización *)
    comienza
      para i=1 hasta n has
         comienza
           para j=1 hasta n has
              C[i,j]= -C[i,j];
         fin;
    fin;
    si no                (* el problema es de minimización *)
      comienza
        para i=1 hasta n has
           comienza
             para j=1 hasta n has
                C[i,j]= C[i,j];
           fin;
      fin;
fin;
```

9.3.- Reduccion de la matriz de peso

El procedimiento reducematriz inicia seleccionando el mínimo de cada renglón y restándoselo a cada entrada del renglón, de manera análoga para las columnas. Este procedimiento llama a una función que calcula el mínimo entre dos valores.

El pseudocodigo de la función y procedimiento se enlista a continuación.

```
función min (i,j: enteros): entero    (* la función min devuelve un
comienza                               valor entero *)
     si i<= j entonces min=i si no min=j
fin;

procedimiento reducematriz;
comienza
  para i=1 hasta n has  (* n tamaño de la matriz *)
    comienza
      temp=m;  (* m es un valor muy grande *)
      para j=1 hasta n has temp=min(temp,C[i,j]); (* calcula el mínimo
      si temp > 0 entonces                           por renglón *)
      comienza
        para j=1 hasta n has
          si C[i,j] < m entonces
          C[i,j]=C[i,j]-temp; (* a cada renglón le resta su respestivo
      fin;                       mínimo *)
    fin;
  para j=1 hasta n has
    comienza
      temp=m;
      para i=1 hasta n has temp=min(temp,C[i,j]); (* calcula el mínimo
      si temp >0 entonces                            por columna *)
      comienza
        para i=1 hasta n has
```

```
        si C[i,j] < m entonces
        C[i,j]=C[i,j]-temp; (* a cada columna le resta su respectivo
    fin;                          mínimo *)
  fin;
fin;
```

9.4.- Gráfica bipartita asociada

La gráfica bipartita asociada se forma con los ceros de la matriz reducida y utiliza dos arreglos los cuales son: Iarc en este se almacenan los renglones y en Jarc almacena las columnas con entrada cero respectivamente, además utiliza la lista ligada hacia adelante vista en el capítulo 2. A continuación se da el pseudocodigo del procedimiento para encontrar la gráfica bipartita asociada.

```
procedure bipartita_asociada;
comienza
  t8=0;
  para i=1 hasta n has (* n dimensión de la matriz de costo *)
  comienza
    para j=1 hasta n has
      si C[i,j] = 0 entonces
      comienza
        t8=t8+1;
        Iarc[t8]=i;
        jarc[t8]=j+n;
      fin;
  fin;
  nodos=n+n;
  nodo1=nodos+1;
  arcos=t8;
  para i=1 hasta arcos has grado[i]=0;
  para i=1 hasta arcos has
  comienza
    grado[Iarc[i]]=grado[Iarc[i]]+1;
    grado[Jarc[i]]=grado[Jarc[i]]+1;
  fin;
  apuntador[1]=1;
  para i=2 hasta nodo1 has apuntador[i]=apuntador[i-1]+grado[i-1];
  para i=1 hasta nodo1 has apuntador1[i]=apuntador[i];
  para i=1 hasta arcos has
    comienza
      endv[apuntador1[Iarc[i]]=Jarc[i];
      endv[apuntador1[Jarc[i]]=Iarc[i];
      apuntador1[Iarc[i]]=apuntador1[Iarc[i]]+1;
      apuntador1[Jarc[i]]=apuntador1[Jarc[i]]+1;
    fin;
fin;
```

9.5.- Cubierta mínima

Tiene el mismo desarrollo que el algoritmo de acoplamiento máximo visto en el capítulo 2, sólo que al procedimiento cubierta mínima se le añaden dos arreglos el *coverV1* y *coverV2* que contiene los vértices cubiertos de V_1 y V_2 respectivamente. Este algoritmo proporciona los renglones y columnas que se van a cubrir en la matiz reducida.

9.6.- Mínimo no cubierto

Este procedimiento encuentra el mínimo de los renglones y columnas no cubiertas, además utiliza la función min vista en el capítulo anterior. A continuación se da el pseudocodigo del procedimiento para encontrar el mínimo.

```
procedimiento min_no_cubiertos;
comienza
  para i=1 hasta n has
    comienza
      si i ≠ coverV1[i] entonces
       comienza
         para j=1 hasta n has
          comienza
            si coverV2[j] = 0 entonces
            temp2=min(temp,C[i.j]);(* calcula el mínimo de los ren
           fin;                     glones y columnas no cubiertas*)
       fin;
    fin;
fin;
```

9.7.- Reducir renglón no cubierto

Aquí lo que se hace es restar el mínimo (de los renglones y columnas no cubiertas) a los renglones no cubiertos. El pseudocodigo de éste procedimiento se da a continuación.

```
procedimiento reducir__reng__ no_cubierto;
comienza
  para i=1 hasta n has
    comienza
      si coverV1[i] = 0 entonces
       comienza                   (* el renglón i no esta cubierto *)
         para j=1 hasta n has
           comienza
             C[i,j]=C[i,j]-temp2;  (* resta el mínimo a todos los
           fin;                       renglones no cubiertos *)
       fin;
    fin;
fin;
```

9.8.- Reducir columna cubierta

Lo que hace es sumar el mínimo (de los renglones y columnas no cubiertas) a las columnas cubiertas. El pseudocodigo para el procedimiento reducir columna cubierta es:

```
procedimiento reducir__col__ cubierta;
comienza
  para j=1 hasta n has
    comienza
      si coverV2[j] ≠ 0 entonces
       comienza                        (* la columna j esta cubierta *)
         para i=1 hasta n has
           comienza
             C[i,j]=C[i,j]+temp2;  (* suma el mínimo a cada renglón
           fin;                          cubierto *)
       fin;
    fin;
fin;
```

9.9.- Método Húngaro

A continuación se da el pseudocodigo del método Húngaro utilizando los procedimientos descritos anteriormente:

```
comienza
     leermatriz;
     tipo_ problema;
     reducematriz;            * Paso 1 *
     bipartita_asociada;      * Paso 2 *
     cubierta_minima;
     mientras K < N has       * Paso 3 *
          comienza
               min_no_cubiertos;
               reducir_reng_no_cubierto;
               reducir_col___cubierta;
               bipartita_asociada;       * repite el paso 2 *
               cubierta;
          end;
end;
```

9.10.- Ejemplo

A la siguiente matriz de peso se le aplicará el método Húngaro, utilizando los algoritmos desarrollados anteriormente y además supóngase que se quiere obtener la asignación de valor mínimo de la siguiente matriz de asignación:

	R_1	R_2	R_3
A_1	0	2	3
A_2	0	4	5
A_3	0	3	5

(3.12)

Formato del archivo de entrada:

1) La primera línea debe contener algún título de preferente relacionado con el problema.

2) La segunda línea esta compuesta por la dimensión de la matriz de peso y 0 (ó 1) si se desea minimizar (o maximizar).

3) Las otras líneas corresponden a la matriz de peso, éstos se introducen igual que un arreglo matricial (con un espacio entre cada entrada).

4) El archivo de entrada contiene el nombre del problema con cualquier extensión pero de preferencia que sea .dat (en este caso asignación.dat).

Formato del archivo de salida :

1) El archivo de salida puede tomar cualquier nombre pero de preferencia que sea el nombre del archivo de entrada con cualquier extensión pero de preferencia que sea .sal (en este caso asignación.sal), el archivo de salida contiene la siguiente información:

a) un letrero que dice: "problema de asignación".
b) título del problema.
c) asignación óptima.
d) valor de la asignación óptima.

Por lo tanto el archivo de entrada de 3.12 es:

```
problema de asignación
3 0
0 2 3
0 4 5
0 3 5
```

Paso 1 Húngaro: Primero hay que buscar los mínimos por cada renglón los cuales son cero por tanto la matriz de asignación queda igual, así que los mínimos son {0,2,3} que correponden a las columnas R_1, R_2 y R_3 respectivamente. Por el algoritmo 9.3 se obtiene la siguiente matriz de pesos reducidos:

	R_1	R_2	R_3
A_1	0	0	0
A_2	0	2	2
A_3	0	1	2

(3.13)

Paso 2 del Húngaro: La gráfica bipartita asociada a 3.13 (la matriz que resultó al aplicarle el algoritmo 9.3) tiene por cubierta mínima $K=\{A_1,R_1\}$ (coverV1=[A_1], coverV2=[R_1]) obtenida en el capítulo 2. Como $|K| < |n|$ vaya al paso 3.

Paso 3 del Húngaro: Mientras $|K| < |n|$ hacer:

Como $K=\{A_1,R_1\}$ se va a cubrir el renglón A_1 y la columna R_1 en la siguiente matriz

	R_1	R_2	R_3
A_1	0	0	0
A_2	0	2	2
A_3	0	1	2

(3.14)

por el algoritmo 9.6 en 3.14 el mínimo es 1, y por los procedimientos 9.7 y 9.8 se obtiene la siguiente matriz:

	R_1	R_2	R_3
A_1	1	0	0
A_2	0	1	1
A_3	0	0	1

(3.15)

Paso 2 del Húngaro: La gráfica bipartita asociada (obtenida por el algoritmo 9.4) y el acoplamiento inicial visto en el capítulo 2 (se muestra con líneas gruesas) es:

Además la trayectoria M-aumentante T: $A_3 \longrightarrow R_2 \longrightarrow A_1 \longrightarrow R_3$ se muestra con *. Obteniéndose el siguiente acoplamiento:

Observe que este acoplamiento es máximo y satura a todos los vertices, por lo tanto se ha encontrado una asignación completa (acoplamiento perfecto) la cual es [R_3,R_1,R_2] con valor 6.

CONCLUSION

En este trabajo se han analizado los fundamentos teóricos de los métodos de solución del problema de acoplamiento de cardinalidad máxima y del problema de acoplamento de peso máximo en gráficas bipartitas (problema de asignación), así como el desarrollo de algunas de sus múltiples aplicaciones e implementación en pascal de dichos métodos.

La utilidad de la implementación de estos métodos es que se puede obtener la solución de problemas reales relativamente grandes en un tiempo computacional razonable. Además estas implementaciones pueden utilizarse como procedimientos para resolver los problemas de acoplamiento en gráficas no bipartitas y el problema del agente viajero.

Para entender como trabajan ambos métodos no se requiere de un conocimiento matemático muy elevado, sólo se necesitan ciertos resultados de teoría de gráficas y de dualidad lineal, en ambos casos muy elementales. Además, por la estructura que el problema presenta en gráficas bipartitas, los algoritmos implementados son muy eficientes lo cual permite resolver instancias suficientemente grandes en un tiempo razonable.

Los problemas de acoplamiento pueden extenderse en las siguientes direcciones:

i) Para gráficas no bipartitas. El problema se complica ya que pueden existir ciclos de cardinalidad impar, sin embargo, los algoritmos siguen siendo polinomiales (para mayor información ver referencia [10] y [13]).

ii) Otra de las variantes es el acoplamiento Max-min (o cuello de botella) el cual consiste en encontrar un acoplamiento de cardinalidad máxima cuya arista de menor peso en el acoplamiento sea lo mayor posible.

iii) Un problema más general es el de asignación cudrática denominado de esta manera ya que su función objetivo es cuadrática (puede consultar referencia [11]). Incluso el problema de asignación cuadrática es una generalización del problema del agente viajero (ver referencia [8]) y este a su vez puede utilizar como relajación al problema de asignación lineal.

A P E N D I C E A

PROBLEMA DUAL

Cada problema lineal tiene asociado otro problema lineal llamado el problema dual (D). El problema dual posee muchas propiedades importantes relativas al problema primal original (P).

Supóngase que el problema primal está dado en la forma

P: minimizar $c^t x$
sujeto a: $Ax=b$
$x \geq 0$.

Entonces el problema dual está definido por:

D: maximizar $w^t b$
sujeto a: $w^t A < c^t$
w no restringida.

Nótese que existe exactamente una variable dual por cada restricción primal, y exactamente una restricción dual por cada variable dual.

Teorema (teorema fundamental de dualidad): Con respecto a los problemas primal y dual, exactamente una de las siguientes proposiciones es cierta.

1) Ambos problemas tienen soluciones óptimas x^* y w^*, con $c^t x = (w^*)^t b$.

2) Uno de los problemas tiene valor objetivo no acotado, en cuyo caso el otro problema debe ser no factible.

3) Ambos problemas son no factibles.

Holgura complementaria: Sean x^* y w^* cualquier par de soluciones óptimas de los problemas primal y dual, respectivamente. Entonces:

$$c^t x^* \geq (w^*)^t A x^* \geq (w^*)^t b$$

pero $c^t x^* = (w^*)^t b$. Luego $c^t x^* = (w^*)^t A x^* = (w^*)^t b$

Esto da $(w^*)^t (A x^* - b) = 0$ y $(c - w^*A)^t x^* = 0$. Puesto que $w^* \geq 0$ y $A x^* - b \geq 0$, entonces $(w^*)^t (A x^* - b) = 0$ implica que $(w_i^*)^t (a_i x^* - b_i) = 0$ para $i=1,2,..,m$. De igual manera $(c - w^*A)^t x^* = 0$ implica que

$(c_j - w^* a_j)^t x_j^* = 0$ para $j=1,2,..,n$.

Teorema (Teorema débil de holgura complementaria): Si x^* y w^* son puntos óptimos cualesquiera de los problemas primal y dual, entonces:

$$(c_j - w^* a_j)^t x_j^* = 0 \text{ para } j=1,2,..,n. \qquad y$$

$$(w_i^*)^t (a_i x^* - b_i) = 0 \quad \text{para } i=1,2,..,m.$$

Este es un teorema muy importante que los problemas primal y dual. Obviamente indica que al menos uno de los dos factores en cada una de las expresiones debe ser cero. En particular,

$$x_j^* > 0 \Rightarrow (w^*)^t a_j = c_j$$

$$(w^*)^t a_j < c_j \Rightarrow x_j^* = 0$$

$$w_i^* > 0 \Rightarrow (a_i)^t x^* = b_i$$

$$(a_i)^t x^* > b_i \Rightarrow w_i^* = 0.$$

El teorema débil de holgura complementaria también se puede enunciar como sigue: En caso de optimalidad, " si una variable en uno de los problemas es positiva, entonces la restricción correspondiente en el otro problema es sin holgura", y " si una restricción en uno de los problemas es con holgura, entonces la variable correspondiente en el otro problema debe ser cero".

Supóngase que $x_{n+j} = (a_i)^t x - b_i \geq 0$, $i=1,2,..,m$ son las m variables de holgura en el problema primal y sean $w_{m+j} = c_j - (w)^t a_j \geq 0$, $j=1,2,..,n$ las n variables de holgura en el problema dual. Entonces se pueden escribir las condiciones de holgura complementaria como sigue:

$$(x_j^*)^t w_{m+j}^* = 0, \quad j=1,2,..,n$$

$$(w_i^*)^t x_{n+i}^* = 0, \quad i=1,2,..,m$$

Esto relaciona las variables en uno de los problemas con las variables de holgura en el otro problema.

Debe observarse que si x^* y w^* son factibles para sus problemas respectivos y satisfacen las condiciones de holgura complementaria, entonces son óptimos.

Véase estas demostraciones en [6] capítulo 6, pags. 233-243 y [7] capítulo 5 pags. 54-65.

A P E N D I C E   B

SOLUCION A LOS PROBLEMAS DE APLICACION DEL CAPITULO I

B.1.- Problema de Traducción de Idiomas

Archivo de entrada

```
Traducción de Idiomas
17 22 7
1  8
1  9
1  17
2  10
2  11
2  12
3  8
3  14
3  15
4  15
5  8
5  11
5  14
5  16
6  9
6  12
6  13
7  8
7  10
7  11
7  16
7  17
```

Archivo de salida

```
PROBLEMA DE ACOPLAMIENTO MAXIMO
Traducción de Idiomas
Número de vértices expuestos = 0
 Vértice     Vértice Acoplado
   1              8
   2             10
   3             14
   4             15
   5             11
   6              9
   7             16
Cubierta Mínima:
Vértices cubiertos de V1
 1  2  3  4  5  6  7
```

El máximo número de trabajos que puede realizar el Departamento de Idiomas son 7, los cuales se muestran a continuación con su respectivo profesor: $P_1 \longrightarrow T_1$, $P_2 \longrightarrow T_3$, $P_3 \longrightarrow T_7$, $P_4 \longrightarrow T_8$, $P_5 \longrightarrow T_4$, $P_6 \longrightarrow T_2$, $P_7 \longrightarrow T_9$.

B.2.- Problema de Despacho de Asesores

Archivo de entrada

```
Despacho de Asesores (SRD)
16 15 7
1 9
1 10
2 9
2 12
3 8
3 13
4 13
5 10
5 11
5 14
6 8
6 13
7 12
7 15
7 16
```

Archivo de salida

```
PROBLEMA DE ACOPLAMIENTO MAXIMO
Despacho de asesores (SRD)
Número de vértices expuestos = 0
Vértice          Vértice Acoplado
  1                 9
  2                12
  3                 8
  4                13
  5                10
  6                 0
  7                15
Cubierta Mínima
Vértices cubiertos de V1
 1  2  3  4  5  6  7
```

Este problema no tiene solución porque la cardinalidad del acoplamiento máximo es 7, es decir, alguno de los asesores deberá representar a más de una empresa en la reunión de evaluación.

B.3.- Problema del Periódico

Archivo de entrada

```
Reporteros
12 12 6
1 9
1 10
1 11
1 12
2 10
2 11
2 12
3 10
3 11
3 12
4 11
5 12
```

Archivo de salida

```
PROBLEMA DE ACOPLAMIENTO MAXIMO
Reporteros (D. Cadenas)
Número de vértices expuestos = 0
Vértice         Vértice Acoplado
  1                 9
  2                10
  3                11
  4                 0
  5                12
  6                 0
Cubierta Mínima
Vértices cubiertos de V1
 1  2  3  4  5  6
```

El mínimo número de reporteros que pueden cubrir el total de las conferencias son 2. Donde el reportero 1 cubre las conferencias 1,3,5, y 6, el reportero 2 cubre las conferencias 2 y 4.

B.4.- Problema del corredor de bienes raíces

Archivo de entrada

```
corredor bienes raíces
5 1
16 15 25 19 20
14 17 24 15 25
17 17 20 2  18
22 3  18 20 21
0  0  0  0  0
```

Archivo de salida

```
Problema de Asignación
Corredor de bienes raíces
Asignación óptima
  El renglón 1 se asigna a la columna 3
  El renglón 2 se asigna a la columna 5
  El renglón 3 se asigna a la columna 2
  El renglón 4 se asigna a la columna 1
  El renglón 5 se asigna a la columna 4
Valor óptimo = 89.00
```

Para que el ingreso total sea máximo es necesario que el comprador C_1 se decida por el lote L_3, C_2 por L_5, C_3 por L_2, C_4 por L_1, C_5 por L_4, cuyo ingreso total es 89.

B.5.- Problema de Calendarización de Vuelos

Archivo de entrada

```
Calendarizacion de vuelos
5 0
17.5 15    9    60   12
16   16.5 10.5 60   10.5
12   15.5 14.5 11   60
60   8    14   17.5 13
13   9.5  8.5  12   18.5
```

Archivo de salida

```
Problema de Asignación
Calendarización de vuelos
Asignación óptima
 El renglón 1 se asigna a la columna 3
 El renglón 2 se asigna a la columna 5
 El renglón 3 se asigna a la columna 1
 El renglón 4 se asigna a la columna 2
 El renglón 5 se asigna a la columna 4
Valor óptimo = 51.50
```

A continuación se da la asignación de tripulaciones a vuelos pero además en que lugar deben de vivir estas tripulaciones.

vuelo	*proviene de*	*Descripción de la asignación óptima*	
A-3	Matriz II	Sale de R-J a 15 h. Llega México 21 h	(dia 1)
		Sale de México a 6 h. Llega R-J 12 h	(dia 2)
B-5	Matriz I	Sale de México a 7:30 h. Llega R-J 13:30 h	(dia 1)
		Sale de R-J a 0:00 h. Llega México 6:00 h	(dia 2)
C-1	Matriz I	Sale de México a 11:30 h.Llega R-J 17:30 h	(dia 1)
		Sale de R-J a 5:30 h. Llega México 11:30 h	(dia 2)
D-2	Matriz I	Sale de México a 19:00 h. Llega R-J 1:00 h	(dia 1)
		Sale de R-J a 9:00 h. Llega México 15:00 h	(dia 2)
E-4	Matriz I	Sale de México a 0:30 h. Llega R-J 6:30 h	(dia 1)
		Sale de R-J a 18:30 h. Llega México 0:30 h	(dia 2)

B.6.- Problema del Enrole de locomotoras

Archivo de entrada

Enrole de locomotoras
18 0

0.09	2.95	30.7	7.79	12.54	15.12	18.95	21.5	31.04	6.87	10.5	15.5	18.3	28	7.2	12	15.3	18.8
21.3	0.25	4	5.09	9.84	12.42	16.25	18.8	28.34	4.17	7.84	12.8	15.6	25.3	4.50	9.34	12.6	16.1
17.6	20.5	0.25	1.34	6.09	8.67	12.50	15.05	24.59	24.42	4.09	9.09	11.9	21.5	0.75	5.59	8.92	12.4
15.2	18.08	21.83	22.92	3.67	6.25	10.08	12.63	22.17	22	25.6	6.67	9.50	19.1	22.3	27.1	30.5	10
33.	12	15.75	16.84	21.59	24.17	28	30.55	16.09	15.92	19.5	24.5	27.4	13	16.2	21	24.4	3.92
9.32	12.18	15.93	17.02	21.77	0.35	4.18	6.73	16.27	16.1	19.7	0.77	3.6	13.2	16.4	21.2	24.6	28.1
5.39	8.25	12	13.09	17.84	20.42	0.25	2.8	12.34	12.17	15.8	20.8	23.6	9.34	12.5	17.3	20.6	27.1
2.67	5.53	9.28	10.37	15.12	17.7	21.53	0.08	33.62	9.45	13.1	18.1	20.9	30.6	9.78	14.6	17.9	21.4
9.89	12.75	16.50	17.59	22.34	0.92	4.75	7.3	16.84	16.67	20.3	1.34	4.17	13.8	17	21.8	25.1	28.6
17.4	20.28	24.03	25.12	29.87	8.45	12.28	14.83	24.37	0.2	3.87	8.87	11.7	21.3	24.5	5.37	8.7	36.2
14.1	17	20.75	21.84	26.59	5.17	9	11.55	21.09	20.92	0.59	5.59	8.42	18	21.2	2.09	5.42	32.9
9.12	11.98	15.73	16.82	21.57	0.15	3.98	6.53	16.07	15.9	19.5	0.57	3.4	13	16.2	21	24.4	27.9
30	8.91	12.66	13.75	18.5	21.08	24.91	27.46	13	12.83	16.5	21.5	24.3	10	13.1	18	21.3	24.8
20.6	23.5	27.25	28.34	33.09	11.67	15.5	18.05	3.59	27.42	31	12	14.9	24.5	27.7	32.5	35.9	15.4
17.5	20.4	0.15	1.24	5.99	8.57	12.4	14.95	24.49	24.32	3.99	8.99	11.8	21.4	0.65	5.49	8.82	12.3
12.5	15.41	19.16	20.25	25	3.58	7.41	9.96	19.5	19.33	23	4	6.83	16.5	19.6	0.5	3.83	31.3
8.97	11.83	15.58	16.67	21.42	24	3.83	6.38	15.92	15.78	19.4	24.4	27.2	12.9	16	20.9	0.25	27.7
19.5	22.41	2.16	3.25	8	10.58	14.41	16.96	26.5	26.33	6	11	13.8	23.5	2.66	7.5	10.8	14.3

Archivo de salida

```
Problema de asignación
Enrole de locomotora
Asignación óptima
   El renglón  1   se  asigna a la columna  1
   El renglón  2   se  asigna a la columna  2
   El renglón  3   se  asigna a la columna  3
   El renglón  4   se  asigna a la columna  5
   El renglón  5   se  asigna a la columna  18
   El renglón  6   se  asigna a la columna  13
   El renglón  7   se  asigna a la columna  7
   El renglón  8   se  asigna a la columna  8
   El renglón  9   se  asigna a la columna  6
   El renglón  10  se  asigna a la columna  10
   El renglón  11  se  asigna a la columna  11
   El renglón  12  se  asigna a la columna  12
   El renglón  13  se  asigna a la columna  14
   El renglón  14  se  asigna a la columna  9
   El renglón  15  se  asigna a la columna  15
   El renglón  16  se  asigna a la columna  16
   El renglón  17  se  asigna a la columna  17
   El renglón  18  se  asigna a la columna  4
 Valor óptimo = 32.63
```

Descripción de la asignación óptima

(1,1)——> bala que llega a Navojoa proveniente del Sur $\xrightarrow{\text{se asigna}}$ bala que sale de Navojoa rumbo a Empalme

(2,2)——> bala que llega a Empalme proveniente de Navojoa $\xrightarrow{\text{se asigna}}$ bala que sale de Empalme rumbo a B. Hill

(3,3)——> bala que llega a B. Hill proveniente de Empalme $\xrightarrow{\text{se asigna}}$ bala que sale de B. Hill rumbo a Nogales

(4,5)——> bala que llega a Nogales proveniente de B. Hill $\xrightarrow{\text{se asigna}}$ bala que sale de Nogales rumbo a B. Hill

(5,18)——> bala que llega a Mexicali proveniente de B. Hill $\xrightarrow{\text{se asigna}}$ burro que sale de Mexicali rumbo a B. Hill

(6,13)——> bala que llega a B. Hill proveniente de Nogales $\xrightarrow{\text{se asigna}}$ burro que sale de B.Hill rumbo a Mexicali

(7,7)——> bala que llega a Empalme proveniente de B. Hill $\xrightarrow{\text{se asigna}}$ bala que sale de Empalme rumbo a Navojoa

(8,8)——>bala que llega a Navojoa proveniente de Empalme $\xrightarrow{\text{se asigna}}$ bala que sale de Navojoa rumbo al Sur

(9,6)——>bala que llega a B. Hill proveniente de Mexicali $\xrightarrow{\text{se asigna}}$ bala que sale de B. Hill rumbo a Empalme

(10,10)——>burro que llega a Navojoa proveniente del Sur $\xrightarrow{\text{se asigna}}$ burro que sale de Navojoa rumbo a Empalme

(11,11)——>bala que llega a Empalme proveniente de Navojoa $\xrightarrow{\text{se asigna}}$ burro que sale de Empalme rumbo a B. Hill

(12,12)——>burro que llega a B. Hill proveniente de Empalme $\xrightarrow{\text{se asigna}}$ burro que sale de B.Hill rumbo a Nogales

(13,14)——>burro que llega a Nogales proveniente de B. Hill $\xrightarrow{\text{se asigna}}$ burro que sale de Nogales rumbo a B. Hill

(14,9)——>burro que llega a Mexicali proveniente de B. Hill $\xrightarrow{\text{se asigna}}$ bala que sale de Mexicali rumbo a B. Hill

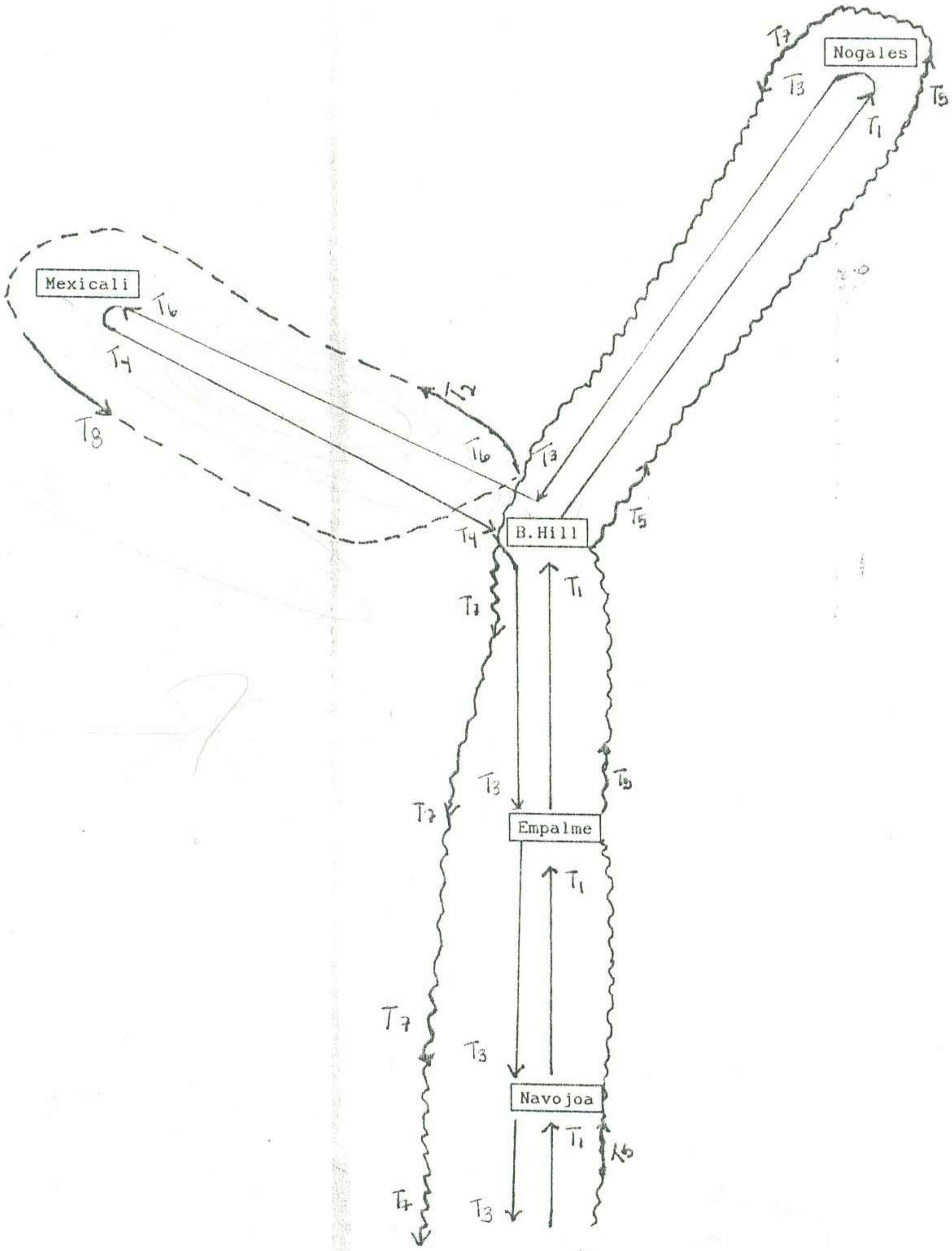
(15,5)——>burro que llega a B. Hill proveniente de Nogales $\xrightarrow{\text{se asigna}}$ burro que sale de B.Hill rumbo a Empalme

(16,16)——>burro que llega a Empalme proveniente de B. Hill $\xrightarrow{\text{se asigna}}$ burro que sale de Empalme rumbo a Navojoa

(17,17)——>burro que llega a Navojoa proveniente de Empalme $\xrightarrow{\text{se asigna}}$ burro que sale de Navojoa rumbo al Sur

(18,4)——>burro que llega a B. Hill proveniente de Mexicali $\xrightarrow{\text{se asigna}}$ bala que sale de B.Hill rumbo a Mexicali

El número total de locomotoras que se necesitan para recorrer todas las estaciones son 3.



——— Indica que una locomotora recorre todas las estaciones.

- - - - Indica que una locomotora recorre las estaciones de B.H y Mexicali

~~~~ Indica que una locomotora recorre las estaciones de Navojoa a Nogales y viceversa.
~~~~

BIBLIOGRAFIA

[1] Bazaraa S. Mokhtar, Jarvis J. John : Programación lineal y flujo en redes. Editorial Limusa, 1986.

[2] Bolobás Béla: Graph theory, An Introductory Course. Springer-Verlag, 1979.

[3] Chvátal Vasek: Linear programming. W. H. Freeman, 1983.

[4] Gale David, "The optimal assignment problem"; en Umap, Vol. 15, 1985.

[5] Hernández N. Miguel, Pulido E. Lázaro, "El enrole de locomotoras para un itinerario de trenes"; en Investigación operacional, no.23, 1978, pag. 31-39.

[6] Halmos R. Paul: Teoría intuitiva de los conjuntos. Editorial Continental, 1984.

[7] Joyanes Aguilar Luis: Turbo pascal 6.0. Editorial McGraw-Hill, 1993.

[8] Lawler L. E., Lenstra K. J., Rinnooy Kan G. A. H., Shmoys B. D.: The Traveling Salesman Problem. John Wile and Sons, 1985.

[9] Nemhauser L. George, Wolsey A. Laurence: Integer and Combinatorial Optimization. John Wile and Sons, 1988.

[10] Papadimitriou H. Christos, Steiglitz Kenneth: Combinatorial Optimization, Algorithms and complexity. Prentice-Hall, 1990.

[11] Prawda Juan: Métodos y modelos de investigación de operaciones, Vol. 1. Editorial Limusa, 1976.

[12] Rockafellar R. T.: Network Flows and Monotropic Optimization. John Wiley and Sons, 1984.

[13] Syslo M. Maciej, Narsingh Deo, Janusz S. Kowalik: Discrete optimization algorithms. Prentice-Hall, 1983.

[14] Wilson J. Robin : Introducción a la teoría de grafos. Editorial Cast, 1983.