



**UNIVERSIDAD NACIONAL  
AUTONOMA DE MEXICO**

---

---

**Facultad de Ciencias**

**PROBLEMAS BASICOS DE REDES DE FLUJO:  
PROPIEDADES Y ALGORITMOS**

**T E S I S**  
Que para obtener el Título de  
**MATEMATICA**

**P r e s e n t a**

**MARIA DEL CARMEN HERNANDEZ AYUSO**



**México, D. F.**

**1985**



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

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

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

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

## I N D I C E .

INTRODUCCION		1
<b>CAPITULO 1.</b>	<b>ARBOL DE PESO MINIMO</b>	
	1.1 Descripción del problema	5
	1.2 Propiedades de árboles	8
	1.3 Métodos de solución y justificación	17
	1.4 Análisis de sensibilidad	29
<b>CAPITULO 2.</b>	<b>RUTA MAS CORTA</b>	
	2.1 Descripción del problema	34
	2.2 Caracterización de una arborescencia	39
	2.3 Métodos de solución y justificación	43
	a. Arborescencia de rutas más cortas	44
	b. Ruta más corta entre todo par de - vértices.	77
<b>CAPITULO 3.</b>	<b>FLUJO MAXIMO</b>	
	3.1 Descripción del problema	95
	3.2 Teorema de flujo máximo-cortadura - mínima	99
	3.3 Método de solución y justificación	103
	3.4 Variantes del problema	120

<b>CAPITULO 4.</b>	<b>FLUJO A COSTO MINIMO ENTRE ORIGEN Y DESTINO</b>	
4.1	Descripción del problema	134
4.2	Conceptos básicos	135
4.3	Métodos de solución	146
	a. Método basado en eliminación de - circuitos negativos	146
	b. Método basado en rutas más cortas	156
<b>CAPITULO 5.</b>	<b>FLUJO A COSTO MINIMO CON OFERTAS EN TODOS LOS VERTICES</b>	
5.1	Descripción del problema	161
5.2	Caracterización y propiedades de bases	164
5.3	Método simplex especializado en redes	174
5.4	Método simplex especializado en redes con variables acotadas	193
<b>CONCLUSIONES:</b>		207
<b>APENDICE</b>	<b>Elementos de teoría de gráficas</b>	208
<b>BIBLIOGRAFIA</b>		216

## **INTRODUCCION**

## INTRODUCCION

La teoría de redes es una clase de modelos matemáticos que involucra la representación gráfica de ciertos problemas de optimización. Las redes tienen una aplicación extensa en diversos campos como son: planeación, administración, ingeniería, química, educación, etc.; en estos campos innumerables situaciones pueden formularse como modelos matemáticos de redes. Algunos ejemplos son: sistemas de producción-distribución, tráfico urbano, transporte colectivo, comunicación, redes eléctricas, reemplazo de equipo, inventarios, presas, flujo de dinero, tuberías, oleoductos, asignación de recursos y otros.

A causa de la vasta aplicación de este tipo de modelos y a la valiosa ayuda que proporcionan para el entendimiento de los sistemas, ha habido gran actividad en su estudio. Gracias a esto, y a la estructura especial que presentan los modelos de redes, se han desarrollado algoritmos eficientes para la solución de los problemas formulados. Incluso se ha desarrollado más de un algoritmo para resolver el mismo tipo de problema en base a las restricciones que en él se consideren. Además, los algoritmos son, en su mayoría, de relativa facilidad en su comprensión y aplicación ya que surgen de manera natural en el desarrollo de la teoría. Otra ventaja de algunos de los algoritmos es que, a través de ellos, es posible detectar cuando un problema de redes no tiene solución.

En la teoría de redes existe un conjunto bien definido de problemas básicos como: ruta más corta, flujo máximo, flujo a costo mínimo, entre otros. Se conocen con el nombre de problemas básicos ya que otros problemas pueden ser formulados como éstos. Por ejemplo, el problema de transporte a costo mínimo puede ser formulado como un problema de flujo a costo mínimo, el problema de acoplamiento de cardinalidad máxima puede formularse como uno de flujo máximo, algunos problemas de remplazo de equipo pueden formularse como uno de ruta más corta, etc.

Este trabajo tiene como objetivo analizar las propiedades de cinco problemas básicos de redes: el problema del árbol de peso mínimo, el de la ruta más corta, el del flujo máximo, el del flujo a costo mínimo entre origen y destino y el del flujo a costo mínimo con ofertas en todos los vértices. También se presentan los algoritmos más utilizados en la práctica para resolver estos problemas.

El contenido del trabajo es como sigue: en el capítulo 1 se analiza el problema del árbol de peso mínimo. Primeramente se describe el problema; después se presentan las propiedades de un árbol que serán utilizadas tanto para justificar los métodos de solución de este problema como para caracterizar ciertas soluciones del problema de la arborescencia de rutas más cortas y del problema del flujo a costo mínimo con ofertas en todos los vértices. Se enuncian y justifican dos métodos de solución para el problema y se propone una herramienta para el análisis de sensibilidad; es decir, se rescata la solución de un problema modificado.

En el capítulo 2 se analizan tres problemas de rutas más cortas: el de la ruta más corta entre dos vértices específicos, el de la arborescencia de rutas más cortas y el de la ruta más corta entre todo par de vértices. Se describen estos problemas y se establecen las condiciones necesarias para garantizar la existencia de soluciones. Se caracterizan las soluciones del problema de la arborescencia de rutas más cortas tomando como base las propiedades de los árboles. Por último, se presentan métodos de solución para estos problemas. Dos de éstos sirven para resolver los primeros dos problemas de rutas más cortas mencionados; uno se restringe a redes con costos no negativos y el otro se aplica a cualquier red.

Los algoritmos del problema de ruta más corta también se utilizan en la solución del problema del flujo máximo en donde la determinación de la existencia de rutas más cortas surge como un subproblema.

En el tercer capítulo se describe el problema del flujo máximo. Se enuncia un teorema de gran importancia para determinación de la solución del problema, así como un método de solución inspirado en tal teorema. Por último se analizan ciertas variantes del problema; es decir, se agregan ciertas restricciones y se reduce el problema para poder determinar su solución a partir del algoritmo presentado.

En el capítulo 4 se describe el problema del flujo a costo mínimo entre origen y destino. Se incluyen algunos conceptos básicos útiles en la determinación de la solución. También se enuncian dos teoremas esenciales para motivar y justificar los métodos de solución presentados.

En el quinto capítulo se analiza el problema del flujo a costo mínimo con ofertas en todos los vértices; se establecen proposiciones referentes a propiedades de ciertas soluciones del problema llamadas bases y se caracterizan éstas. En esta parte también se formula el problema como uno de programación lineal. Considerando esto, se especializa el método simplex para el caso de redes; primeramente se considera el problema con variables no acotadas y posteriormente con variables acotadas.

Finalmente se incluye un anexo en el cual se presentan los elementos de teoría de gráficas necesarios para definir los problemas tratados durante el trabajo.

## **ARBOL DE PESO MINIMO**

**CAPITULO 1.**

## 1.1 DESCRIPCION DEL PROBLEMA

Considérese el siguiente problema: En un lago hay  $n$  islas, denotadas  $x_1, x_2, \dots, x_n$ , y se desea construir puentes para comunicarlas. La construcción del puente  $(x_i, x_j)$  cuesta  $c_{ij}$  pesos. El problema consiste en determinar dónde construir los puentes de tal manera que cada par de islas queden conectadas por medio de éstos y que el costo total de construcción sea mínimo.

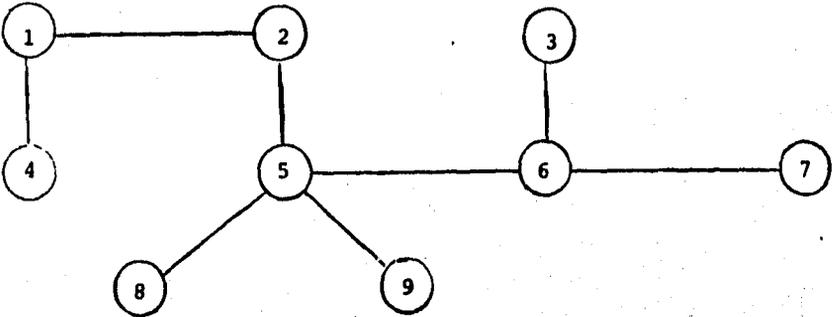
Sea  $G = [X, A]$  una gráfica, no-dirigida, donde el conjunto de vértices  $X$  representa al conjunto de islas y cada elemento  $(x_i, x_j)$  del conjunto de aristas  $A$ , representa la posible construcción de un puente entre las islas  $x_i$  y  $x_j$ . Sea  $c$  una función que asocia, a cada elemento de  $A$ , el costo de construcción del puente respectivo. Obsérvese que una solución para este problema es una gráfica parcial  $T = [X, A']$  de  $G$ . Esta gráfica parcial deberá cumplir los tres puntos siguientes:

- a.  $T$  es conexa, puesto que se desea que exista una cadena que una a todo par de vértices.
- b.  $T$  no deberá tener ciclos puesto que, de ser así, se incurriría en un costo innecesario.
- c. El costo de  $T$  deberá ser mínimo.

En base a lo anterior, se definen los siguientes conceptos.

**Definición.** Un árbol es una gráfica  $T = [X, A]$  conexa y acíclica.

La siguiente gráfica es un árbol:

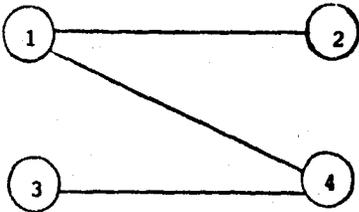
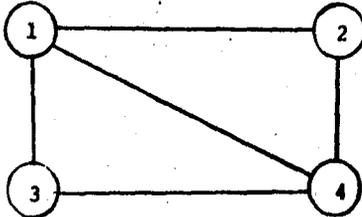


**Definición.** Sea  $G = [X, A]$  una gráfica no dirigida. Un árbol expandido de  $G$  es una gráfica parcial  $T = [X, A']$ , de  $G$ , que es un árbol.

Como ejemplo, considérense la siguiente gráfica  $G$ , un árbol expandido de ella y un árbol no expandido.

Gráfica

G



Árbol expandido de G



Árbol no expandido de G

Nótese que, entonces, la gráfica  $T$  que será solución para el problema de las islas debe ser un árbol expandido de  $G$ . Por otro lado, obsérvese que una gráfica puede tener varios árboles expandidos. Por esta razón, existen diferentes alternativas de solución para el problema de las islas; sin embargo, se tiene interés en la mejor de todas ellas, es decir, aquélla con costo total de construcción mínimo. Se definirá, entonces, el costo de un árbol.

Considérese una función  $p$  que asocia un real a cada arista de una gráfica. A este real se le llama peso de la arista. La función puede representar costo, distancia, tiempo, etc. En el caso del problema de las islas la función  $p$  fué denotada con la letra  $c$  y representa el costo de construcción de un puente.

Definición. El peso de un árbol es la suma de los pesos de las aristas que lo forman.

En base a los conceptos anteriores, es sencillo concluir que la solución óptima al problema de las islas está dada por el árbol expandido de peso mínimo asociado a la gráfica  $G$ .

Los métodos de determinación del árbol expandido de peso mínimo de una gráfica se discuten en este capítulo. Sin embargo, antes de describirlos, conviene postular algunas de las propiedades más importantes de las gráficas denominadas árboles.

## 1.2 PROPIEDADES DE ARBOLES

Existen distintas definiciones de árboles todas ellas equivalentes entre sí. Con el propósito de demostrar tales equivalencias se empezará a analizar ciertas propiedades elementales de una gráfica. La siguiente proposición muestra el efecto de añadir una arista a una gráfica con varias componentes conexas.

Proposición 1. Sea  $G = [X, A]$  una gráfica. Supóngase que se agrega la arista  $a = (x, y)$  a la gráfica  $G$ . Entonces:

i. El número de componentes conexas de  $G$  disminuye una unidad si los extremos de  $a$  pertenecen a dos componentes conexas distintas. En este caso, la arista  $a$  no pertenece a ningún ciclo de la gráfica  $G' = [X, A \cup \{a\}]$ .

ii. El número de componentes conexas de  $G$  permanece igual si los extremos de  $a$  pertenecen a la misma componente conexa.

En este caso,  $a$  pertenece a un ciclo de la gráfica  $G' = [X, A \cup \{a\}]$ .

Demostración. i. Sean  $G_1 = [X_1, A_1]$  y  $G_2 = [X_2, A_2]$  las dos componentes conexas distintas de  $G$  tales que  $x \in X_1$  y  $y \in X_2$ . Entonces la gráfica  $G_1 \cup G_2 = [X_1 \cup X_2, A_1 \cup A_2 \cup \{a\}]$  es una componente conexa de  $G'$ . En efecto; sean  $i, j$  dos vértices de  $X_1 \cup X_2$ . Como  $G_1$  y  $G_2$  son gráficas conexas, se tiene que: Si  $i$  y  $j$  pertenecen ambos a  $X_1$  (o a  $X_2$ ), entonces existe la cadena  $(i=i_1, i_2, \dots, i_k = j)$  que une  $i$  con  $j$  en  $G_1$  (o en  $G_2$ ) y por lo tanto en  $G'$ .

Si  $i \in X_1$  y  $j \in X_2$  entonces existen la cadena  $(i=i_1, i_2, \dots, i_k=x)$ , que une  $i$  con  $x$  en  $G_1$ , y la cadena  $(y=j_1, j_2, \dots, j_\ell=j)$  que une  $y$  con  $j$  en  $G_2$ . Entonces  $(i=i_1, i_2, \dots, i_k=x, y=j_1, j_2, \dots, j_\ell=j)$  es una cadena que une  $i$  con  $j$  en  $G'$ . Si  $i \in X_2$  y  $j \in X_1$  el razonamiento es análogo. Supóngase que  $a$  pertenece a un ciclo de  $G'$ . Sea  $(x=x_1, x_2, \dots, x_k=y, x)$  este ciclo; entonces  $(x=x_1, x_2, \dots, x_k=y)$  es una cadena que une  $x$  con  $y$  en  $G$ . Esto es una contradicción puesto que  $x$  y  $y$  pertenecen a componentes conexas distintas de  $G$ . Por lo tanto,  $a$  no pertenece a ningún ciclo de  $G'$ .

ii. Si  $x$  y  $y$  pertenecen a la misma componente conexa, entonces  $G$  y  $G'$  tienen las mismas componentes conexas. Por otro lado, existe la cadena  $(x=x_1, x_2, \dots, x_k=y)$  que une  $x$  con  $y$  en  $G$ . Entonces  $(x=x_1, x_2, \dots, x_k=y, x)$  es un ciclo, en  $G'$ , que contiene la arista  $a$  y la prueba termina. ■

Es también importante notar, antes de proceder a enunciar las definiciones equivalentes de árbol, que el número de aristas en un árbol es igual al número de vértices menos uno. Esto es una consecuencia inmediata de la siguiente proposición.

Proposición 2. Sea  $G = [X, A]$  una gráfica con  $n$  vértices y  $m$  aristas.

- a. Si  $G$  es conexa entonces  $m \geq n-1$
- b. Si  $G$  es acíclica entonces  $m \leq n-1$

Demostración. Supóngase que se construye la gráfica  $G=[X,A]$  agregando, una a una, sus aristas a la gráfica  $G_0=[x,\phi]$  que consiste de  $n$  componentes conexas.

a. Obsérvese que cada vez que se agrega una arista, el número de componentes conexas disminuye, a lo más, en una unidad. Por lo tanto, para obtener la gráfica conexa  $G$  a partir de la gráfica  $G_0$  con  $n$  componentes conexas se necesita agregar, al menos,  $n-1$  aristas.

b. Es inmediato que toda gráfica parcial de  $G$  es acíclica. Entonces cada vez que se agrega una arista, el número de componentes conexas disminuye exactamente una unidad. Dado que  $G$  tiene, al menos, una componente conexa se concluye que el número de aristas que hay que agregar es a lo más  $n-1$  y la prueba termina. ■

El teorema que a continuación se enuncia es muy importante puesto que postula la equivalencia de las definiciones de árbol. Es fácil demostrar la equivalencia de estas definiciones contando con las herramientas proporcionadas por las dos proposiciones anteriores. Por otro lado, las propiedades de los árboles expuestas en el teorema serán de gran utilidad para caracterizar las bases del problema de programación lineal asociado al problema de flujo a costo mínimo que se analiza en el quinto capítulo.

Teorema 3. Sea  $G = [X, A]$  una gráfica con  $n$  vértices. Supóngase que  $n \geq 2$ . Los postulados siguientes son equivalentes y caracterizan un árbol:

- a.  $G$  es conexa y acíclica
- b.  $G$  es acíclica y tiene  $n-1$  aristas
- c.  $G$  es acíclica y si se agrega una arista se forma exactamente un ciclo.
- d.  $G$  es conexa y tiene  $n-1$  aristas
- e.  $G$  es conexa pero deja de serlo si se elimina una arista
- f. Existe, en  $G$ , una única cadena entre todo par de vértices.

Demostración. (a implica b). Que  $G$  sea acíclica se sigue directamente de la hipótesis. Que  $G$  tenga  $n-1$  aristas se sigue inmediatamente de la proposición 2.

(b implica c). Supóngase que se agrega la arista  $a = (x, y)$  a  $G$ .

Sea  $G' = [X, A']$ , donde  $A' = A \cup \{a\}$ . Entonces el número de aristas de  $G'$  es  $n$ . Por lo tanto se tiene que  $G'$  tiene, al menos un ciclo (proposición 2). Supóngase que  $G'$  tiene, al menos dos ciclos. Sean éstos  $(x = x_1, x_2, \dots, x_i = y, x)$  y  $(x = y_1, y_2, \dots, y_j = y, x)$ ; entonces  $(x = x_1, x_2, \dots, x_i = y = y_j, y_{j-1}, y_{j-2}, \dots, y_1 = x)$  es un ciclo que no contiene la arista  $a$  y, por lo tanto, está contenido en  $G$ . Esto es una contradicción ya que  $G$  es, por hipótesis, acíclica. Por lo tanto  $G'$ , tiene exactamente un ciclo.

(c implica d). Supóngase que  $G$  no es conexa; entonces existen dos vértices  $x$  y  $y$  tales que no existe una cadena, en  $G$ , que los une. Luego,  $x$  y  $y$  están en dos componentes conexas distintas de  $G$ . De la proposición 1 se tiene que  $G' = [X, AU\{(x,y)\}]$  es acíclica. Esto es una contradicción a las hipótesis de c. Por lo tanto  $G$  es conexa. Además por hipótesis,  $G$  es acíclica. Entonces  $G$  tiene  $n-1$  aristas.

(d implica e). Sea  $a$  una arista de  $G$ . Sea  $G' = [X, A - \{a\}]$ . El número de aristas de  $G'$  es  $n-2$ ; entonces, por la proposición 2,  $G'$  no es conexa.

(e implica f). Como  $G$  es conexa existe, al menos, una cadena entre todo par de vértices. Supóngase que entre los vértices  $x$  y  $y$  existen, al menos, dos cadenas. Sean  $C_1$  y  $C_2$  dos de tales cadenas. Sea  $a$  una arista de  $C_1$  que no está en  $C_2$ . Entonces la gráfica  $G' = [X, A - \{a\}]$  es conexa. Esto es una contradicción a la hipótesis. Por lo tanto, la cadena que une a un par de vértices es única.

(f implica a). Que  $G$  sea conexa es inmediato de la hipótesis. Supóngase que  $G$  tiene un ciclo. Sea éste  $(x_1, x_2, \dots, x_1, x_1)$ . Entonces existen dos cadenas entre  $x_1$  y  $x_1$ , a saber:  $(x_1, \dots, x_1)$  y  $(x_1, x_1)$ . Esto es una contradicción a la hipótesis. Por lo tanto  $G$  es acíclica.

Con ésto se concluye la demostración del teorema. ■

Otra de las propiedades importantes de los árboles es que existen ciertas operaciones que, al aplicarse a un árbol, producen otro árbol. Una de estas operaciones es: se agrega un vértice y se conecta al árbol por medio de una arista. La otra es: se eliminan del árbol un vértice que tiene una sola arista adyacente y dicha arista. Estas dos operaciones serán utilizadas, posteriormente, en el método simplex especializado en redes. En el capítulo donde se presenta este último método se probará que las soluciones básicas del problema de programación lineal, asociado al problema de flujo a costo mínimo, corresponden a árboles expandidos de la red de flujos y viceversa; se probará también que las bases de este problema son triangularizables. Precisamente en esta parte se utilizarán las operaciones mencionadas anteriormente.

La justificación de conservación de árboles bajo estas operaciones se dará en la proposición 5; sin embargo, antes de establecer ésta considérese la siguiente definición.

Definición. Sea  $G=[X,A]$  una gráfica. Sea  $x$  un vértice de  $G$ . Se dice que  $x$  es un vértice pendiente si su grado es uno.

En otras palabras, un vértice pendiente en una gráfica, es aquél que tiene una sola arista adyacente. Los árboles tienen, al menos, dos vértices con esta característica; esto se demuestra en la proposición 4.

Proposición 4. Sea  $G=[X,A]$  un árbol con  $n$  vértices, donde  $n > 2$ . Entonces  $G$  tiene, al menos, dos vértices pendientes.

Demostración. Obsérvese, primeramente, que la suma de los grados de los vértices, en una gráfica no dirigida, es dos veces el número de aristas puesto que cada arista contribuye dos veces a la suma. Sea  $n$  el número de vértices de  $G$ . Puesto que  $G$  es un árbol, tiene  $n-1$  aristas; luego la suma de los grados de todos los vértices de  $G$  es  $2(n-1)$ . Por otro lado, puesto que  $G$  es conexa, el grado de todo vértice es mayor que cero.

Supóngase que todos los vértices de  $G$  tienen grado mayor que uno. Entonces la suma de los grados de todos los vértices es mayor o igual que  $2n$  ( $> 2(n-1)$ ), lo cual es una contradicción. Por lo tanto  $G$  tiene, al menos, un vértice pendiente. Supóngase ahora que todos los vértices de  $G$ , excepto uno, tienen grado mayor que uno. Entonces la suma de los grados de todos los vértices es mayor o igual que  $2n-1$  ( $> 2(n-1)$ ), lo cual también es una contradicción. Por lo tanto  $G$  tiene, al menos, dos vértices pendientes y la prueba termina. ■

En la siguiente proposición se prueba lo referente a la conservación de árboles bajo las operaciones antes mencionadas.

Proposición 5. Sea  $G = [X, A]$  un árbol. Entonces:

- a. Si  $x$  es un vértice pendiente de  $G$  y  $a$  es la única arista adyacente a él, la gráfica  $G' = [X - \{x\}, A - \{a\}]$  es un árbol.
- b. Si se agrega un vértice  $x$  a  $G$  y se conecta a  $G$  por medio de una arista,  $a$ , la gráfica resultante,  $G' = [X \cup \{x\}, A \cup \{a\}]$  es un árbol.

Demostración. Sean  $n$  el número de vértices de  $G$ ,  $m$  el número de aristas de  $G$ ,  $n'$  el número de vértices de  $G'$  y  $m'$  el número de aristas de  $G'$ .

a. Por construcción de  $G'$  se tiene que  $n' = n - 1$  y  $m' = m - 1$ . Luego,  $m' = n' - 1$ ; además,  $G'$  es conexa. Entonces, por el teorema 3,  $G'$  es un árbol.

b. De nuevo, por construcción de  $G'$ , se tiene que  $n' = n + 1$  y  $m' = m + 1$ . Luego,  $m' = n' - 1$ ; además  $G'$  es conexa. Entonces,  $G'$  es un árbol y la prueba termina. ■

Considérese, ahora, el problema de encontrar el árbol expandido de peso mínimo en una gráfica con una función de peso asociada a sus aristas. Debe notarse que, para que el problema tenga solución, la gráfica debe ser conexa; si esto se satisface puede garantizarse la existencia de un árbol expandido y, por lo tanto, podrá procederse a la búsqueda del mejor árbol expandido. La existencia de dicho árbol se afirma en un corolario del teorema 3.

Corolario 6. (Del teorema 3). Sea  $G = [X, A]$  una gráfica conexa, entonces  $G$  posee un árbol expandido  $T = [X, A']$ .

Demostración. En efecto, si no existe una arista  $a \in A$ , tal que la gráfica  $G' = [X, A - \{a\}]$  es conexa, entonces  $G$  es un árbol expandido. Si sí existe tal arista, considérese ahora la gráfica conexa  $G'$  en vez de  $G$  y repítase este procedimiento. ■

En la siguiente sección del capítulo se presentan métodos de determinación del árbol expandido de peso mínimo en una gráfica conexa.

### 1.3 METODO DE SOLUCION Y JUSTIFICACION.

En esta parte del capítulo se presentan dos métodos de solución para el problema de determinar el árbol expandido de peso mínimo en una red conexa  $R = [X, A, p]$ , con  $n$  vértices, así como la justificación de los mismos.

El primer método es el algoritmo de Kruskal también conocido como algoritmo "glotón". Este método consiste en ordenar las aristas en orden ascendente de peso. Las aristas se irán examinando en el orden establecido y serán consideradas en el árbol si no forman ciclo con las anteriormente consideradas; de este modo se obtendrá una gráfica acíclica. El algoritmo termina cuando el número de aristas consideradas sea igual al número de vértices menos uno garantizando, de esta manera, la generación de un árbol expandido de  $G$ . Obsérvese que, en cada iteración, la gráfica formada por las aristas consideradas y sus extremos no necesariamente es conexa, excepto en la última iteración.

Debe también notarse que el número de iteraciones será, al menos, el número de vértices menos uno (en el caso de que las primeras  $n-1$  aristas en el orden establecido no formen ciclo) y, a lo más, el número de aristas.

## ALGORITMO DE KRUSKAL

Propósito. Determinar el árbol expandido de peso mínimo en una gráfica  $G = [X, A]$  conexa con  $n$  vértices y  $m$  aristas.

### Descripción

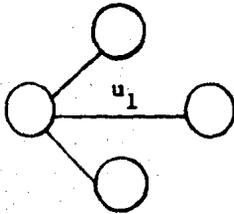
1. Ordénese el conjunto de aristas de manera creciente con respecto a la función de peso. Sean  $a_1, a_2, \dots, a_m$  las aristas ordenadas. Hacer  $k = j = 0$  y  $A' = \emptyset$ .
2. Hacer  $j = j + 1$ . Si la arista  $a_j$  no forma ciclo con las aristas de  $A'$  entonces  $A' = A' \cup \{a_j\}$ . Hacer  $k = k + 1$  e ir a 3. Si la arista  $a_j$  forma ciclo ir a 3.
3. Si  $k = n - 1$  terminar. La gráfica  $T = [X, A']$  es el árbol expandido de peso mínimo de  $G$ . Si  $k < n - 1$  regrésese a 2.

Justificación del algoritmo. Sea  $T$  la gráfica generada por el algoritmo. Por construcción  $T$  es acíclica y tiene  $n - 1$  aristas; por lo tanto,  $T$  es un árbol expandido de  $G$ . Ahora se demostrará que  $T$  es de peso mínimo. Sea  $S$  un árbol de peso mínimo de  $G$  y supóngase que  $T$  es distinto de  $S$ . Basta probar que  $T$  y  $S$  tienen el mismo peso para que el algoritmo quede justificado. Para ello se procederá de la siguiente manera:

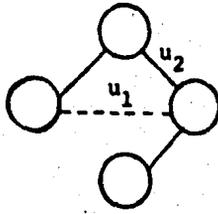
A partir de los árboles  $T$  y  $S$  se construirá otro árbol,  $S_1$ , de peso mínimo de  $G$  con la característica de que es más "parcido" a  $T$  que  $S$ ; es decir, si  $T$  y  $S$  tienen  $h$  aristas en común,  $T$  y  $S_1$  tendrán  $h + 1$  aristas en común. Después, si  $T \neq S_1$ , a partir de estos dos árboles se construirá otro árbol de peso mínimo de  $G$ ,  $S_2$ , que tendrá  $h + 2$  aristas en común con  $T$ . Como el procedimiento propuesto es finito, llegará un momento en el que se construirá un árbol  $S_k$  de peso mínimo de  $G$  que tendrá  $n - 1$  aristas en común con  $T$ ; es decir, se tendrá  $S_k = T$  y se podrá concluir que  $T$  es un árbol de peso mínimo de  $G$ .

Para la construcción del árbol  $S_r$  ( $r=1, \dots, k$ ) considérese lo siguiente: Si  $T \neq S_{r-1}$  ( $S_0 = S$ ) difieren, al menos, en una arista. Sea  $u_1$  la arista de menor peso que está en  $T$  pero no en  $S_{r-1}$ ; es decir,  $p(u_1) = \min \{p(a)\}$ , para toda arista  $a \in T$  tal que  $a \notin S_{r-1}$ . Obsérvese que  $u_1$  es la primera arista

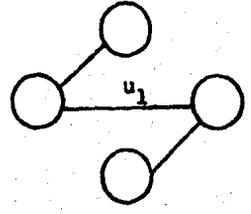
examinada, durante el algoritmo, que pertenece a  $T$  pero no a  $S_{r-1}$ . Por otro lado, como  $S_{r-1}$  es árbol, existe en él una cadena que une los extremos de  $u_1$ . Esta cadena contiene una arista,  $u_2$ , que no pertenece a  $T$ , puesto que si estuviera toda la cadena se formaría un ciclo con  $u_1$  en  $T$ . Agréguese  $u_1$  a  $S_{r-1}$  y elimínese  $u_2$  de  $S_{r-1}$ . Sea  $S_r$  la gráfica resultante y obsérvese que ésta es un árbol de  $G$ .



Arbol  $T$



Arbol  $S_{r-1}$



Arbol  $S_r$

Ahora debe probarse que  $S_r$  es de peso mínimo. Esto se hará por inducción. Para  $s = 0$ , se tiene que  $S_s = S$  es un árbol de peso mínimo de  $G$ . Supóngase válida la afirmación para  $s = r - 1$ ; ésto es, supóngase que  $S_{r-1}$  es un árbol de peso mínimo de  $G$ . Esto implica que

$$p(S_{r-1}) \leq p(S_r)$$

y como  $S_{r-1}$  y  $S_r$  difieren sólo en una arista se tiene:

$$p(u_2) \leq p(u_1) \dots (*)$$

Basta probar que  $p(u_1) \leq p(u_2)$  para poder concluir que

$p(S_{r-1}) = p(S_r)$ . Para ésto, supóngase que  $p(u_2) < p(u_1)$ ; nótese que esta suposición implica que la arista  $u_2$  fué examinada antes que la arista  $u_1$  durante el algoritmo de Kruskal que generó  $T$ . Por otro lado, como  $u_2$  no fué incluida en  $T$ , esta arista forma ciclo con las aristas examinadas antes que ella y que fueron incluidas en  $T$ . Sean éstas  $b_1, b_2, \dots, b_j$ . Debe observarse que  $p(b_i) < p(u_1)$ , para  $i=1, 2, \dots, j$ ; luego, dada la definición de  $u_1$ , se concluye que  $b_1, b_2, \dots, b_j$  pertenecen a  $S_{r-1}$ . Esto es una contradicción puesto que  $b_1, b_2, \dots, b_j$  y  $u_2$  formarían ciclo en  $S_{r-1}$ . Por lo tanto se concluye

$$p(u_1) \leq p(u_2)$$

Esta expresión junto con la expresión (\*) implica

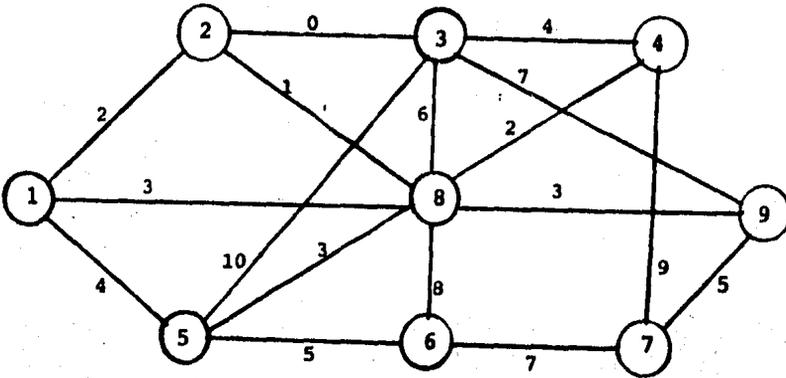
$$p(u_1) = p(u_2)$$

y por lo tanto:

$$p(S_{r-1}) = p(S_r)$$

es decir,  $S_r$  es un árbol de peso mínimo de  $G$ . Nótese que  $T$  y  $S_r$  contienen ambos a la arista  $u_1$  (que no estaba contenida en  $S_{r-1}$ ); luego, si  $T$  y  $S_{r-1}$  tienen  $h$  aristas en común,  $T$  y  $S_r$  tienen  $h + 1$  aristas en común. Si  $T = S_r$  entonces  $T$  es de peso mínimo; en caso contrario, constrúyase  $S_{r+1}$ , del mismo modo que como se construyó  $S_r$ , a partir de  $T$  y  $S_r$ . Este procedimiento es finito puesto que el número de aristas de  $T$  es finito. Con ésto queda justificado el algoritmo. ■

Ejemplo 1. Considérese la siguiente gráfica:



Determinese el árbol de peso mínimo en la gráfica mediante el algoritmo de Kruskal.

Iteración 1. Se ordenan las aristas:

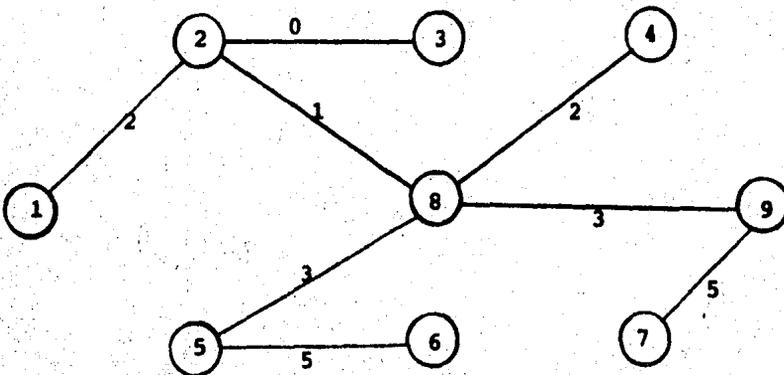
$a_1 = (2,3)$ ,  $a_2 = (2,8)$ ,  $a_3 = (1,2)$ ,  $a_4 = (8,4)$ ,  $a_5 = (1,8)$ ,  $a_6 = (5,8)$   
 $a_7 = (8,9)$ ,  $a_8 = (1,5)$ ,  $a_9 = (3,4)$ ,  $a_{10} = (5,6)$ ,  $a_{11} = (7,9)$ ,  $a_{12} = (3,8)$   
 $a_{13} = (6,7)$ ,  $a_{14} = (3,9)$ ,  $a_{15} = (8,6)$ ,  $a_{16} = (4,7)$  y  $a_{17} = (3,5)$ .

$A' = \{a_1\}$ ;  $k = 1$ . Puesto que  $k < n - 1 = 8$ , aún no se tiene el árbol.

En el siguiente cuadro se presenta un resumen de las operaciones realizadas en cada iteración. En la primera columna aparece el número de iteración; en la segunda, la arista que se agrega al conjunto  $A'$  para ir formando el árbol; por último, en la tercera columna se tiene el número de aristas que ya se han incluido en el árbol hasta esa iteración.

Número de iteración (j)	Arista agregada a A'	Valor de k
1	$a_1 = (2, 3)$	1
2	$a_2 = (2, 8)$	2
3	$a_3 = (1, 2)$	3
4	$a_4 = (8, 4)$	4
5	ninguna ( $a_5$ forma ciclo)	4
6	$a_6 = (5, 8)$	5
7	$a_7 = (8, 9)$	6
8	ninguna ( $a_8$ forma ciclo)	6
9	ninguna ( $a_9$ forma ciclo)	6
10	$a_{10} = (5, 6)$	7
11	$a_{11} = (7, 9)$	8

Puesto que en la iteración 11 se tiene que  $k = 8 = n - 1$ , se ha obtenido la solución óptima; ésta está dada por la gráfica  $T = [X, A']$ , donde  $A' = \{a_1, a_2, a_3, a_4, a_6, a_7, a_{10}, a_{11}\}$



Este árbol expandido tiene un peso de  $0+1+2+2+3+3+5+5 = 21$ .

## Método Alternativo de Solución.

El método alternativo de solución para el problema del árbol de peso mínimo de una red conexa con  $n$  vértices, es el algoritmo de Prim. Este algoritmo consiste en considerar, inicialmente, una gráfica formada por cualquier vértice de la gráfica; después se agregará la arista de menor peso adyacente a él y su otro extremo. Luego se aumenta la arista más pequeña, que tenga exactamente un extremo en la gráfica formada, junto con su otro extremo. Se procede de esta manera, sucesivamente, hasta tener  $n-1$  aristas en la gráfica generada.

A diferencia del algoritmo de Kruskal, la gráfica construida en cada iteración es conexa. En particular, la última gráfica obtenida es conexa y además tiene  $n-1$  aristas; luego, esta gráfica es un árbol expandido de la gráfica original. Debe notarse también, que el algoritmo termina en  $n-1$  iteraciones exactamente; esto constituye otra diferencia con el algoritmo de Kruskal.

## ALGORITMO DE PRIM

Propósito. Determinar el árbol expandido de peso mínimo en una gráfica  $G = [X, A]$  conexa con  $n$  vértices.

### Descripción

1. Sea  $x_0$  cualquier vértice de la gráfica  $G$ ; sean  $X_0 = \{x_0\}$  y  $A_0 = \phi$ . Hacer  $k = 0$ .
2.  $k = k + 1$
3. Sea  $C_k$  el conjunto de aristas que tienen exactamente un extremo en  $X_{k-1}$ . Sea  $a_k$  la arista de peso mínimo de  $C_k$  y sea  $x_k$  el extremo de  $a_k$  que no pertenece a  $X_{k-1}$ . Hacer

$$X_k = X_{k-1} \cup \{x_k\} ; A_k = A_{k-1} \cup \{a_k\}$$

4. Si  $k < n-1$  ir a (2). Si  $k = n-1$  terminar. La gráfica  $T_{n-1} = [X_{n-1}, A_{n-1}]$  es un árbol expandido de peso mínimo de  $G$ .

Justificación del algoritmo. Basta demostrar que cada gráfica  $T_k = [X_k, A_k]$ ,  $k = 1, 2, \dots, n-1$  es una subgráfica conexa, con  $k+1$  vértices, de un árbol de peso mínimo de  $G$ . Esto se hará por inducción sobre  $k$ .

Para  $k=0$  la afirmación se cumple puesto que  $T_0$  consta de un solo vértice. Supóngase que la afirmación es válida para  $k=r-1$ . Ahora se demostrará que también lo es para  $k=r$ . Ya que  $T_{r-1}$  tiene  $r$  vértices y es conexa entonces, por construcción,  $T_r$  tiene  $r + 1$  vértices y es conexa.

Sea  $T^* = [X, A^*]$  un árbol expandido de peso mínimo de  $G$  que contiene todas las aristas de  $T_{r-1}$ . Si  $a_r$  es una arista de  $T^*$ , entonces  $T_r$  es una subgráfica conexa de  $T^*$ . Si, al contrario,  $T^*$  no contiene a  $a_r$ , entonces al agregar  $a_r$  a  $T^*$  se formará un ciclo con  $a_r$  y la cadena que une sus extremos en  $T^*$ . Dada la definición del conjunto  $C_r$ , dicha cadena contiene una arista de él; sea ésta  $u$ .

Sea el árbol  $T' = [X, A]$ , donde  $A' = (A^* \cup \{a_r\}) - \{u\}$ . Nótese que  $T_r$  es una subgráfica conexa del árbol  $T'$ . Ahora, puesto que

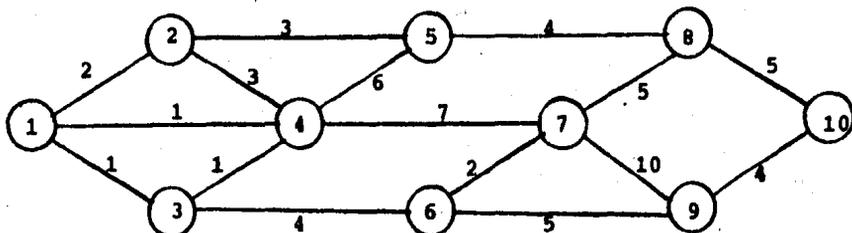
$$p(a_k) = \min_{a \in C_k} (p(a)), \text{ para todo } k = 1, 2, \dots, n - 1$$

se tiene que:

$$p(T') \leq p(T^*)$$

y como  $T^*$  es un árbol expandido de peso mínimo de  $G$ , entonces  $T'$  también lo es y, por lo tanto, la afirmación es válida. ■

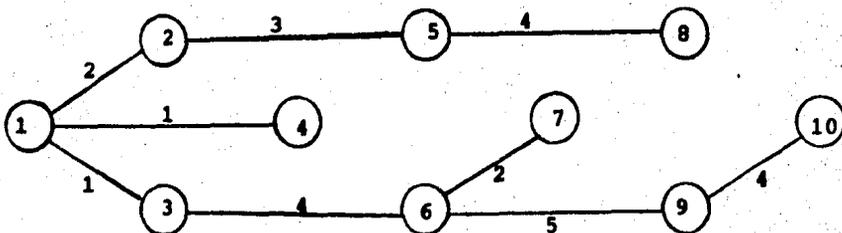
EJEMPLO 2. Considérese la siguiente gráfica.



Determinése el árbol expandido de peso mínimo mediante el algoritmo de Prim.

Se empezará a aplicar el algoritmo definiendo  $X_0 = \{7\}$ .

En el siguiente cuadro se presentan, resumidas, las iteraciones del algoritmo. En la primera columna aparece el número de iteración ( $k$ ), en la segunda el conjunto  $C_k$  resultante. en la  $k$ -ésima iteración, en la tercera la arista de peso mínimo de  $C_k$  ( $a_k$ ), y, por último, en las columnas cuarta y quinta respectivamente, los vértices y las aristas que irán formando el árbol expandido de peso mínimo de la gráfica. En la iteración  $k=n-1=9$  puede observarse que el árbol expandido de peso mínimo generado por el algoritmo es:



cuyo peso es:  $2+4+1+1+2+3+4+5+4 = 26$ .

Iteración (k)	$C_k$	$a_k$	$x_k$	$A_k$
1	{(4,7), (6,7), (7,8), (7,9)}	(6, 7)	{7,6}	{(6,7)}
2	{(3,6), (6,9), (4,7), (7,8), (7,9)}	(3, 6)	{7,6,3}	{(6,7), (3,6)}
3	{(1,3), (3,4), (6,9), (4,7), (7,8), (7,9)}	(1, 3)	{7,6,3,1}	{(6,7), (3,6), (1,3)}
4	{(1,2), (1,4), (3,4), (6,9), (4,7), (7,8), (7,9)}	(1, 4)	{7,6,3,1,4}	{(6,7), (3,6), (1,3), (1,4)}
5	{(1,2), (2,4), (4,5), (6,9), (7,8), (7,9)}	(1, 2)	{7,6,3,1,4,2}	{(6,7), (3,6), (1,3), (1,4), (1,2)}
6	{(2,5), (4,5), (6,9), (7,8), (7,9)}	(2, 5)	{7,6,3,1,4,2,5}	{(6,7), (3,6), (1,3), (1,4), (1,2), (2,5)}
7	{(5,8), (6,9), (7,8), (7,9)}	(5, 8)	{7,6,3,1,4,2,5,8}	{(6,7), (3,6), (1,3), (1,4), (1,2), (2,5), (5,8)}
8	{(6,9), (7,9), (8,10)}	(6, 9)	{7,6,3,1,4,2,5,8,9}	{(6,7), (3,6), (1,3), (1,4), (1,2), (2,5), (5,8), (6,9)}
9	{(8,10), (9,10)}	(9,10)	{7,6,3,1,4,2,5,8,9,10}	{(6,7), (3,6), (1,3), (1,4), (1,2), (2,5), (5,8), (6,9), (9,10)}

#### 1.4 ANALISIS DE SENSIBILIDAD

Una vez presentados los métodos de solución para el problema del árbol de expansión mínima es interesante observar el efecto de eliminar una arista de la gráfica original. Es claro que, si esta arista no pertenece a la solución óptima, la solución óptima del problema de encontrar el árbol de peso mínimo en la gráfica resultante de remover dicha arista, es la misma que para la gráfica original. Sin embargo, si la arista forma parte de la solución óptima, es importante contar con una herramienta que permita resolver el nuevo problema aprovechando los resultados que ya se han obtenido. Enseguida se analiza este problema.

Supóngase que se ha obtenido el árbol expandido de peso mínimo  $T = [X, A^*]$  de una gráfica conexa  $G = [X, A]$ . Supóngase que, por alguna razón, es necesario eliminar una arista de la gráfica  $G$  que, además, pertenece al árbol  $T$ . Considérese, ahora, el problema de encontrar el árbol expandido de peso mínimo de la gráfica resultante de eliminar dicha arista.

Sea  $a = (x, y)$  la arista que debe eliminarse y sea  $G'$  la gráfica resultante de tal eliminación; es decir,  $G' = [X, A - \{a\}]$ . Entonces, el problema consiste en encontrar el árbol expandido de peso mínimo de  $G'$ . Es claro que una manera de resolver el problema es aplicar, a  $G'$ , algún algoritmo para encontrar el árbol requerido; pero el procedimiento que se describe a continuación, permite obtenerlo aprovechando que ya se cuenta con el árbol  $T$ .

Al eliminar  $(x,y)$  de  $T$ , éste se descompone en dos componentes conexas. Sean éstas  $T_1=[X_1, A_1]$  y  $T_2=[X_2, A_2]$ . Obsérvese que un extremo de la arista  $a$  está en  $X_1$  y el otro en  $X_2$ . Supóngase, sin pérdida de generalidad, que  $x$  está en  $X_1$  y que  $y$  está en  $X_2$ . Sea  $U$  el conjunto de aristas de  $G$  distintas de la arista  $a$  que tienen un extremo en  $X_1$  y otro en  $X_2$ . Obsérvese que si  $U$  es vacío el problema no tiene solución puesto que  $G'$  no sería conexa y por lo tanto no existe árbol expandido de  $G'$ . Supóngase, entonces, que  $U$  no es vacío. Sea  $u^*$  la arista de  $U$  tal que  $p(u^*) \leq p(u)$ , para toda arista  $u$  de  $U$ .

La solución del problema queda establecida en la siguiente proposición.

Proposición 7. La gráfica  $T^*=[X, A_1 \cup a \cup U \cup \{u^*\}]$  es árbol expandido de peso mínimo de  $G'$ .

Demostración. Sean las gráficas conexas  $G_1=[X_1, A^1]$  y  $G_2=[X_2, A^2]$ , donde  $A^i$  es el conjunto de aristas de  $A$  que tienen ambos extremos en  $X_i$  ( $i=1,2$ ). Se demostrará primero que  $T_1$  y  $T_2$  son los árboles expandidos de peso mínimo de  $G_1$  y  $G_2$  respectivamente. Supóngase que  $T_1$  no es un árbol expandido de peso mínimo de  $G_1$ . Sea  $N=[X_1, U_1]$  el árbol expandido de peso mínimo de dicha gráfica; se tiene, entonces, que  $p(N) < p(T_1)$ . Sea  $T'=[X, U_1 \cup a \cup U \cup \{u^*\}]$ . Nótese que  $T'$  es un árbol expandido de  $G$ . En efecto,  $T'$  es acíclica ya que la arista  $a$  tiene un extremo en  $X_1$  y otro en  $X_2$  y las gráficas  $N$  y  $T_2$  son acíclicas. Ahora bien,

$$p(T') = p(U_1) + p(A_2) + p(a) = p(N) + p(A_2) + p(a)$$

por otro lado

$$p(T) = p(A_1) + p(A_2) + p(a) = p(T_1) + p(A_2) + p(a)$$

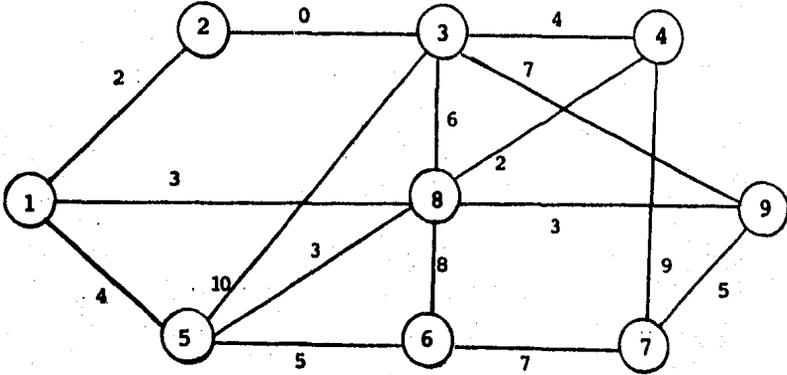
por lo tanto,  $p(T') < p(T)$ ; ésto contradice que  $T$  sea un árbol expandido de peso mínimo de  $G$ . Por lo tanto,  $T_1$  es un árbol expandido de peso mínimo de  $G_1$ . Análogamente se prueba que  $T_2$  es un árbol expandido de peso mínimo de  $G_2$ .

Por otro lado,  $G'$  puede expresarse como la gráfica  $[X, A^1 \cup A^2 \cup U]$  y, puesto que  $p(u^*) \leq p(u)$  para toda arista  $u$  de  $U$ , se tiene que  $T^*$  es un árbol expandido de peso mínimo de  $G'$  y con ésto se concluye la prueba. ■

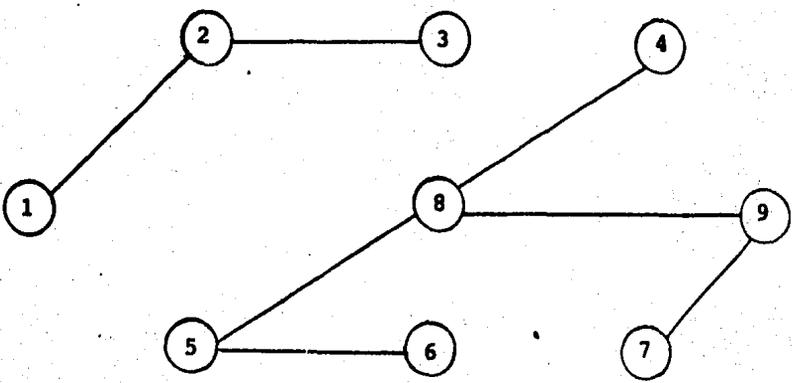
Es importante hacer notar que este mismo procedimiento puede aplicarse en el siguiente problema:

Supóngase que se ha encontrado el árbol de peso mínimo,  $T=[X, A^*]$ , de la gráfica  $G=[X, A]$  y supóngase también que se modifica el peso de una arista a  $\epsilon A^*$ . Se desea encontrar ahora el árbol de peso mínimo en la gráfica  $G$  con la función de peso, asociada a las aristas de  $G$ , modificada para la arista  $a$ . Si el nuevo peso de  $a$  es menor que el que tenía anteriormente, es claro que la solución óptima seguirá siendo la misma; el problema surge cuando el nuevo peso de  $a$  es mayor que su peso anterior. Es en este último caso cuando puede aplicarse el método anteriormente expuesto. En efecto, para encontrar la nueva solución, la única modificación que deberá hacerse al procedimiento será no excluir a la arista  $a$  del conjunto  $U$ .

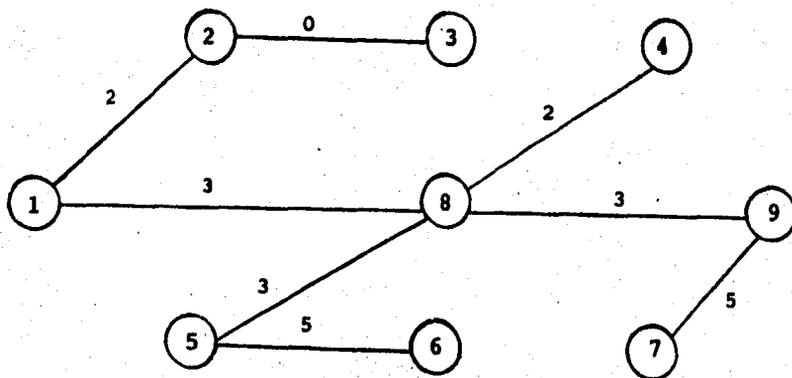
**Ejemplo 3.** . Considérese la gráfica del ejemplo 1 y su árbol expandido de peso mínimo. Supóngase que se elimina la arista (2,8). La gráfica resultante,  $G'$ , es:



**Determinase** el árbol expandido de peso mínimo en esta nueva gráfica. Las componentes conexas en que se divide  $T$  al eliminar (2,8) son:



El conjunto  $U$  estará formado, entonces, por las aristas:  
 $(1,5), (1,8), (3,4), (3,5), (3,8)$  y  $(3,9)$ , cuyos pesos son:  
 $4, 3, 4, 10, 6$  y  $7$  respectivamente. De aquí se tiene que  
 $u^* = (1,8)$ . Por lo tanto, el árbol expandido de peso mínimo  
de  $G'$  es:



cuyo peso es:  $0+2+2+3+3+3+5+5 = 23 = p(T) - p(2,8) + p(1,8)$ .

**RUTA MAS CORTA**

## 2.1 DESCRIPCION DEL PROBLEMA

En este capítulo se presentan métodos de solución para los siguientes problemas de rutas más cortas en una red  $R=[X,A,d]$ :

1. Ruta más corta entre dos vértices específicos  $s$  y  $t$ .
2. Rutas más cortas entre un vértice específico  $s$  y todo vértice  $x$  en la red.
3. Ruta más corta entre todo par de vértices.

Para ejemplificar considérese el siguiente problema:

En una terminal de camiones para pasajeros se desea establecer la ruta que deberá seguir el autobús que presta servicio de la ciudad  $s$  a la ciudad  $t$  de tal manera que la distancia recorrida sea lo más corta posible. A este problema se le puede asociar una red  $R = [X,A,d]$  donde:

$X = \{\text{Ciudades a las cuales se ofrece el servicio}\}$

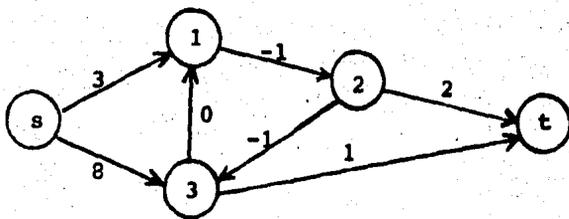
$A = \{\text{Tramos de carretera entre las ciudades}\}$

$d : A \rightarrow R$  donde, para todo  $a \in A$ ,  $d(a) = \text{longitud o distancia del tramo de carretera } a$ .

En general, en una red  $R = [X,A,d]$ , al número  $d(a)$  asociado a cada arco se le llama longitud o costo de  $a$ . Por otro lado se define la longitud de una ruta o camino como la suma de las longitudes de los arcos que la forman; aquella ruta tal que su longitud sea mínima se le llama ruta más corta o camino más corto.

El problema de la terminal de autobuses es entonces encontrar la ruta más corta entre dos vértices específicos; los que representan a las ciudades  $s$  y  $t$ . Obsérvese que en este caso las longitudes definidas son no negativas; sin embargo el problema de encontrar la ruta más corta entre dos vértices específicos puede generalizarse a cualquier red puesto que la función de longitud,  $d$ , puede representar, además de distancia o tiempo, costos o alguna otra cantidad.

Si la red contiene arcos con longitudes negativas pueden presentarse circuitos negativos (circuitos de longitud negativa). En este caso el problema puede ser no acotado puesto que dada cualquier ruta entre  $s$  y  $t$  que contenga al circuito negativo existe otra mejor, a saber aquella que contiene una vez más al circuito. En la siguiente red ocurre esto:



Una ruta de  $s$  a  $t$  es:  $s, 3, 1, 2, 3, t$  de longitud 7. Otra ruta de  $s$  a  $t$ , mejor que la anterior, es:  $s, 3, 1, 2, 3, 1, 2, 3, t$ , de longitud 5. De hecho si se consiera una ruta que contenga el arco  $(s, 3)$ ,  $M$  veces el circuito  $3, 1, 2, 3$  ( $M > 0$ ) y el arco  $(3, t)$ ,

la longitud será  $9-2M_i$  de tal modo que si  $M$  tiende a infinito, la longitud de la ruta tiende a menos infinito. Luego, la ruta más corta entre  $s$  y  $t$  no existe.

Se concluye entonces que, para que el problema de la ruta más corta entre dos vértices específicos tenga solución, deberá cumplirse que:

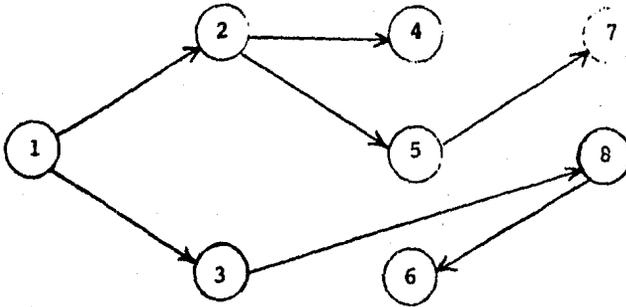
- (i) Exista algún camino entre  $s$  y  $t$ ;
- (ii) No existan circuitos negativos tales que haya un camino de  $s$  a algún vértice del circuito y otro de algún vértice del circuito a  $t$ .

Supóngase ahora que en la terminal de autobuses se desea mejorar el servicio que se proporciona a la ciudad  $s$ ; con este objeto se requiere encontrar las rutas más cortas entre la ciudad  $s$  y todas las demás ciudades a las cuales se da el servicio. A este problema puede asociarse, de nuevo, la red definida anteriormente.

Antes de analizar cuándo tiene solución este problema se definirán unos conceptos que serán de utilidad.

Sea  $G = [X, A]$  una gráfica dirigida y sea  $s \in X$ ; entonces  $s$  se llama raíz de  $G$ , si existe un camino de  $s$  a  $x$  para todo  $x \in X$ .

Una arborescencia es un árbol que tiene una raíz. La siguiente gráfica es una arborescencia de raíz  $l$ :



Sea  $G=[X,A]$  una gráfica dirigida. Una arborescencia de  $G$  es un árbol expandido de  $G$  que tiene un vértice que es raíz.

Se demostrará, más adelante, que en una arborescencia de raíz  $s$  el camino de  $s$  a  $x$ , para todo  $x \in X$ , es único.

Considérese ahora una red  $R=[X,A,d]$ . Una arborescencia de rutas más cortas de  $R$  es aquella arborescencia tal que la única ruta de  $s$  a  $x$ , para todo  $x \in X$ , es una ruta más corta de  $s$  a  $x$ .

Una vez definidos estos conceptos, puede decirse que el problema de la terminal de autobuses es encontrar la arborescencia de rutas más cortas de raíz  $s$  de la red  $R=[X,A,d]$ .

Haciendo una analogía con el problema de la ruta más corta entre dos vértices específicos puede concluirse que para que exista la arborescencia de rutas más cortas de raíz  $s$  en una red cualquiera  $R=[X,A,d]$ , ésta deberá cumplir que:

(i) Existen caminos de  $s$  a  $x$ , para todo  $x \in X$ . Es decir, que  $s$  sea raíz de la red.

(ii) No existen circuitos negativos en la red  $R$ ; ya que de presentarse éstos el problema sería no acotado (por la misma razón expuesta para el problema de ruta más corta entre dos vértices específicos).

Finalmente, supóngase que en la terminal de autobuses se tiene interés en encontrar las rutas más cortas para todos los camiones que prestan servicio entre cada par de ciudades. Nuevamente, se asocia a este problema la red  $R = [X, A, d]$  definida anteriormente; se deben encontrar, entonces, las rutas más cortas entre todo par de vértices en la red  $R$ .

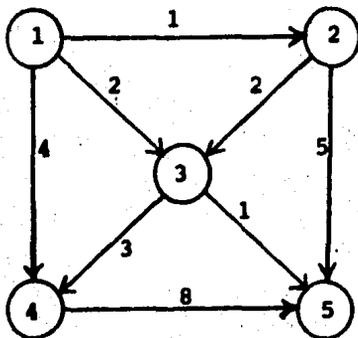
Este último problema tratado es una generalización inmediata de los anteriores. Por esto se deduce que, para que exista solución en cualquier red  $R = [X, A, d]$ , deberá cumplirse lo siguiente:

- (i) Existe, al menos, un camino entre todo par de vértices.
- (ii) No existen circuitos negativos en la red  $R$ .

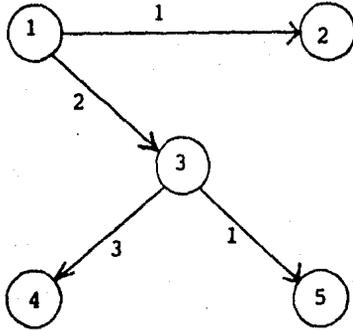
Antes de exponer los métodos de solución para estos tres problemas, se demostrarán algunas de las propiedades de las arborescencias.

## 2.2 CARACTERIZACION DE UNA ARBORESCENCIA

Existen ciertas propiedades de las arborescencias que serán utilizadas en la búsqueda de la solución del problema de la arborescencia de rutas más cortas de raíz  $x_0$ . Estas propiedades se demuestran en el siguiente teorema en el cual se postulan las distintas caracterizaciones de este tipo de gráficas. Es importante hacer notar que, en general, la arborescencia de rutas más cortas es distinta del árbol del peso mínimo de una red puesto que, mientras que el segundo concepto es no orientado el primero solo es aplicable a redes dirigidas. Aún en el caso en que la red sea dirigida, un árbol no necesariamente es arborescencia ya que éste puede no tener raíz. Por esto, para determinar la arborescencia de rutas más cortas no es posible aplicar ninguno de los algoritmos presentados en el capítulo anterior. Podría suceder incluso que un árbol de peso mínimo resulte ser una arborescencia; sin embargo, de ningún modo puede garantizarse que ésta sea de rutas más cortas. Por ejemplo, en la siguiente red:



Un árbol de peso mínimo resulta ser:



el cual, resulta ser una arborescencia de raíz 1. Sin embargo, la única ruta en la arborescencia entre los vértices 1 y 4 tiene una longitud de 5 unidades y existe otra ruta de longitud menor entre estos dos vértices en la red; en efecto, la ruta 1,4 tiene una longitud de 4 unidades. Por lo tanto, puede concluirse que la arborescencia no es de rutas más cortas.

En la siguiente parte del capítulo se presentarán los métodos de solución para el problema de la arborescencia de rutas más cortas, así como para los otros dos problemas anteriormente descritos.

Teorema. Sea  $G=(X,A)$  una gráfica con  $n$  vértices. Supóngase que  $n \geq 2$ . Los postulados siguientes son equivalentes y caracterizan una arborescencia.

- a.  $G$  es un árbol y tiene un vértice  $x_0$  que es raíz.
- b. Para todo vértice  $x$  existe un camino único de  $x_0$  a  $x$ .
- c.  $G$  tiene al vértice  $x_0$  que es raíz y si se elimina un arco entonces  $x_0$  ya no es raíz.
- d.  $G$  es conexa,  $g^-(x_0)=0$  y  $g^-(x)=1$ , para todo vértice  $x$  distinto de  $x_0$ .
- e.  $G$  es acíclica,  $g^-(x_0)=0$  y  $g^-(x)=1$ , para todo vértice  $x$  distinto de  $x_0$ .
- f.  $G$  tiene como raíz a  $x_0$  y es acíclica.
- g.  $G$  tiene como raíz a  $x_0$  y posee  $n-1$  arcos.

Demostración. (a implica b) sea  $x$  un vértice de  $G$ . Puesto que  $x_0$  es raíz de  $G$  entonces existe un camino de  $x_0$  a  $x$ . Supóngase que existen, al menos, dos caminos de  $x_0$  a  $x$ . Entonces  $G$  tendría un ciclo y esto contradice que  $G$  es un árbol. Luego, el camino de  $x_0$  a  $x$  es único.

(b implica c) Puesto que existe un camino entre  $x_0$  y  $x$ , para todo vértice  $x$ , entonces  $x_0$  es raíz. Sea  $a=(x_i, x_j)$  un arco de  $G$  y supóngase que  $x_0$  es raíz de la gráfica  $G'=(X, A-(a))$ ; entonces existe un camino de  $x_0$  a  $x_j$  en  $G'$  (adtese que este camino no contiene a  $a$ ). Ahora, puesto que  $x_0$  es raíz de  $G$ , existe un camino de  $x_0$  a  $x_i$ ; este camino y el arco  $a$ , forman un camino de  $x_0$  a  $x_j$ ; entonces existen, en  $G$ ,

dos caminos distintos de  $x_0$  a  $x_j$ , lo cual es una contradicción. Por lo tanto,  $G$  no tiene a  $x_0$  como raíz.

(c implica d) Puesto que  $x_0$  es raíz de  $G$  entonces  $G$  es conexa. Supóngase que  $g^-(x_0) \neq 0$ ; es decir, que existe un arco  $(x, x_0)$ . Como  $x_0$  es raíz, entonces existe al menos un camino de  $x_0$  a  $x$ ; luego, al eliminar  $(x, x_0)$  de  $G$ ,  $x_0$  sigue siendo raíz. Esto es una contradicción y por lo tanto  $g^-(x_0) = 0$ . Por otro lado, por ser  $x_0$  raíz, se tiene que  $g^-(x) \geq 1$  para todo vértice  $x$  distinto de  $x_0$ . Supóngase que  $g^-(x) > 1$ ; es decir, que existen al menos dos arcos  $(x_1, x)$  y  $(x_j, x)$ . Entonces, al suprimir cualquiera de estos 2 arcos,  $x_0$  sigue siendo raíz de  $G$ . Por lo tanto  $g^-(x) = 1$ , para todo  $x$  distinto de  $x_0$ .

(d implica e) La gráfica  $G$  es conexa y tiene  $n-1$  arcos puesto que  $g^-(x_0) = 0$  y  $g^-(x) = 1$  para todo  $x \neq x_0$ . Entonces,  $G$  es un árbol y como consecuencia es acíclica.

(e implica f) Si  $x_0$  no fuera raíz entonces  $G$  tendría un ciclo puesto que  $g^-(x) = 1$ , para todo  $x \neq x_0$ .

(f implica g)  $G$  es acíclica y además conexa puesto que  $x_0$  es raíz. Entonces  $G$  es un árbol y por lo tanto tiene  $n-1$  arcos.

(g implica a)  $G$  tiene  $n-1$  arcos y es conexa puesto que  $x_0$  es raíz. Luego,  $G$  es un árbol. ■

### 2.3 METODOS DE SOLUCION Y JUSTIFICACION

En esta parte del capítulo se presentan los métodos de solución para el problema de la arborescencia de rutas más cortas de raíz  $s$  en una red  $R = [X, A, d]$ . El primero de ellos sólo es aplicable a redes que tienen arcos con costos no negativos y el segundo puede utilizarse para cualquier red.

En la práctica existe gran cantidad de problemas que involucran costos no negativos (tiempo, distancia, etc.); es por esta razón que se justifica el desarrollo de algoritmos que se aplican sólo a estos casos. Por otro lado, el método presentado para redes con esta característica, constituye el primer paso del algoritmo general posteriormente expuesto. Es importante mencionar que estos algoritmos tienen la ventaja de que, además de proporcionar la solución óptima cuando existe, detectan cuando ésta no existe; ya sea que dicha solución no exista porque  $s$  no es raíz de la red o por la presencia de circuitos negativos. Cabe señalar que estos métodos sirven también para encontrar la solución óptima del problema de la ruta más corta entre todo par de vértices.

Para el problema de la ruta más corta entre todo par de vértices se presenta un algoritmo con la misma ventaja que los anteriores. Más aún, si la ruta más corta existe para algunos pares de vértices y para otros no, el algoritmo proporciona las longitudes de las rutas existentes y detecta para qué pares de vértices no existe ninguna ruta.

Otra observación importante que debe hacerse es que, como se verá en capítulos posteriores, el problema de la ruta más corta resulta un subproblema del problema de flujo a costo mínimo. Otro subproblema que surge en la búsqueda del flujo a costo mínimo, es el de la detección de circuitos negativos por lo cual los métodos expuestos en este capítulo serán de gran utilidad.

#### a. ARBORESCENCIA DE RUTAS MAS CORTAS

##### Caso de redes con costos no negativos.

El método de solución presentado para el problema de la arborescencia de rutas más cortas en redes que tienen arcos con costos no negativos fué desarrollado por Dijkstra (1959) y está considerado como el método más eficiente para resolver este problema. Este método se basa en la asignación de etiquetas "permanentes" a los vértices para los cuales ya se conocen las longitudes de las rutas más cortas de la raíz a ellos. Sea  $S$  este conjunto de vértices.

Las etiquetas de los vértices de  $S$  representan precisamente las longitudes de las rutas más cortas buscadas. Los vértices restantes se etiquetan "temporalmente" con una cota superior de la longitud más corta de la raíz al vértice etiquetado. En la primera iteración el conjunto  $S$  contendrá únicamente al vértice raíz; es decir, sólo la raíz estará etiquetada permanentemente. Las etiquetas temporales se mejoran continuamente y en cada iteración se agrega exactamente un vértice  $x$  a  $S$ ; este vértice es aquél tal que la

longitud desde la raíz es la más corta posible. Puesto que to dos los arcos tienen costos no-negativos, siempre puede encon trarse una ruta más corta de la raíz a  $x$  que pase sólo por vér tices de  $S$ ; en este caso la etiqueta de  $x$  representa la longi- tud de la ruta más corta correspondiente. Una vez que todos los vértices estén en  $S$ , las etiquetas de todos los vértices se rán las correspondientes a las longitudes más cortas desde la raíz y por lo tanto se habrá encontrado la solución deseada. En el caso en que se desee sólo la ruta más corta entre dos vér tices específicos, se obtendrá la solución cuando se etiquete "permanentemente" el vértice final del camino buscado.

## ALGORITMO DE DIJKSTRA

Propósito. Obtener la arborescencia de rutas más cortas de raíz  $s$  en una red  $R = [X, A, d]$  con costos no negativos en los arcos.

### Descripción

Paso 1. (Inicialización de etiquetas). Sea  $d(s)=0$  y márchese esta etiqueta como permanente. Sea  $d(x)=\infty$ , para todo  $x \neq s$  y considérense estas etiquetas como temporales. Sean  $a(x)=x$  (estas etiquetas indicarán el predecesor de  $x$  en la arborescencia). Sea  $p = s$ .

Paso 2. (Actualización de etiquetas). Para todo  $x \in V^+(p)$  que tenga etiqueta temporal, actualizar etiquetas de acuerdo a:

$$d(x) = \min \{d(x), d(p) + d(p, x)\}$$

si  $d(x)$  se modificó, hacer  $a(x)=p$ . Sea  $x^*$  tal que  $d(x^*) = \min \{d(x) \mid d(x) \text{ es temporal}\}$ . Si  $d(x^*) = \infty$ , terminar. En este caso no existe arborescencia alguna de raíz  $s$ . En otro caso, marcar la etiqueta  $d(x^*)$  como permanente. Sea  $p=x^*$ .

Paso 3. (i) (Si sólo se desea la ruta de  $s$  a  $t$ ). Si  $p = t$ , terminar:  $d(p)$  es la longitud del camino más corto. Si  $p \neq t$ , ir al paso 2. (ii) (Si se desea la arborescencia). Si todos los vértices tienen etiquetas permanentes, terminar; esta es la longitud del camino deseado y el conjunto de arcos  $(a(x), x)$  forman la arborescencia de caminos más cortos. En otro caso ir al paso 2.

## Justificación del algoritmo.

Primeramente obsérvese que el algoritmo termina en un número finito de iteraciones, ya sea en el paso 2 o en el paso 3 puesto que el número de vértices es finito. Se justificará la optimalidad del algoritmo para el caso del problema de arborescencia de rutas más cortas ya que el otro caso está comprendido en éste.

Nótese que si el algoritmo termina en el paso 3, la gráfica generada tendrá  $n-1$  arcos y  $s$  como raíz; por esta razón, dicha gráfica es una arborescencia. Por otro lado se tiene que, por construcción,  $d(x)$  es la longitud del único camino de  $s$  a  $x$  en esta arborescencia. Ahora se probará que la arborescencia generada es de rutas más cortas. Para ello se demostrará, por inducción sobre el número de iteraciones, que las etiquetas permanentes de los vértices, son las longitudes de las rutas más cortas de  $s$  a  $x$ , para todo  $x \in X$ . Esto es claro en la primera iteración. Supóngase que también es válido en la  $k$ -ésima iteración. Sean  $S$  el conjunto de vértices con etiquetas permanentes y  $S'$  el conjunto de vértices con etiquetas temporales en la iteración  $k$ . Al final del paso 2 de la iteración  $k+1$  la etiqueta temporal  $d(x)$ , para  $x \in S'$  es la longitud de una ruta más corta de  $s$  a  $x$  que contiene solamente vértices de  $S$ . En efecto, en cada iteración, sólo se etiqueta permanentemente un vértice; por lo tanto, sólo es necesaria la comparación efectuada en el paso 2. En particular, esto sucede para  $x^*$  (vértice con la mínima etiqueta temporal). Supóngase ahora que la ruta más

corta de la raíz a  $x^*$  no contiene sólo vértices de  $S$ . Sea  $y$  el primer vértice, en el camino más corto de  $s$  a  $x^*$ , que no está en  $S$ . Puesto que los costos de los arcos son no negativos entonces la longitud de la porción del camino de  $s$  a  $x^*$ , que une a  $y$  con  $x^*$ , es no negativa. Sea  $D$  esta longitud. Nótese que la porción del camino de  $s$  a  $x^*$ , que une a  $s$  con  $y$ , es un camino que contiene solamente vértices de  $S$ . Pero  $d(y)$  es la longitud de una ruta más corta que contiene todos sus vértices en  $S$ , luego:

$$d(y) + D < d(x^*)$$

lo que implica

$$d(y) < d(x^*) - D < d(x^*)$$

lo cual constituye una contradicción puesto que  $d(x^*)$  es el mínimo de las etiquetas temporales. Luego, se concluye que la ruta más corta de  $s$  a  $x^*$  contiene sólo vértices de  $S$  y, por lo tanto,  $d(x^*)$  es su longitud. Por lo tanto, la etiqueta permanente de  $x$  es igual a la longitud de la ruta más corta de  $s$  a  $x$  en la iteración  $k + 1$ .

Finalmente, obsérvese que si  $d(x^*)$  es igual a infinito (paso 2) en alguna iteración, entonces existe algún vértice ( $x^*$  y todos los que tengan etiqueta temporal) para el cual no existe ruta alguna desde  $s$ ; puede entonces concluirse que el problema no tiene solución puesto que en este caso  $s$  no es raíz de la red. ■

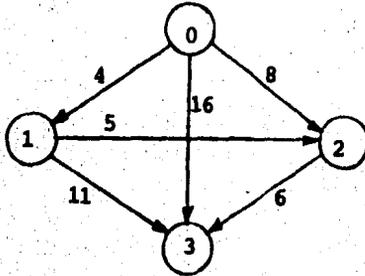
Ejemplo 1. Supóngase que en un aeropuerto se considera la posible adquisición de un equipo que, a causa de ciertas modificaciones, será inútil dentro de tres años. Los costos de utilización y de reventa del equipo usado son conocidos. El equipo puede ser utilizado durante uno, dos o tres años, revenderse al final del periodo de utilización y comprarse uno nuevo para usarse durante el tiempo restante. Los costos totales en los que se incurre (costo de compra más costo de utilización menos precio de venta) están dados en la matriz de costos.

$$C = \begin{matrix} & \begin{matrix} 1 & 2 & 3 \end{matrix} \\ \begin{matrix} 0 \\ 1 \\ 2 \end{matrix} & \begin{bmatrix} 4 & 8 & 16 \\ - & 5 & 11 \\ - & - & 6 \end{bmatrix} \end{matrix}$$

El elemento  $(i, j)$  de la matriz es igual al costo total en millones de pesos en el que se incurre si se compra un equipo al final del año  $i$ , se utiliza hasta el final del año  $j$  y se revende. Se desea encontrar la estrategia más económica de compra y reventa de equipo.

Para resolver este problema considérese la red  $R = [X, A, C]$  donde:  $X = \{0, 1, 2, 3\}$  y cada elemento de  $X$  indica el final de un año;  $(i, j) \in A$  si y sólo si es posible comprar el equipo al final del año  $i$  y revenderlo al final del año  $j$ ; finalmente

$C(i,j)$  es el costo total en el que se incurre por comprar el equipo al final de año  $i$ , usarlo y revenderlo al final del año  $j$ . Una forma esquemática de la red es:



Obsérvese que una ruta de 0 a 3 en la red corresponde a una estrategia posible y viceversa; por otro lado, el costo de una estrategia posible es el mismo que el de la ruta correspondiente. Por ejemplo, la ruta 0,1,3 (de longitud 15) corresponde a comprar al final del año 0, utilizar el equipo durante un año, revenderlo y comprar uno nuevo al final del año 1, utilizarlo durante dos años y revenderlo al final del año 3 (con un costo total de  $4+11=15$  millones). Por otra parte, comprar el equipo al final del año 0, utilizarlo durante tres años y revenderlo al final del año 3 (con un costo total de 16 millones) corresponde a la ruta 0, 3 (de longitud 16).

Para encontrar la estrategia óptima se deberá, entonces, encontrar la ruta más corta entre los vértices 0 y 3 en la red. Para esto se aplicará el algoritmo de Dijkstra considerando  $s = 0$  y  $t = 3$ .

Iteración 1.  $d(0) = 0$  (etiqueta permanente);  $d(x) = \infty$ , para  $x=1,2,3$  (etiquetas temporales).  $p = 0$ .

Actualización de etiquetas:  $\Gamma^+(p) = \{1,2,3\}$

$$\begin{array}{ll} d(1) = \min \{\infty, 4\} = 4 & a(1) = 0 \\ d(2) = \min \{\infty, 8\} = 8 & a(2) = 0 \\ d(3) = \min \{\infty, 16\} = 16 & a(3) = 0 \end{array}$$

De donde  $x^* = 1$  (vértice con mínima etiqueta temporal). Se marca  $d(1)$  como permanente;  $p = 1$ . ( $p \neq t = 3$ ).

Iteración 2. Actualización de etiquetas:  $\Gamma^+(p) = \{2, 3\}$

$$\begin{array}{ll} d(2) = \min \{8, 4 + 5\} = 8 & a(2) \text{ no cambia} \\ d(3) = \min \{16, 4+11\} = 15 & a(3) = 1 \end{array}$$

De donde  $x^* = 2$  (vértice con mínima etiqueta temporal). Se marca  $d(2)$  como permanente;  $p = 2$  ( $p \neq t$ ).

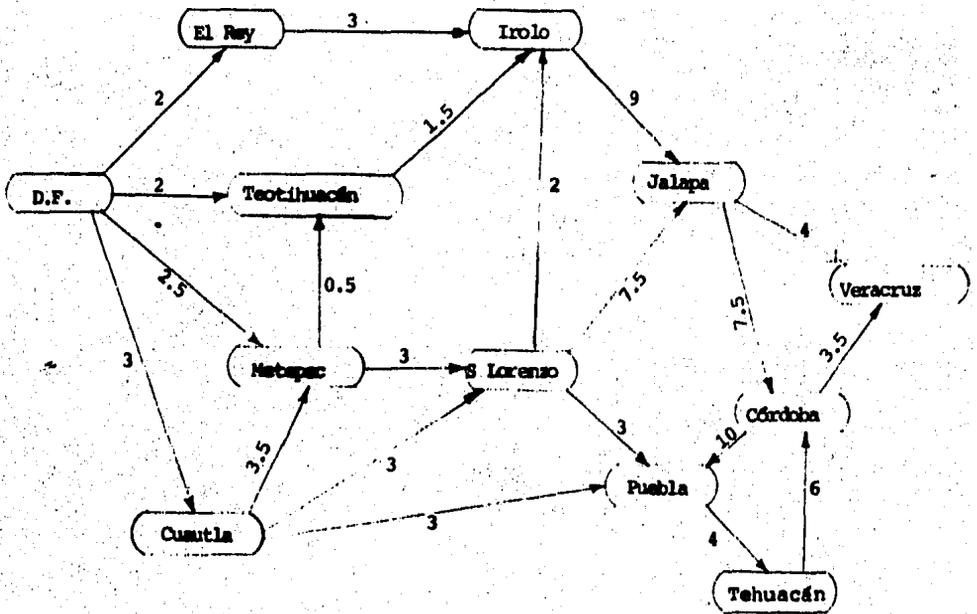
Iteración 3. Actualización de etiquetas:  $\Gamma^+(p) = \{3\}$

$$d(3) = \min \{15, 8 + 6\} = 14 \quad a(3) = 2$$

De donde  $x^* = 3$  (vértice con mínima etiqueta temporal). Se marca  $d(3)$  como permanente;  $p = 3 = t$ . Se termina.

Para recuperar la ruta más corta de 0 a 3, de longitud  $d(3) = 14$ , se utilizan las etiquetas  $a(x)$ . La ruta deseada, en sentido inverso, es: 3,  $a(3) = 2$ ,  $a(2) = 0$ . Luego la estrategia óptima para el problema del aeropuerto es comprar el equipo al final del año 0, utilizarlo durante dos años, revenderlo y comprar uno nuevo al final del año 2, utilizar este último durante un año y venderlo al final del año 3 con un costo total de 14 millones de pesos.

**Ejemplo 2.** Considérese el siguiente mapa de los Ferrocarriles Nacionales de México, donde las vías están representadas por arcos y el número asociado al arco  $(i,j)$  representa el tiempo, en horas, que tarda un ferrocarril en recorrer el tramo de vía  $(i,j)$ .



Supóngase que se desea encontrar las rutas más rápidas para los ferrocarriles que prestan servicio desde el D.F. hasta cada una de las ciudades del mapa.

En términos de gráficas se desea determinar la arborescencia de rutas más cortas de raíz D.F. Para resolver este problema se utilizará el algoritmo de Dijkstra puesto que los costos de los arcos son no negativos.

Iteración 1.  $d(D.F.) = 0$  (etiqueta permanente);  $d(x) = \infty$  para todo  $x \in X$ ,  $x \neq D.F.$  (etiquetas temporales).  $p = D.F.$

Actualización de etiquetas:  $\Gamma^+(p) = \{\text{El Rey, Teotihuacán, Metepec, Cuautla}\}$ :

$d(\text{El Rey})$	$= \min \{\infty, 2\} = 2$	;	$a(\text{El Rey})$	$= D.F.$
$d(\text{Teotihuacán})$	$= \min \{\infty, 2\} = 2$	;	$a(\text{Teotihuacán})$	$= D.F.$
$d(\text{Metepec})$	$= \min \{\infty, 2.5\} = 2.5$	;	$a(\text{Metepec})$	$= D.F.$
$d(\text{Cuautla})$	$= \min \{\infty, 3\} = 3$	;	$a(\text{Cuautla})$	$= D.F.$

De donde  $x^* = \text{El Rey}$ , puesto que éste es el vértice con mínima etiqueta temporal (los empates se rompen arbitrariamente).

Se marca  $d(\text{El Rey})$  como permanente;  $p = \text{El Rey}$ . (Existen aún etiquetas temporales).

Iteración 2. Actualización de etiquetas:  $\Gamma^+(p) = \{\text{Irolo}\}$

$d(\text{Irolo}) = \min \{\infty, 2+3\} = 5$  ;  $a(\text{Irolo}) = \text{El Rey}$

De donde  $x^* = \text{Teotihuacán}$  (vértice con mínima etiqueta temporal). Se marca  $d(\text{Teotihuacán})$  como permanente;  $p = \text{Teotihuacán}$ . (Aún no están etiquetados todos los vértices permanentemente).

Iteración 3. Actualización de etiquetas:  $\Gamma^+(p) = \{\text{Irolo}\}$

$$d(\text{Irolo}) = \min \{5, 2+1.5\} = 3.5; \quad a(\text{Irolo}) = \text{Teotihuacán}$$

De donde  $x^* = \text{Metepec}$ . Se marca la etiqueta de este vértice como permanente;  $p = \text{Metepec}$ . (Existen aún etiquetas temporales).

Iteración 4. Actualización de etiquetas:  $\Gamma^+(p) = \{\text{Teotihuacán, San Lorenzo}\}$  (Teotihuacán ya tiene etiqueta permanente).

$$d(\text{San Lorenzo}) = \min \{\infty, 2.5+3\} = 5.5; \quad a(\text{San Lorenzo}) = \text{Metepec}$$

$x^* = \text{Cuautla}$ . Se marca la etiqueta de Cuautla como permanente;  $p = \text{Cuautla}$ . (Existen aún etiquetas temporales).

Iteración 5. Actualización de etiquetas:  $\Gamma^+(p) = \{\text{Metepec, San Lorenzo, Puebla}\}$

$$d(\text{San Lorenzo}) = \min \{5.5, 3+3\} = 5.5; \quad a(\text{San Lorenzo}) \text{ no se modifica}$$

$$d(\text{Puebla}) = \min \{\infty, 3+3\} = 6; \quad a(\text{Puebla}) = \text{Cuautla}$$

$x^* = \text{Irolo}$ . Se marca la etiqueta de Irolo como permanente;  $p = \text{Irolo}$ . (existen aún etiquetas temporales)

Iteración 6. Actualización de etiquetas:  $\Gamma^+(p) = \{\text{Jalapa}\}$ .

$$d(\text{Jalapa}) = \min \{\infty, 3.5+9\} = 12.5; \quad a(\text{Jalapa}) = \text{Irolo}$$

$x^* = \text{San Lorenzo}$ . Se marca la etiqueta de San Lorenzo como permanente;  $p = \text{San Lorenzo}$ . (Existen etiquetas temporales).

Iteración 7. Actualización de etiquetas:  $\Gamma^+(p) = \{Jalapa, Puebla, Irolo\}$ .

$d(Jalapa) = \min \{12.5, 5.5 + 7.5\} = 12.5$  ;  $a(Jalapa)$  no se modifica

$d(Puebla) = \min \{6, 5.5 + 3\} = 6$  ;  $a(Puebla)$  no se modifica

$x^* = Puebla$ . Se marca la etiqueta de Puebla como permanente;

$p = Puebla$ . (Existen etiquetas temporales).

Iteración 8. Actualización de etiquetas:  $\Gamma^+(p) = \{Tehuacán\}$

$d(Tehuacán) = \min \{\infty, 6 + 4\} = 10$  ;  $a(Tehuacán) = Puebla$

$x^* = Tehuacán$ . Se marca la etiqueta de este vértice como permanente;

$p = Tehuacán$  (Existen etiquetas temporales).

Iteración 9. Actualización de etiquetas:  $\Gamma^+(p) = \{Córdoba\}$

$d(Córdoba) = \min \{\infty, 10 + 6\} = 16$  ;  $a(Córdoba) = Tehuacán$

$x^* = Jalapa$ . Se marca la etiqueta de Jalapa como permanente;

$p = Jalapa$ . (Existen etiquetas temporales).

Iteración 10. Actualización de etiquetas:  $\Gamma^+(p) = \{Veracruz, Córdoba\}$

$d(Veracruz) = \min \{\infty, 12.5 + 4\} = 16.5$  ;  $a(Veracruz) = Jalapa$

$d(Córdoba) = \min \{16, 12.5 + 7.5\} = 16$  ; no se modifica  $a(Córdoba)$

$x^* = Córdoba$ . Se marca la etiqueta de Córdoba como permanente;

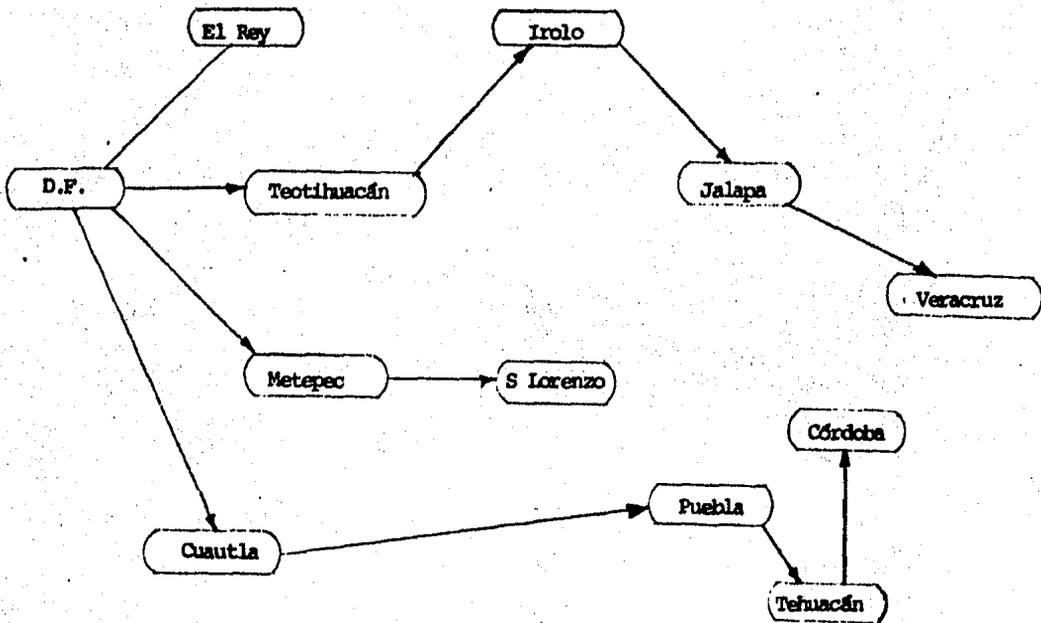
$p = Córdoba$ . (Existe aún una etiqueta temporal).

Iteración 11. Actualización de etiquetas:  $\Gamma^+(p) = \{Puebla, Veracruz\}$ ; (Puebla ya tiene etiqueta permanente).

$d(\text{Veracruz}) = \min \{16.5, 16 + 3.5\} = 16.5$      $a(\text{Ver.})$  no se modifica  
 $x^* = \text{Veracruz}$  (se marca permanente).

Todos los vértices tienen etiqueta permanente.    Alto.

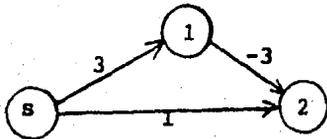
La arborescencia de rutas más cortas de raíz D.F. está formada por el conjunto de arcos  $(x, a(x))$ . Esta arborescencia es:



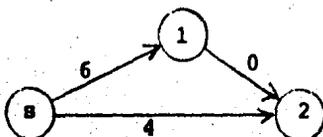
El único camino entre el D.F. y  $x$ , para todo  $x \in X$ , en esta arborescencia es la ruta más rápida a seguir por el ferrocarril que proporciona servicio entre las ciudades D.F. y  $x$ , y su longitud es el número asociado al vértice  $x$ . (Etiqueta permanente).

### Caso general.

Para resolver el problema de la arborescencia de rutas más cortas en una red donde se admiten arcos con costos negativos es necesario un método distinto del algoritmo de Dijkstra puesto que éste puede conducir a un error. Para ejemplificar esto considérese la siguiente red y supóngase que se desea encontrar la arborescencia de rutas más cortas de raíz  $s$ :

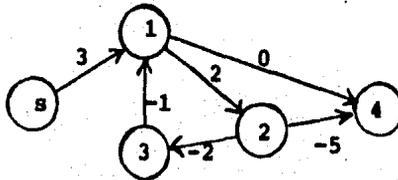


Puede verificarse fácilmente que si se aplica el algoritmo de Dijkstra a la red anterior, la ruta "más corta" que se obtiene entre los vértices  $s$  y  $2$  es el arco  $(s,2)$  de longitud  $1$ . Sin embargo la ruta  $s,1,3$  es más corta que la anterior puesto que tiene longitud  $0$ . El algoritmo de Dijkstra no puede, por tanto, aplicarse en estos casos. Un primer método que podría ocurrirse para resolver el problema es sumar el recíproco del costo más negativo de la red a los costos de todos los arcos y aplicar entonces el algoritmo de Dijkstra. Sin embargo, esto tampoco resultaría puesto que las longitudes de las rutas serían afectadas de manera distinta ya que pueden tener distinto número de arcos. Por ejemplo, si en la red anterior se suma  $3$  a los costos de todos los arcos se obtiene lo siguiente:

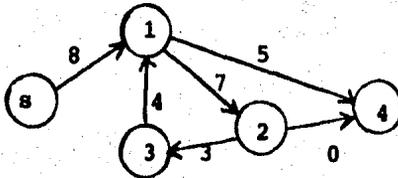


Si ahora se aplica el algoritmo de Dijkstra se obtiene de nuevo como ruta "más corta" de s a 2 el arco (s,2) que, como se observó anteriormente, no constituye una ruta más corta de s a 2 en la red original.

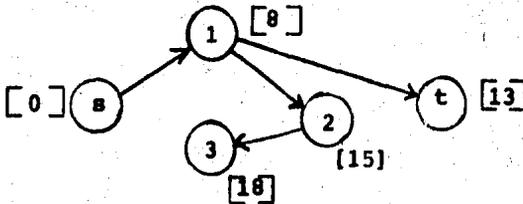
Puede suceder también que, si se utiliza este método, se obtenga una "solución" aún cuando ésta no exista. Considérese, por ejemplo la siguiente red:



Si se suma 5 a todos los costos se obtiene:



Al aplicar el algoritmo Dijkstra a esta red se obtiene la siguiente arborescencia de rutas "más cortas" de raíz s:



Sin embargo, en la red original, existe un circuito negativo, 1,2,3,1, y por lo tanto la arborescencia de rutas más cortas no existe.

Por las razones expuestas anteriormente es necesario presentar un método de solución para el caso de redes que admiten cualquier costo. Este método, que es una generalización del algoritmo de Dijkstra, consiste, a grandes rasgos, en lo siguiente:

Primeramente se aplica el algoritmo de Dijkstra a la red  $R = [X, A, d]$  para obtener una arborescencia cualquiera. Después se "prueba" la optimalidad de esta arborescencia; en el caso que la arborescencia no sea óptima (es decir, que no sea la de rutas más cortas) se procede a cambiar ésta por otra mejor y se repite el procedimiento hasta obtener la optimalidad. Una arborescencia no es de rutas más cortas si existe algún vértice para el cual el único camino de la raíz a él no es el más corto; por esto una manera natural de probar la optimalidad consiste en verificar si al agregar un arco  $a = (x, y)$  a la arborescencia, el nuevo camino formado entre la raíz y el extremo final de este arco,  $y$ , es más corto que el correspondiente en la arborescencia. Si este es el caso, es claro que al intercambiar este arco por el único arco que está en la arborescencia que tiene como extremo final al vértice  $y$ , se obtiene una arborescencia mejor. En caso contrario la arborescencia es óptima.

Por otro lado, puesto que la red admite cualquier costo, podría no existir la solución óptima si la red contiene circuitos negativos; es importante entonces contar con una herramienta que permita la detección de tales circuitos. Con este objeto se deberá observar si el extremo final de un arco que mejore una ru-

ta es la raíz, ya que en este caso se habrá encontrado un camino de  $s$  a  $s$  de longitud negativa (es decir, un circuito negativo), o si al agregar un arco y eliminar el correspondiente en la arborescencia resulta que el último no formaba parte del ciclo formado; en este caso la gráfica resultante no es una arborescencia ya que contiene un ciclo. Este ciclo es además un circuito puesto que se construye una gráfica en la cual el grado interior de todo vértice es uno. Este circuito es negativo, para probar esta afirmación considérese lo siguiente:

La operación de agregar un arco adecuado  $a=(x,y)$  y eliminar el correspondiente  $(z,y)$  conduce a mejorar no sólo la ruta de la raíz al vértice  $y$ , sino todas las rutas que contengan como vértice intermedio a  $y$  (puesto que todas ellas se modifican). De esta manera, si en el resultado se obtiene un circuito se mejoran las rutas de todos los vértices de tal circuito a través de él mismo. Por esto puede concluirse que éste es negativo.

Con esto como motivación se procede ahora a enunciar el algoritmo.

## ALGORITMO GENERAL

Propósito. Obtener la arborescencia de rutas más cortas, de raíz  $s$ , en una red  $R = [X, A, d]$  que admite cualquier costo.

### Descripción

Paso 1. (Inicio). Encontrar una arborescencia cualquiera de raíz  $s$  en la red. Puede obtenerse ésta a partir del algoritmo de Dijkstra.

Paso 2. (Optimalidad). Calcular, para todo arco  $(i, j)$  que no esté en la arborescencia, el escalar  $d(i) + d(i, j)$ . Si se cumple que  $d(i) + d(i, j) \geq d(j)$  para todo  $(i, j) \in A$ , terminar. Se obtuvo la arborescencia de caminos más cortos. De otra manera  $d(i) + d(i, j) < d(j)$  para algún  $(i, j) \in A$ , ir al paso 3.

Paso 3. (Cambio de arborescencia). Si  $j = s$  terminar ya que existe un circuito negativo. En otro caso sea  $(k, j) \in A$  el único arco, incidente a  $j$ , que está en la arborescencia. Reemplazar dicho arco por  $(i, j)$ . Si la gráfica resultante es una arborescencia actualizar las etiquetas de todos los vértices  $x$ , para los cuales la ruta de  $s$  a  $x$  contenga a  $j$  como vértice intermedio o final de acuerdo con:

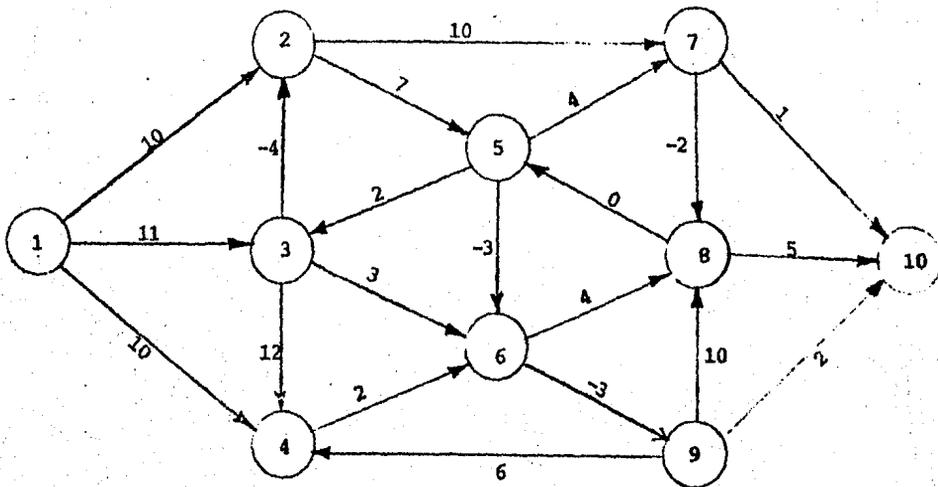
$$d(x) = d(x) - d'$$

donde  $d' = d(j) - d(i) - d(i, j)$ . Ir al paso 2. En otro caso, terminar; existe un circuito negativo y por lo tanto no hay solución.

### Justificación del algoritmo.

Obsérvese que el algoritmo termina con una arborescencia si  $d(i) + d(i,j) \geq d(j)$ , para todo arco  $(i,j) \in A$ . Supóngase que  $d(j)$ , al final del algoritmo, no es la longitud de una ruta más corta de  $s$  a  $j$ , para algún  $j \in X$ . Si existe más de un vértice con esta característica, considérese aquél tal que la ruta de  $s$  a  $j$  (en la arborescencia) tenga el menor número de arcos. Por construcción,  $d(j)$  es la longitud de un camino de  $s$  a  $j$ . Sea  $P$  un camino más corto de  $s$  a  $j$ , entonces  $d(P) < d(j)$ . Sea  $x$  el predecesor de  $j$  en el camino  $P$ . Entonces  $d(x)$  es la longitud de un camino más corto de  $s$  a  $x$  y por lo tanto  $d(P) = d(x) + d(x,j) < d(j)$ , lo cual constituye una contradicción. Luego  $d(j)$  es igual a la longitud de una ruta más corta de  $s$  a  $j$  y por lo tanto la arborescencia generada es de rutas más cortas de raíz  $s$ . ■

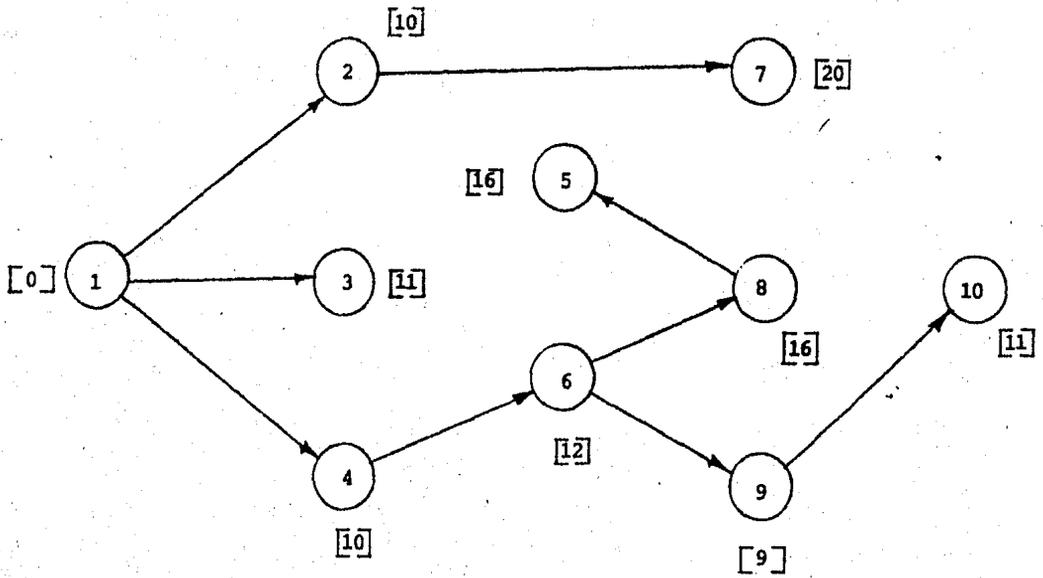
Ejemplo 3. Considérese la siguiente red:



Encuéntrese la arborescencia de rutas más cortas de raíz 1 en esta red.

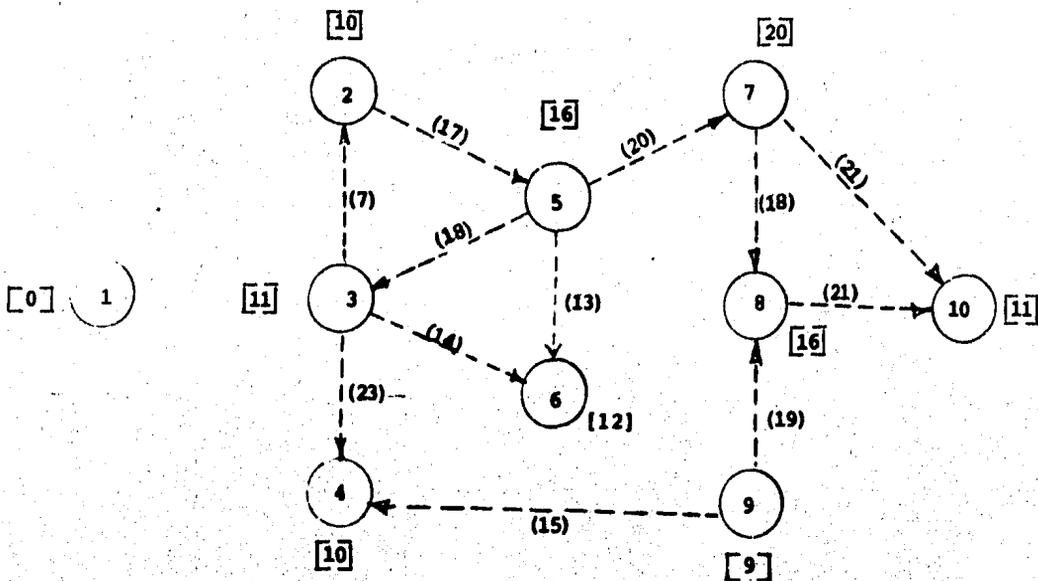
Dado que la red contiene arcos con costos negativos, se utilizará el algoritmo general cuyo desarrollo es:

Iteración 1. Puede verificarse fácilmente que, al aplicar el algoritmo de Dijkstra, se obtiene la siguiente arborescencia inicial de raíz 1.

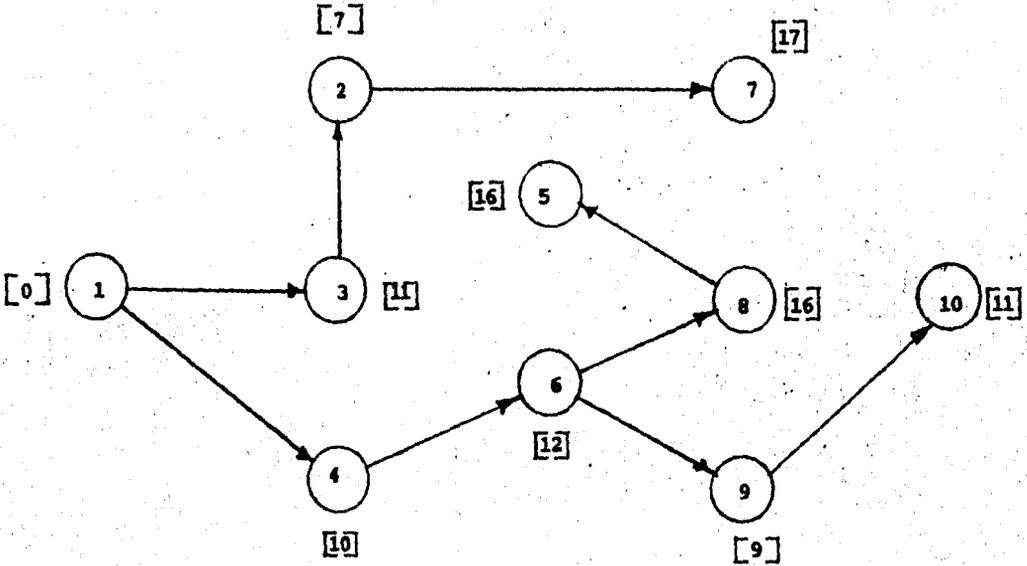


Donde el número asociado a cada vértice  $x$ , denotado  $d(x)$ , es la longitud de la única ruta, en esta arborescencia, del vértice 1 a  $x$ .

Para verificar si esta arborescencia es de rutas más cortas se calcula  $d(i) + d(i,j)$ , para todo arco  $(i,j)$  que no está en la arborescencia, y se compara con  $d(j)$ . En la siguiente figura se presentan estos arcos:



Donde el número asociado a cada arco  $(i, j)$  es  $d(i) + d(i, j)$ .  
 Obsérvese que, para el arco  $(3, 2)$ ,  $d(3) + d(3, 2) = 7 < d(2) = 10$ , por lo cual es conveniente agregar el arco  $(3, 2)$  a la arborescencia original. El único arco en la arborescencia, con extremo final 2, es  $(1, 2)$ . Por lo tanto se reemplaza este arco por  $(3, 2)$ , obteniéndose la siguiente arborescencia:



donde se actualizó:

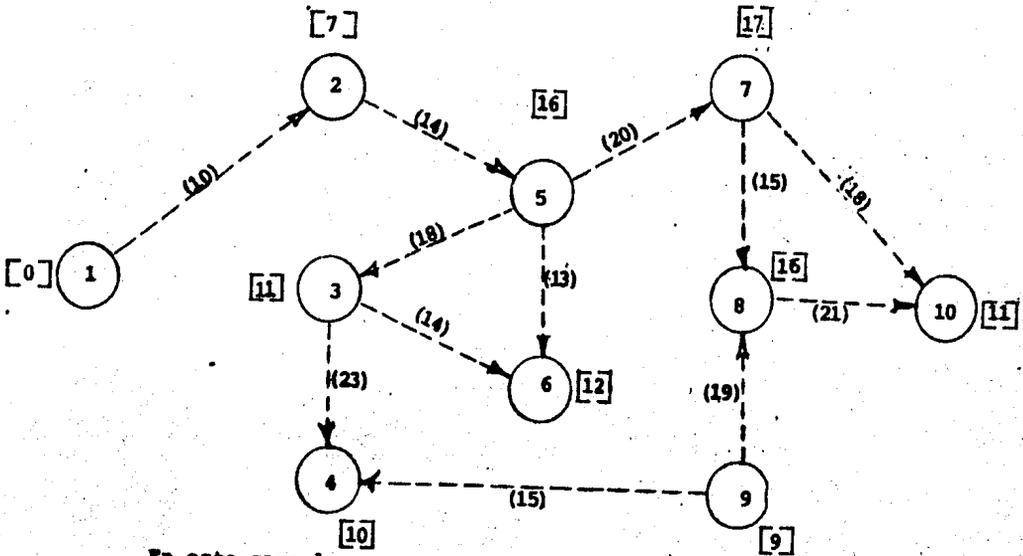
$$d' = d(2) - d(3) - d(3,2) = 10 - 11 + 4 = 3,$$

$$d(2) = d(2) - d' = 10 - 3 = 7$$

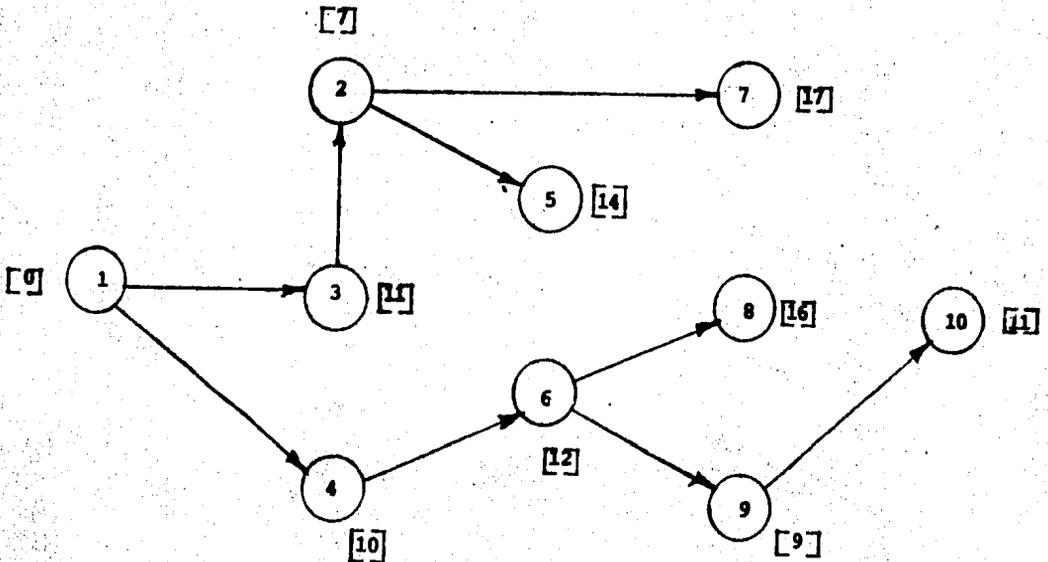
$$d(7) = d(7) - d' = 20 - 3 = 17$$

puesto que las rutas de 1 a 2 y de 1 a 7 pasan por el vértice 2.

Iteración 2. Ahora se verificará la optimalidad de esta nueva arborescencia. De nuevo, el número asociado a cada arco en la siguiente red es  $d(i) + d(i,j)$ :



En este caso los arcos que "rompen" la optimalidad de la arborescencia son (2,5) y (7,8). Elíjase (2,5) para agregar a la arborescencia. El único arco con extremo final 5 en la arborescencia es (8,5). Reemplazando éste último por (2,5) se obtiene:

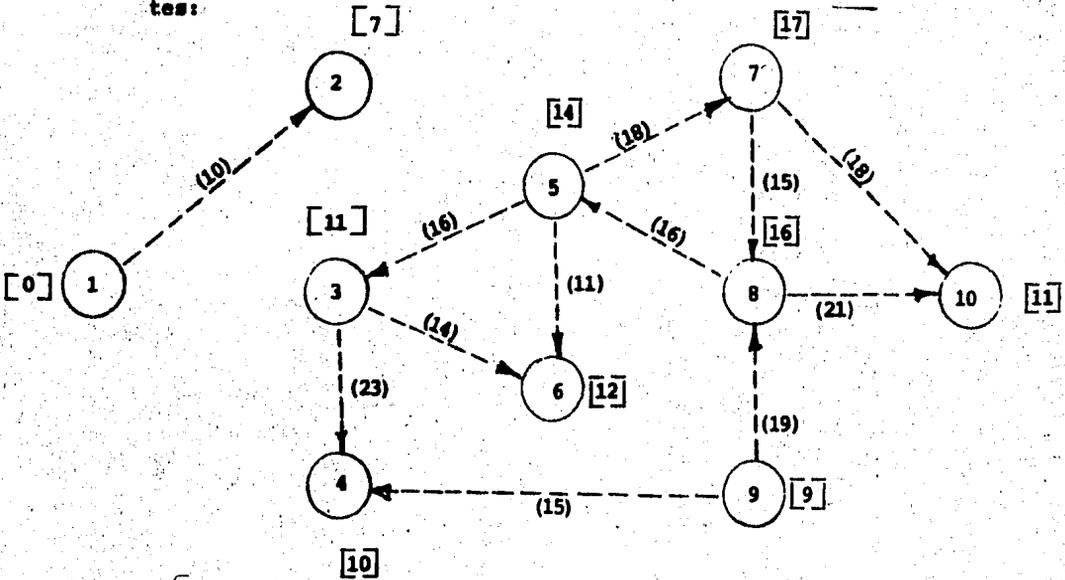


donde sólo fué necesario actualizar la etiqueta del vértice 5 de acuerdo a:

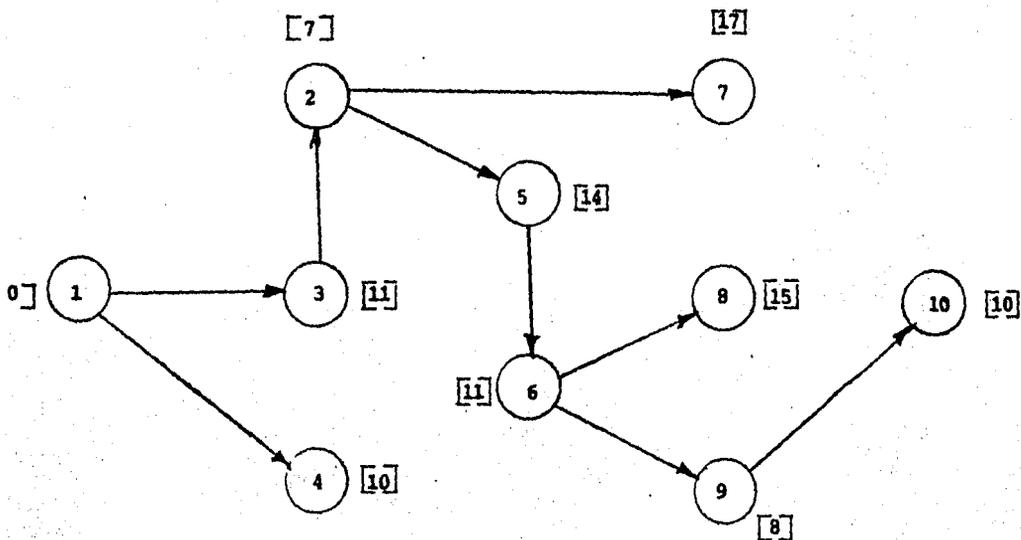
$$d' = d(5) - d(2) - d(2,5) = 16 - 7 - 7 = 2$$

$$d(5) = d(5) - d' = 16 - 2 = 14$$

Iteración 3. Se verifica ahora la optimalidad de esta última arborescencia. A continuación se presentan los arcos que no están en la arborescencia junto con los cálculos pertinentes:



En esta iteración, los arcos que permiten garantizar que esta última arborescencia no es óptima son: (5,6) y (7,8). El único arco con extremo final 6 en la arborescencia es (4,6). Intercambiando este arco por (5,6) se obtiene la siguiente arborescencia:



donde se actualizaron las etiquetas de los vértices 6,8,9 y 10 mediante:

$$d' = d(6) - d(5) - d(5,6) = 12 - 14 + 3 = 1$$

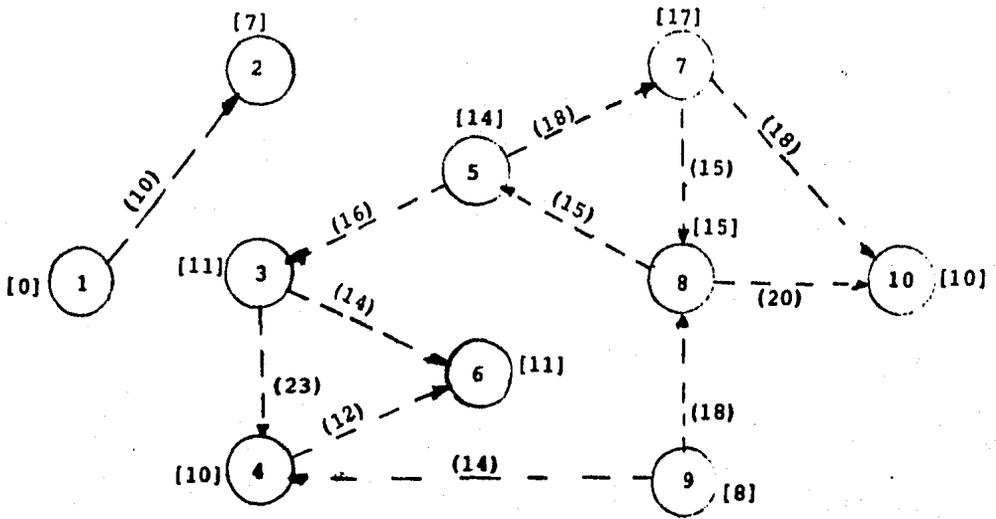
$$d(6) = d(6) - d' = 12 - 1 = 11$$

$$d(8) = d(8) - d' = 16 - 1 = 15$$

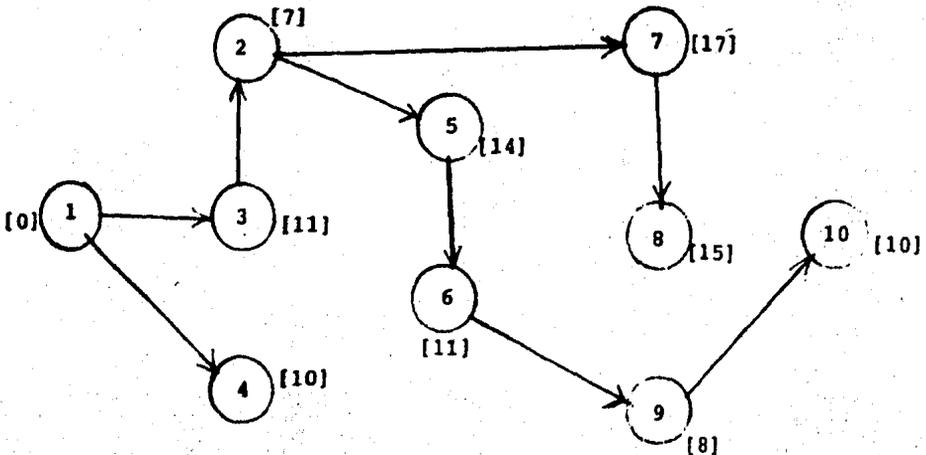
$$d(9) = d(9) - d' = 9 - 1 = 8$$

$$d(10) = d(10) - d' = 11 - 1 = 10$$

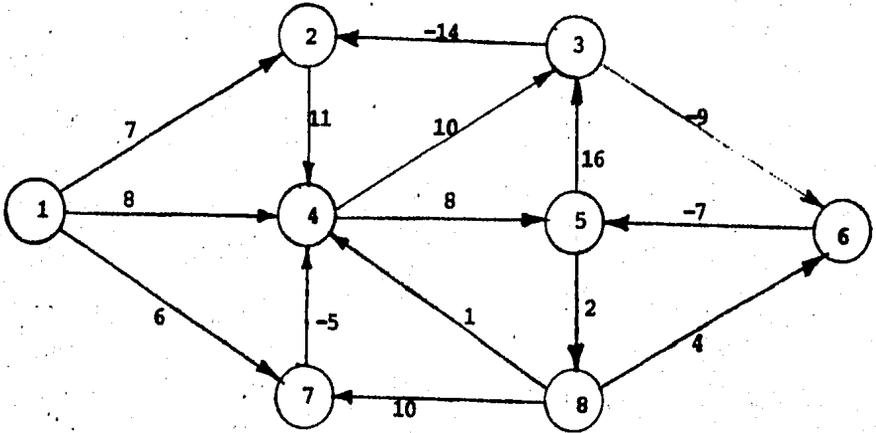
**Iteración 4.** Se comprobará la optimalidad de la arborescencia actual:



En esta figura se cumple, para todo arco,  $d(i)+d(i,j) \geq d(j)$ , por lo cual se concluye que la última arborescencia es de rutas más cortas. Por otro lado, puede notarse que la solución óptima no es única ya que  $d(7)+d(7,8)=17-2=15=d(8)$ . Si se reemplaza el arco (6,8), que es el único con extremo final 8 en la arborescencia, por el arco (7,8), se obtiene otra arborescencia de rutas más cortas:



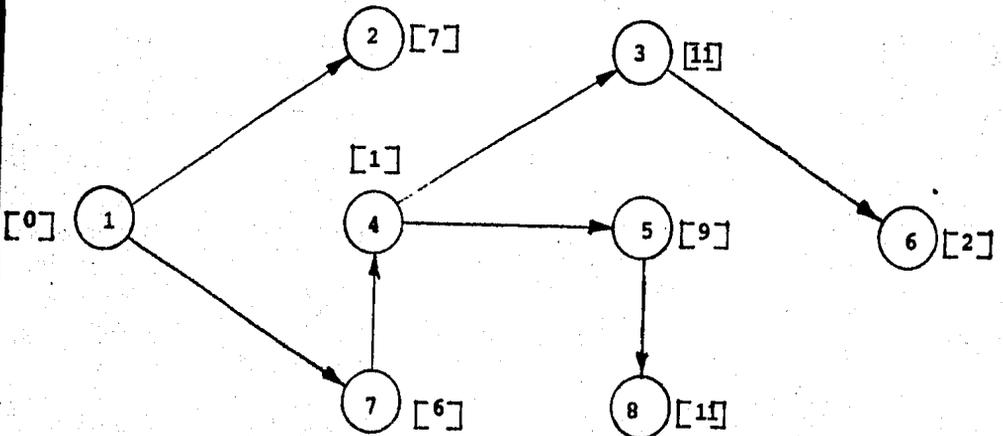
**Ejemplo 4.** Considérese la siguiente red:



Encuéntrese la arborescencia de rutas más cortas de raíz 1.

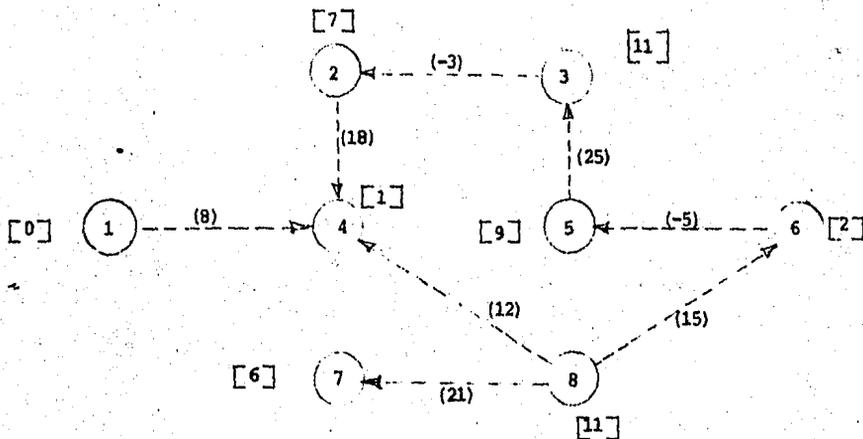
Dado que existen arcos con costos negativos se utilizará el algoritmo general cuyo desarrollo es como sigue:

Iteración 1. Aplicando el algoritmo de Dijkstra se obtiene la siguiente arborescencia inicial:

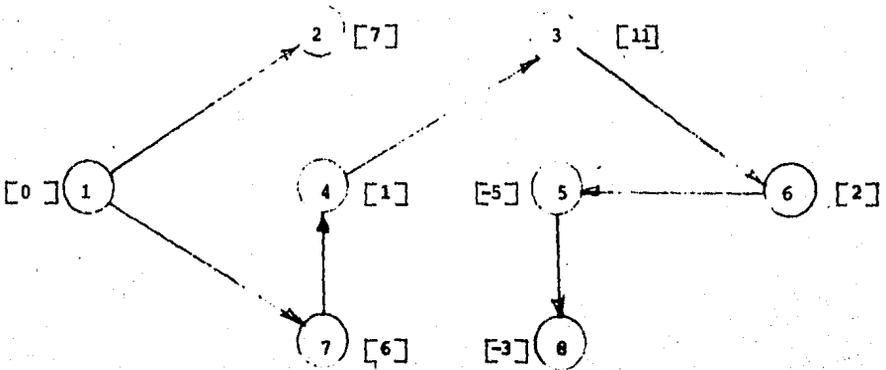


donde el número asociado a cada vértice  $x$ , denotado  $d(x)$ , es la longitud de la única ruta, del vértice 1 a  $x$ , en esta arborescencia.

A continuación se calculará  $d(i) + d(i,j)$ , para todo arco  $(i,j)$  que no esté en la arborescencia, con objeto de verificar la optimalidad. En la siguiente figura se asocia a cada arco  $(i,j)$  el número  $d(i) + d(i,j)$ :



En esta figura existen dos arcos que permiten afirmar que esta arborescencia no es de rutas más cortas. En efecto, los arcos  $(3,2)$  y  $(6,5)$  son tales que  $d(i) + d(i,j) < d(j)$ . El único arco, en la arborescencia, que tiene extremo final 5 es  $(4,5)$ . Reemplazando este arco por  $(6,5)$ , se obtiene:



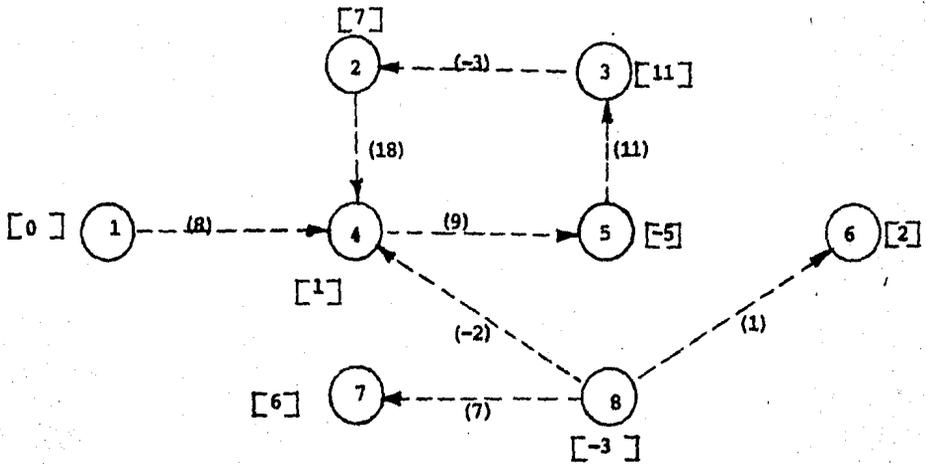
donde se actualizaron las etiquetas de los vértices 5 y 8 (ya que las rutas de 1 a 5 y de 1 a 8 contienen al vértice 5) de acuerdo con:

$$d' = d(5) - d(6) - d(6,5) = 9 - 2 + 7 = 14$$

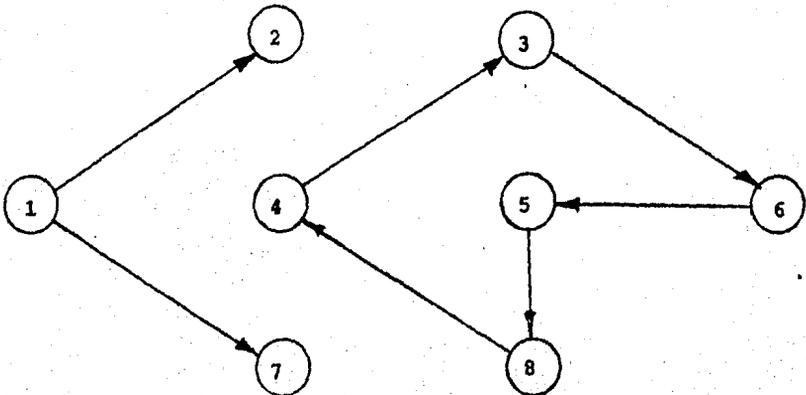
$$d(5) = d(5) - d' = 9 - 14 = -5$$

$$d(8) = d(8) - d' = 11 - 14 = -3$$

Iteración 2. Se verificará ahora la optimalidad de esta nueva arborescencia. Asociado a cada arco  $(i,j)$  de la siguiente red, que son los que no están en la arborescencia, se tiene el número  $d(i) + d(i,j)$ :



Los arcos que "rompen" la optimalidad de la última arborescencia son (3,2), (8,4) y (8,6). El único arco con extremo final 4 en la arborescencia es (7,4). Intercambiando este último arco por (8,4) se obtiene:



Esta gráfica no es una arborescencia puesto que contiene al circuito 4,3,6,5,8,4 y por lo tanto se concluye que el problema no tiene solución. En efecto, obsérvese que dicho circuito tiene longitud  $10 - 9 - 7 + 2 + 1 = -3 < 0$  y por lo tanto no existe arborescencia de rutas más cortas (problema no acotado).

b. RUTA MAS CORTA ENTRE TODO PAR DE VERTICES.

Como se habfa mencionado anteriormente, una manera de resolver el problema de las rutas más cortas entre todo par de vértices en una red  $R = [X, A, d]$  es encontrar la arborescencia de rutas más cortas de raíz  $x$  para todo  $x \in X$ . Sin embargo, existen procedimientos más eficientes como el que se expodrá en esta parte del capítulo. El procedimiento que se discute fué desarrollado por R. W. Floyd (1962) y es aplicable a redes que admiten cualquier costo en sus arcos. En dicho algoritmo se supondrá una numeración de los vértices de la red  $1, 2, \dots, n$  y se utilizará una matriz  $C$ , de orden  $n \times n$ , para calcular las longitudes de las rutas más cortas entre cada par de vértices; al terminar de aplicar el algoritmo, la longitud de la ruta más corta entre los vértices  $i$  y  $j$  es dada por el elemento  $(i, j)$  de  $C$ .

En el algoritmo de Floyd, en la  $k$ -ésima iteración se calcula la longitud de la ruta más corta entre  $i$  y  $j$  que puede admitir a los primeros  $k$  vértices, o a algunos de ellos, como vértices intermedios; este número se almacena en la entrada  $(i, j)$  de la matriz  $C$ . Al inicio se asigna el costo del arco  $(i, j)$ , al elemento  $(i, j)$  de la matriz  $C$ , si  $i \neq j$ ; si dicho arco no existe entonces se asigna  $\infty$ . Los valores de la diagonal serán 0. Con ésto quedan calculadas las longitudes de las rutas más cortas, entre todo par de vértices  $i$  y  $j$ , que no contengan ningún vértice como vértice intermedio.

Al principio de la  $k$ -ésima iteración, la entrada  $(i, j)$  de  $C$  es igual a la longitud de la ruta más corta, entre  $i$  y  $j$ , que contiene a los primeros  $k-1$  vértices, o a algunos de ellos, como vértices intermedios. Durante esta iteración se compara la longitud de esta ruta con la de aquella formada por la unión de las rutas más cortas, que contienen a los primeros  $k-1$  vértices como vértices intermedios, entre  $i$  y  $k$  y  $k$  y  $j$ ; de esta manera se obtiene la ruta más corta, entre  $i$  y  $j$ , que contiene a los primeros  $k$  vértices, o a algunos de ellos, como vértices intermedios. Procediendo de este modo se tendrá que, al final de la  $n$ -ésima iteración, la entrada  $(i, j)$  de  $C$  es la longitud de la ruta más corta, entre  $i$  y  $j$ , que contiene a los primeros  $n$  vértices como vértices intermedios o a algunos de ellos; es decir, se habrá calculado la longitud de la ruta más corta entre  $i$  y  $j$ .

Debe observarse que si, al finalizar de aplicar el algoritmo, alguna entrada de  $C$  es igual a  $\infty$ , esto querrá decir que no existe ruta alguna entre los vértices correspondientes. Por otro lado si algún elemento de la diagonal de  $C$ , supóngase el  $(i, i)$ , es menor que cero en alguna iteración, se habrá encontrado una ruta de  $i$  a  $i$  de longitud negativa (es decir, un circuito negativo). Luego, en este caso, el problema no tiene solución. Por esto mencionado últimamente, este algoritmo será de gran utilidad en problemas de detección de circuitos negativos. A continuación se describe detalladamente el algoritmo de Floyd.

## ALGORITMO DE FLOYD

Propósito. Obtener las rutas más cortas entre todo par de vértices en una red  $R = [X, A, d]$  con  $n$  vértices

### Descripción

Paso 1. Constrúyase la matriz  $C$ , de  $n \times n$ , de elementos  $C_{ij}$ ,

$$C_{ij} = \begin{cases} 0 & \text{si } i = j \\ \infty & \text{si } (i, j) \notin A \\ d(i, j) & \text{si } (i, j) \in A \end{cases}$$

Hágase  $k = 0$ .

Paso 2. Hacer  $k = k + 1$ . Para todo  $i \neq k$  tal que  $C_{ik} \neq \infty$  y para todo  $j \neq k$  tal que  $C_{kj} \neq \infty$ , hacer:

$$C_{ij} = \min \{ C_{ij}, C_{ik} + C_{kj} \}$$

Paso 3. (i) Si  $C_{ii} < 0$  para alguna  $i$ , terminar. En este caso existe un circuito negativo que contiene al vértice  $i$  y por lo tanto no hay solución.

(ii) Si  $C_{ii} \geq 0$ , para toda  $i$ , y  $k = n$ , terminar;  
 $C_{ij}$  es la longitud del camino más corto de  $i$  a  $j$ .

(iii) Si  $C_{ii} \geq 0$ , para toda  $i$ , y  $k < n$ , ir al paso 2.

### Recuperación de las rutas

Para recuperar las rutas más cortas puede construirse una matriz  $A$  de dimensión  $n \times n$ ; el elemento  $a_{ij}$  de esta matriz será el predecesor del vértice  $j$  en la ruta de  $i$  a  $j$  encontrada en cada iteración. Dada la definición de  $A$ , sus entradas se inicializarán  $a_{ij} = i$ , para todo par de vértices  $i, j \in X$ .  $A$  será modificada en el paso 2 de la  $k$ -ésima iteración de acuerdo con:

$$a_{ij} = \begin{cases} a_{kj} & \text{si } C_{ik} + C_{kj} < C_{ij} \\ \text{no cambia,} & \text{si } C_{ij} \leq C_{ik} + C_{kj} \end{cases}$$

### Justificación del algoritmo

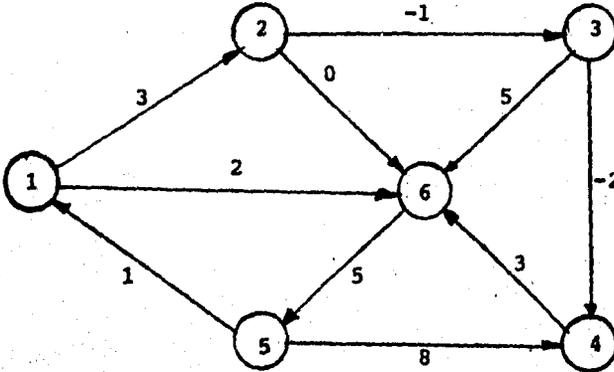
El algoritmo de Floyd termina en exactamente  $n$  iteraciones, donde  $n$  es el número de vértices de la red, a menos que termine en el inciso (i) del paso 3; en éste último caso, se terminará en menos de  $n$  iteraciones. Se mostrará la optimalidad por inducción sobre el número de iteraciones.

Al principio de la iteración 1 (Paso 1) la entrada  $(i, j)$  de la matriz representa la longitud de la ruta más corta, que no contiene ningún vértice intermedio, entre los vértices  $i$  y  $j$ . Durante el paso 2 de esta iteración se realiza una comparación entre dicha longitud ( $C_{ij}$ ) y la de aquella ruta

formada por la unión de la ruta entre  $i$  y  $l$  y la ruta entre  $l$  y  $j$  ( $C_{il} + C_{lj}$ ); de este modo se obtiene la longitud de la ruta más corta, entre los vértices  $i$  y  $j$ , que no contiene ningún vértice intermedio o que contiene al vértice  $l$  como intermedio. Supóngase que al final de la iteración  $k-1$ ,  $C_{ij}$  representa la longitud de la ruta más corta, entre  $i$  y  $j$ , que contiene a los primeros  $k-1$  vértices como vértices intermedios, o a algunos de ellos. Durante el paso 2, de la iteración  $k$ , se realiza la comparación entre esta última longitud ( $C_{ij}$ ) y la de aquella ruta formada por la unión de las rutas más cortas, que admiten a los primeros  $k-1$  vértices como vértices intermedios, entre  $i$  y  $k$  y entre  $k$  y  $j$  ( $C_{ik} + C_{kj}$ ); entonces, al final de la  $k$ -ésima iteración,  $C_{ij}$  es igual a la longitud de la ruta más corta, entre  $i$  y  $j$ , que contiene a los primeros  $k$  vértices como intermedios o a algunos de ellos.

De lo anterior se concluye que, al final de la iteración  $n$ ,  $C_{ij}$  es la longitud de la ruta más corta entre los vértices  $i$  y  $j$ . ■

Ejemplo 5. Considérese la siguiente red:



Determinense las rutas más cortas entre todo par de vértices utilizando el algoritmo de Floyd.

Iteración 1. Las matrices C y A resultantes son:

$$C = \begin{bmatrix} 0 & 3 & \infty & \infty & \infty & 2 \\ \infty & 0 & -1 & \infty & \infty & 0 \\ \infty & \infty & 0 & -2 & \infty & 5 \\ \infty & \infty & \infty & 0 & \infty & 3 \\ 1 & \infty & \infty & 8 & 0 & \infty \\ \infty & \infty & \infty & \infty & 5 & 0 \end{bmatrix} \quad A = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 & 4 & 4 \\ 5 & 5 & 5 & 5 & 5 & 5 \\ 6 & 6 & 6 & 6 & 6 & 6 \end{bmatrix}$$

Se asigna:  $k = 1$  y se actualizan las matrices C y A.

$$C_{52} = \min \{\infty, 1 + 3\} = 4 \quad a_{52} = a_{12} = 1$$

$$C_{56} = \min \{\infty, 1 + 2\} = 3 \quad a_{56} = a_{16} = 1$$

Estos son los únicos elementos que se modifican puesto que  $C_{ii} = \infty$ , para  $i = 2, 3, 4, 6$  y  $C_{1j} = \infty$ , para  $j=3, 4, 5$ . Las matrices resultantes son entonces:

$$C = \begin{bmatrix} 0 & 3 & \infty & \infty & \infty & 2 \\ \infty & 0 & -1 & \infty & \infty & 0 \\ \infty & \infty & 0 & -2 & \infty & 5 \\ \infty & \infty & \infty & 0 & \infty & 3 \\ 1 & 4 & \infty & 8 & 0 & 3 \\ \infty & \infty & \infty & \infty & 5 & 0 \end{bmatrix} \quad A = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 & 4 & 4 \\ 5 & 1 & 5 & 5 & 5 & 1 \\ 6 & 6 & 6 & 6 & 6 & 6 \end{bmatrix}$$

Puesto que  $k < n = 6$  y  $C_{ii} = 0$ , para todo  $i$ , aún no se han terminado de revisar las rutas.

Iteración 2. Se asigna  $k = 2$  y se actualizan los elementos

$$C_{13} = \min \{\infty, 3 - 1\} = 2 \quad a_{13} = a_{23} = 2$$

$$C_{53} = \min \{\infty, 4 - 1\} = 3 \quad a_{53} = a_{23} = 2$$

$$C_{16} = \min \{2, 3 + 0\} = 2 \quad \text{no se modifica } a_{16}$$

$$C_{56} = \min \{3, 4 + 0\} = 3 \quad \text{no se modifica } a_{56}$$

Las matrices resultantes en esta iteración son:

$$C = \begin{bmatrix} 0 & 3 & 2 & \infty & \infty & 2 \\ \infty & 0 & -1 & \infty & \infty & 0 \\ \infty & \infty & 0 & -2 & \infty & 5 \\ \infty & \infty & \infty & 0 & \infty & 3 \\ 1 & 4 & 3 & 8 & 0 & 3 \\ \infty & \infty & \infty & \infty & 5 & 0 \end{bmatrix} \quad A = \begin{bmatrix} 1 & 1 & 2 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 & 4 & 4 \\ 5 & 1 & 2 & 5 & 5 & 1 \\ 6 & 6 & 6 & 6 & 6 & 6 \end{bmatrix}$$

Puesto que  $k = 2 < n = 6$  y  $C_{ii} = 0$ , para todo  $i$ , se realiza otra iteración.

Iteración 3. Se asigna  $k = 3$  y se actualizan los elementos

$$\begin{aligned} C_{14} &= \min \{\infty, 2 - 2\} = 0 & a_{14} &= a_{34} = 3 \\ C_{16} &= \min \{2, 2 + 5\} = 2 & & \text{no se modifica } a_{16} \\ C_{24} &= \min \{\infty, -1 - 2\} = -3 & a_{24} &= a_{34} = 3 \\ C_{26} &= \min \{0, -1 + 5\} = 0 & & \text{no se modifica } a_{26} \\ C_{54} &= \min \{8, 3 - 2\} = 1 & a_{54} &= a_{34} = 3 \\ C_{56} &= \min \{3, 3 + 5\} = 3 & & \text{no se modifica } a_{56} \end{aligned}$$

Las matrices resultantes de la tercera iteración son:

$$C = \begin{bmatrix} 0 & 3 & 2 & 0 & \infty & 2 \\ \infty & 0 & -1 & -3 & \infty & 0 \\ \infty & \infty & 0 & -2 & \infty & 5 \\ \infty & \infty & \infty & 0 & \infty & 3 \\ 1 & 4 & 3 & 1 & 0 & 3 \\ \infty & \infty & \infty & \infty & 5 & 0 \end{bmatrix} \quad A = \begin{bmatrix} 1 & 1 & 2 & 3 & 1 & 1 \\ 2 & 2 & 2 & 3 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 & 4 & 4 \\ 5 & 1 & 2 & 3 & 5 & 1 \\ 6 & 6 & 6 & 6 & 6 & 6 \end{bmatrix}$$

Como  $k < n$ , aún no se tiene la solución.

Iteración 4. Se asigna  $k = 4$  y se actualizan los elementos

$$C_{16} = \min \{2, 0 + 3\} = 2 \quad \text{no se modifica } a_{16}$$

$$C_{26} = \min \{0, -3 + 3\} = 0 \quad \text{no se modifica } a_{26}$$

$$C_{36} = \min \{5, -2 + 3\} = 1 \quad a_{36} = a_{46} = 4$$

$$C_{56} = \min \{3, 1 + 3\} = 3 \quad \text{no se modifica } a_{56}$$

En base a lo anterior se tienen las siguientes matrices.

$$C = \begin{bmatrix} 0 & 3 & 2 & 0 & \infty & 2 \\ \infty & 0 & -1 & -3 & \infty & 0 \\ \infty & \infty & 0 & -2 & \infty & 1 \\ \infty & \infty & \infty & 0 & \infty & 3 \\ 1 & 4 & 3 & 1 & 0 & 3 \\ \infty & \infty & \infty & \infty & 5 & 0 \end{bmatrix} \quad A = \begin{bmatrix} 1 & 1 & 2 & 3 & 1 & 1 \\ 2 & 2 & 2 & 3 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 & 4 \\ 4 & 4 & 4 & 4 & 4 & 4 \\ 5 & 1 & 2 & 3 & 5 & 1 \\ 6 & 6 & 6 & 6 & 6 & 6 \end{bmatrix}$$

Aún no se ha determinado la solución ya que  $k < n$ .

Iteración 5. Se asigna  $k = 5$  y se actualizan los elementos

$$C_{61} = \min \{\infty, 1 + 5\} = 6 \quad a_{61} = a_{51} = 5$$

$$C_{62} = \min \{\infty, 4 + 5\} = 9 \quad a_{62} = a_{52} = 1$$

$$C_{63} = \min \{\infty, 3 + 5\} = 8 \quad a_{63} = a_{53} = 2$$

$$C_{64} = \min \{\infty, 1 + 5\} = 6 \quad a_{64} = a_{54} = 3$$

$$C_{66} = \min \{0, 3 + 5\} = 0 \quad \text{no se modifica } a_{66}$$

Se tiene entonces, que las nuevas matrices son:

$$C = \begin{bmatrix} 0 & 3 & 2 & 0 & \infty & 2 \\ \infty & 0 & -1 & -3 & \infty & 0 \\ \infty & \infty & 0 & -2 & \infty & 1 \\ \infty & \infty & \infty & 0 & \infty & 3 \\ 1 & 4 & 3 & 1 & 0 & 3 \\ 6 & 9 & 8 & 6 & 5 & 0 \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & 1 & 2 & 3 & 1 & 1 \\ 2 & 2 & 2 & 3 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 & 4 \\ 4 & 4 & 4 & 4 & 4 & 4 \\ 5 & 1 & 2 & 3 & 5 & 1 \\ 5 & 1 & 2 & 3 & 6 & 6 \end{bmatrix}$$

$k < n$ , entonces se realiza otra iteración.

Iteración 6. Se asigna  $k = 6$ . Se actualizan los elementos:

$C_{11} = \min \{0, 6 + 2\} = 0$	no se modifica $a_{11}$
$C_{12} = \min \{3, 9 + 2\} = 3$	no se modifica $a_{12}$
$C_{13} = \min \{2, 8 + 2\} = 2$	no se modifica $a_{13}$
$C_{14} = \min \{0, 6 + 2\} = 0$	no se modifica $a_{14}$
$C_{15} = \min \{\infty, 5 + 2\} = 7$	$a_{15} = a_{65} = 6$
$C_{21} = \min \{\infty, 6 + 0\} = 6$	$a_{21} = a_{61} = 5$
$C_{22} = \min \{0, 9 + 0\} = 0$	no se modifica $a_{22}$
$C_{23} = \min \{-1, 8 + 0\} = -1$	no se modifica $a_{23}$
$C_{24} = \min \{-3, 6 + 0\} = -3$	no se modifica $a_{24}$
$C_{25} = \min \{\infty, 5 + 0\} = 5$	$a_{25} = a_{65} = 6$
$C_{31} = \min \{\infty, 6 + 1\} = 7$	$a_{31} = a_{61} = 5$
$C_{32} = \min \{\infty, 9 + 1\} = 10$	$a_{32} = a_{62} = 1$
$C_{33} = \min \{0, 8 + 1\} = 0$	no se modifica $a_{33}$
$C_{34} = \min \{-2, 6 + 1\} = -2$	no se modifica $a_{34}$
$C_{35} = \min \{\infty, 5 + 1\} = 6$	$a_{35} = a_{65} = 6$
$C_{41} = \min \{\infty, 6 + 3\} = 9$	$a_{41} = a_{61} = 5$
$C_{42} = \min \{\infty, 9 + 3\} = 12$	$a_{42} = a_{62} = 1$
$C_{43} = \min \{\infty, 8 + 3\} = 11$	$a_{43} = a_{63} = 2$
$C_{44} = \min \{0, 6 + 3\} = 0$	no se modifica $a_{44}$
$C_{45} = \min \{\infty, 5 + 3\} = 8$	$a_{45} = a_{65} = 6$
$C_{51} = \min \{1, 6 + 3\} = 1$	no se modifica $a_{51}$
$C_{52} = \min \{4, 9 + 3\} = 4$	no se modifica $a_{52}$
$C_{53} = \min \{3, 8 + 3\} = 3$	no se modifica $a_{53}$
$C_{54} = \min \{1, 6 + 3\} = 1$	no se modifica $a_{54}$
$C_{55} = \min \{0, 5 + 3\} = 0$	no se modifica $a_{55}$

De lo anteriormente calculado, se tienen las siguientes matrices C y A:

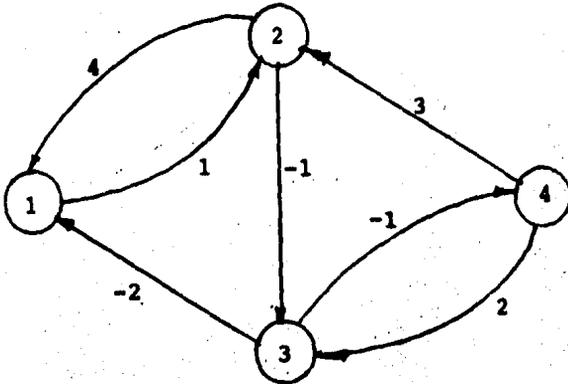
$$C = \begin{bmatrix} 0 & 3 & 2 & 0 & 7 & 2 \\ 6 & 0 & -1 & -3 & 5 & 0 \\ 7 & 10 & 0 & -2 & 6 & 1 \\ 9 & 12 & 11 & 0 & 8 & 3 \\ 1 & 4 & 3 & 1 & 0 & 3 \\ 6 & 9 & 8 & 6 & 5 & 0 \end{bmatrix} \quad A = \begin{bmatrix} 1 & 1 & 2 & 3 & 6 & 1 \\ 5 & 2 & 2 & 3 & 6 & 2 \\ 5 & 1 & 3 & 3 & 6 & 4 \\ 5 & 1 & 2 & 4 & 6 & 4 \\ 5 & 1 & 2 & 3 & 5 & 1 \\ 5 & 1 & 2 & 3 & 6 & 6 \end{bmatrix}$$

Puesto que  $k = n = 6$ , la última matriz es la matriz de longitudes más cortas entre todo par de vértices.

Para recuperar las rutas se utiliza la matriz A. Por ejemplo, la ruta de 1 a 1 se obtiene de la siguiente manera:

El predecesor de 1 en la ruta 1 a 1 es 1; es decir, se tiene la ruta vacía de longitud 0. La ruta entre los vértices 3 y 1 se obtiene del modo siguiente: el predecesor de 1, en la ruta de 3 a 1 es  $a_{31} = 5$ ; el predecesor de 5 en la ruta de 3 a 5 es  $a_{35} = 6$ ; el predecesor de 6 en la ruta de 3 a 6 es  $a_{36} = 4$ ; el predecesor de 4 en la ruta de 3 a 4 es  $a_{34} = 3$ . Entonces la ruta entre los vértices 3 y 1 es 3, 4, 6, 5, 1 de longitud  $C_{31} = 7$ . Para encontrar la ruta de 6 a 5 se tiene: el predecesor de 5 en la ruta de 6 a 5 es 6; entonces la ruta más corta entre los vértices 6 y 5 es 6, 5 de longitud  $C_{65} = 5$ . Análogamente se obtienen las demás rutas.

**Ejemplo 6.** Determinense las rutas más cortas, entre todo par de vértices, en la siguiente red:



**Iteración 1.** Las matrices C y A iniciales son:

$$C = \begin{bmatrix} 0 & 1 & = & \infty \\ 4 & 0 & -1 & = \\ -2 & = & 0 & -1 \\ = & 3 & 2 & 0 \end{bmatrix} \quad A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 \end{bmatrix}$$

Se asigna  $k = 1$  y se actualizan los elementos

$$C_{22} = \min \{0, 4 + 1\} = 0 \quad \text{no se modifica } a_{22}$$

$$C_{32} = \min \{=, -2 + 1\} = -1 \quad a_{32} = a_{12} = 1$$

Entonces las matrices resultantes son:

$$C = \begin{bmatrix} 0 & 1 & \infty & \infty \\ 4 & 0 & -1 & \infty \\ -2 & -1 & 0 & -1 \\ \infty & 3 & 2 & 0 \end{bmatrix} \quad A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 1 & 3 & 3 \\ 4 & 4 & 4 & 4 \end{bmatrix}$$

puesto que  $c_{ii} \geq 0$ , para toda  $i$ , y  $k = 1 < n = 4$ , aún no se ha determinado la solución.

Iteración 2. Se asigna  $k = 2$ . Se actualizan los elementos

$$C_{11} = \min \{0, 1 + 4\} = 0 \quad \text{no se modifica } a_{11}$$

$$C_{13} = \min \{\infty, 1 - 1\} = 0 \quad a_{13} = a_{23} = 2$$

$$C_{31} = \min \{-2, -1 + 4\} = -2 \quad \text{no se modifica } a_{31}$$

$$C_{33} = \min \{0, -1 - 1\} = -2 \quad a_{33} = a_{23} = 2$$

$$C_{41} = \min \{\infty, 3 + 4\} = 7 \quad a_{41} = a_{21} = 2$$

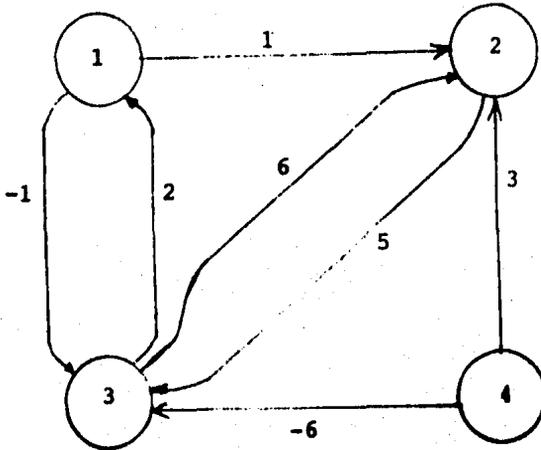
$$C_{43} = \min \{2, 3 - 1\} = 2 \quad \text{no se modifica } a_{43}$$

Las matrices C y A resultantes en esta iteración son:

$$C = \begin{bmatrix} 0 & 1 & 0 & \infty \\ 4 & 0 & -1 & \infty \\ -2 & -1 & -2 & -1 \\ 7 & 3 & 2 & 0 \end{bmatrix} \quad A = \begin{bmatrix} 1 & 1 & 2 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 1 & 2 & 3 \\ 2 & 4 & 4 & 4 \end{bmatrix}$$

Obsérvese que  $C_{33} < 0$ ; luego, puede concluirse que el problema es no acotado puesto que existe un circuito negativo que pasa por el vértice 3. En efecto, este circuito puede ponerse en evidencia utilizando la matriz A: el predecesor del vértice 3 en la ruta de 3 a él mismo es  $a_{33} = 2$ , el predecesor de 2 en la ruta de 3 a 2 es  $a_{32} = 1$ , el predecesor de 1 en la ruta de 3 a 1 es  $a_{31} = 3$ ; de este modo se obtiene el circuito 3, 1, 2, 3 de longitud  $C_{33} = - 2$ .

Ejemplo 7. Considérese la siguiente red y determinense las rutas más cortas entre todo par de vértices.



Se utilizará el algoritmo de Floyd para la determinación de las rutas.

Iteración 1. Las matrices C y A iniciales son las siguientes:

$$C = \begin{bmatrix} 0 & 1 & -1 & \infty \\ \infty & 0 & 5 & \infty \\ 2 & 6 & 0 & \infty \\ \infty & 3 & -6 & 0 \end{bmatrix} \quad A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 \end{bmatrix}$$

Sea  $k = 1$ . Se actualizan los valores de los elementos:

$$C_{32} = \min \{6, 2 + 1\} = 3 \quad a_{32} = a_{12} = 1$$

$$C_{33} = \min \{0, 2 - 1\} = 0 \quad \text{no se modifica } a_{33}$$

Entonces, las matrices resultantes de esta iteración son las siguientes:

$$C = \begin{bmatrix} 0 & 1 & -1 & \infty \\ \infty & 0 & 5 & \infty \\ 2 & 3 & 0 & \infty \\ \infty & 3 & -6 & 0 \end{bmatrix} \quad A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 1 & 3 & 3 \\ 4 & 4 & 4 & 4 \end{bmatrix}$$

Iteración 2. Se asigna  $k = 2$  y se actualizan los elementos:

$$C_{13} = \min \{-1, 1 + 5\} = -1 \quad \text{no se modifica } a_{13}$$

$$C_{33} = \min \{0, 3 + 5\} = 0 \quad \text{no se modifica } a_{33}$$

$$C_{43} = \min \{-6, 3 + 5\} = -6 \quad \text{no se modifica } a_{43}$$

Luego, las matrices C y A no fueron modificadas durante la segunda iteración.

Iteración 3. Se asigna  $k = 3$  y se actualizan los elementos

$$C_{11} = \min \{0, -1 + 2\} = 0 \quad \text{no se modifica } a_{11}$$

$$C_{12} = \min \{1, -1 + 3\} = 1 \quad \text{no se modifica } a_{12}$$

$$C_{21} = \min \{\infty, 5 + 2\} = 7 \quad a_{21} = a_{31} = 3$$

$$C_{22} = \min \{0, 5 + 3\} = 0 \quad \text{no se modifica } a_{22}$$

$$C_{41} = \min \{\infty, -6 + 2\} = -4 \quad a_{41} = a_{31} = 3$$

$$C_{42} = \min \{3, -6 + 3\} = -3 \quad a_{42} = a_{32} = 1$$

Las matrices A y C resultantes son, respectivamente:

$$C = \begin{bmatrix} 0 & 1 & -1 & \infty \\ 7 & 0 & 5 & \infty \\ 2 & 3 & 0 & \infty \\ -4 & -3 & -6 & 0 \end{bmatrix} \quad A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 3 & 2 & 2 & 2 \\ 3 & 1 & 3 & 3 \\ 3 & 1 & 4 & 4 \end{bmatrix}$$

Iteración 4. Se asigna  $k = 4$ . En esta iteración no se realiza ningún cambio puesto que  $C_{i4} = \infty$ , para  $i = 1, 2, 3$ . Entonces la última matriz C es la matriz de longitudes más cortas. El elemento  $a_{ij}$  de la última matriz A es el predecesor del vértice j, en la ruta más corta de i a j, siempre y cuando  $C_{ij} \neq \infty$ . Obsérvese que, puesto que  $C_{i4} = \infty$  (para  $i=1, 2, 3$ ), entonces no existe ninguna ruta entre los vértices i y 4 (para  $i = 1, 2, 3$ ); ésto puede verificarse fácilmente en la red.

**FLUJO MAXIMO**

**CAPITULO 3**

### 3.1 DESCRIPCIÓN DEL PROBLEMA

Un problema típico de flujo máximo entre origen y destino es el siguiente: Supóngase que en un poblado  $s$  se dispone de cierto producto. Este producto es demandado en un poblado  $t$ . Se requiere obtener la cantidad máxima posible del producto en  $t$  considerando que al transportarse, éste puede pasar por otros poblados que no tienen oferta ni demanda. Supóngase además que hay una capacidad máxima de transporte entre cada par de poblados. A este problema puede asociarse la red  $R = [X, A, q]$  donde:

- $X$  representa al conjunto de poblados,
- si  $i, j \in X$ ,  $(i, j) \in A \Leftrightarrow$  es posible transportar el producto del poblado  $i$  al poblado  $j$ .
- $q: A \rightarrow \mathbb{R}$ , donde para  $(i, j) \in A$ ,  $q(i, j) =$  capacidad máxima de transporte del poblado  $i$  al poblado  $j$ .

Se desea entonces determinar la cantidad del producto a transportar del poblado  $i$  al poblado  $j$  de manera tal que en  $t$  se obtenga la máxima cantidad posible de tal producto. Debe observarse que la cantidad total que entra a un poblado  $i$ , distinto de  $s$  y de  $t$ , debe ser igual a la cantidad total que sale de él puesto que no tiene oferta ni demanda; por otro lado, la cantidad total que se obtenga en  $t$  debe ser igual a la cantidad total que sale de  $s$  puesto que no se genera ni se pierde el producto a través de la red; por último la cantidad a enviar de  $i$  a  $j$ , para  $(i, j) \in A$ , debe ser menor o igual que  $q(i, j)$ . En general, en una red  $R = [X, A, q]$ , al número  $q(i, j)$  asociado a cada arco se le llama capacidad del arco  $(i, j)$  y a la cantidad a enviar a través de él se le llama flujo a través del arco  $(i, j)$ . Un flujo factible en  $R$  se define de la siguiente manera:

Definición. Un flujo factible en una red  $R = [X, A, q]$  es una función  $f: A \rightarrow \mathbb{R}$  tal que:

$$(i) \quad \sum_{j \in \Gamma^+(i)} f(i, j) - \sum_{k \in \Gamma^-(i)} f(k, i) = \begin{cases} v, & \text{si } i = s \\ 0, & \text{si } i \neq s, t \\ -v, & \text{si } i = t \end{cases}$$

$$(ii) \quad 0 \leq f(i, j) \leq q(i, j), \text{ para todo } (i, j) \in A.$$

Al número  $v$  se le llama valor del flujo  $f$  y a las ecuaciones (i) se les conoce como "ecuaciones de conservación de flujo"; a los vértices  $s$  y  $t$  (únicos con oferta y demanda respectivamente) se les llama origen y destino respectivamente. Por comodidad de notación se utilizará  $f_{ij}$  y  $q_{ij}$  para denotar  $f(i, j)$  y  $q(i, j)$  respectivamente.

Se dice que un flujo  $f$  es máximo, si es de valor máximo; es decir, si genera el mayor valor posible de  $v$ .

Para exponer el método de solución para el problema son necesarios los siguientes conceptos:

Sea  $R = [X, A, q]$  una red y sea  $f$  un flujo factible definido en ella. Sea  $C = (s=i_1, a_1, i_2, a_2, \dots, i_k, a_k, i_{k+1}=t)$  una cadena de  $s$  a  $t$  y sean  $F$  y  $B$  dos subconjuntos de arcos de  $C$  tales que:

$$a_j \in F \Leftrightarrow a_j = (i_j, i_{j+1})$$

$$a_j \in B \Leftrightarrow a_j = (i_{j+1}, i_j)$$

Es decir, los arcos de  $F$  son aquellos arcos de  $C$  que tienen el sentido de  $s$  hacia  $t$ ; los arcos de  $B$  son aquellos arcos de  $C$  que tienen el sentido inverso. En base a esto considérese la siguiente definición.

**Definición.** Una cadena de  $s$  a  $t$  es aumentante si  $f_{ij} < q_{ij}$  para todo  $(i,j) \in F$  y  $f_{ij} > 0$  para todo  $(i,j) \in B$ . Recibe el nombre de cadena aumentante ya que a través de ella puede enviarse flujo de  $s$  a  $t$  construyéndose, de este modo, un flujo factible de mayor valor. Las cadenas que se muestran en las figuras 1(a), 1(b) y 1(c) son aumentantes; la cadena de la figura 1(d) no es aumentante. Puede verificarse fácilmente que se construye un flujo factible  $f'$  de mayor valor que el flujo factible  $f$  definido en las primeras tres cadenas si se procede como sigue:

$$f'_{ij} = f_{ij} + z, \quad \text{para todo } (i,j) \in F$$

$$f'_{ij} = f_{ij} - z, \quad \text{para todo } (i,j) \in B$$

donde  $z$  es una cantidad tal que  $f_{ij} + z \leq q_{ij}$ , para todo  $(i,j) \in F$ , y  $f_{ij} - z \geq 0$  para todo  $(i,j) \in B$ . Nótese que si  $v$  es el valor de  $f$  entonces el valor de  $f'$  es  $v + z$ . Puesto que se desea el flujo máximo es importante calcular el mayor valor posible de  $z$ ; esta cantidad se conoce como capacidad incremental de la cadena.

**Definición.** La capacidad incremental de una cadena aumentante  $C$  es la máxima cantidad de flujo que puede enviarse aún a través de ella de  $s$  a  $t$ ; se denota con  $q(C)$ . En base a la manera de incrementar el flujo expuesta anteriormente,  $q(C)$  se calcula:

$$q(C) = \text{Min} \left\{ \begin{array}{l} \text{Min}_{(i,j) \in F} [q_{ij} - f_{ij}], \\ \text{Min}_{(i,j) \in B} [f_{ij}] \end{array} \right\}$$

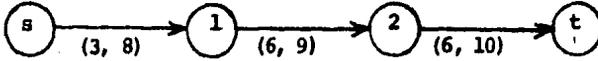
Las capacidades incrementales de las cadenas de las figuras 1(a), 1(b) y 1(c), son 3, 5 y 1 respectivamente.

Antes de exponer el método de solución para el problema del flujo máximo se demuestra un teorema que servirá para justificar dicho método.

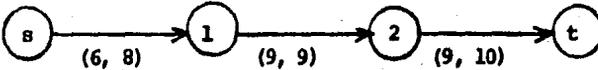
Figura 1.

En las siguientes cadenas la pareja de números asociada a cada arco es  $(f_{ij}, q_{ij})$ .

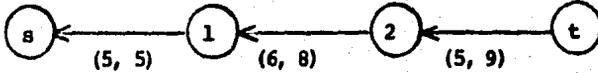
(a) Flujo inicial:



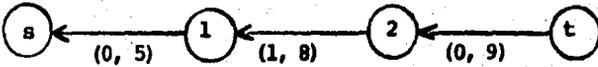
Flujo actualizado:



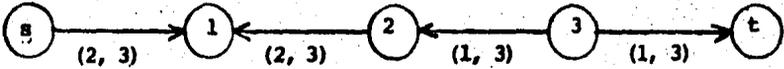
(b) Flujo inicial:



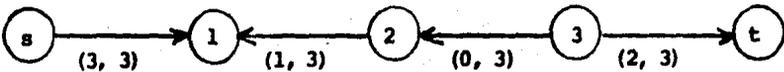
Flujo actualizado:



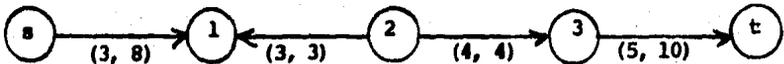
(c) Flujo inicial:



Flujo actualizado:



(d) Flujo inicial:



### 3.2 TEOREMA DE FLUJO MAXIMO - CORTADURA MINIMA.

En vista de los conceptos de cadena aumentante y capacidad incremental puede concluirse que un procedimiento natural para determinar el flujo máximo en una red es encontrar cadenas aumentantes en ésta e incrementar el flujo a través de ellas lo más que se pueda. Sin embargo, se requiere una herramienta que indique cuándo se alcanza la optimalidad. El teorema que se demuestra en esta sección proporciona dicha herramienta; antes, considérese lo siguiente:

Sean  $R = [X, A, q]$  una red y  $N \subset X$ . Sea  $\bar{N} = X - N$ . Se denota con  $(N, \bar{N})$  al conjunto de arcos que tienen un extremo en  $N$  y el otro fuera de  $N$ .

Definición. El conjunto de arcos  $(N, \bar{N})$  es una cortadura de  $R$  si  $s \in N$  y  $t \in \bar{N}$ , donde  $s$  y  $t$  son el origen y el destino de  $R$ . Obsérvese que si se remueve este conjunto de arcos de  $R$  ya no existe camino alguno de  $s$  a  $t$ .

Definición. La capacidad de una cortadura  $(N, \bar{N})$  es la suma de las capacidades de los arcos que la forman. Una cortadura mínima es aquella con mínima capacidad.

Existen ciertas relaciones entre cortaduras y flujos en una red. Una de ellas se demuestra en la siguiente proposición:

Proposición. Sea  $R = [X, A, q]$  una red. Sea  $f$  un flujo factible de valor  $v$  y sea  $(N, \bar{N})$  una cortadura de  $R$ . Entonces:

$$v \leq q(N, \bar{N})$$

Demostración. Sumando las ecuaciones de conservación de flujo para  $i \in N$  se tiene:

$$\begin{aligned} v &= \sum_{i \in N} \sum_{j \in \Gamma^+(i)} f_{ij} - \sum_{i \in N} \sum_{k \in \Gamma^-(i)} f_{ki} = \\ &= \sum_{\substack{i \in N \\ j \in N}} f_{ij} + \sum_{\substack{i \in N \\ j \in N}} f_{ij} - \sum_{\substack{i \in N \\ k \in N}} f_{ki} - \sum_{\substack{i \in N \\ k \in N}} f_{ki} = \\ &= \sum_{\substack{i \in N \\ j \in N}} f_{ij} - \sum_{\substack{i \in N \\ k \in N}} f_{ki} \end{aligned}$$

Por otro lado se sabe que  $0 \leq f_{ij} \leq q_{ij}$  para todo  $(i, j) \in A$ , entonces:

$$v = \sum_{\substack{i \in N \\ j \in N}} f_{ij} - \sum_{\substack{i \in N \\ k \in N}} f_{ki} \leq \sum_{\substack{i \in N \\ j \in N}} f_{ij} \leq \sum_{\substack{i \in N \\ j \in N}} q_{ij} = q(N, \bar{N})$$

con lo cual queda demostrada la proposición. ■

Otra de las relaciones importantes entre flujos y cortaduras se establece en el siguiente teorema que, como se había indicado anteriormente, proporciona una herramienta de optimalidad para el problema del flujo máximo. En este teorema se afirma que el valor del flujo máximo es igual a la capacidad de la cortadura mínima. La demostración del mismo es constructiva y durante ella se propone un método para incrementar el flujo; por ésto se contará no sólo con una herramienta de optimalidad sino también con un algoritmo para la solución del problema.

Teorema. (Flujo máximo-cortadura mínima). En una red  $R = [X, A, q]$  el valor del flujo máximo es igual a la capacidad de la cortadura mínima.

Demostración. En vista de la proposición anterior, basta demostrar que existe un flujo factible en  $R$  con valor igual a la capacidad de una cortadura de  $R$ . Para ello, considérese cualquier flujo factible  $f$  en  $R$  y constrúyase una cortadura aplicando el siguiente procedimiento:

- (i). Sea  $N = \{s\}$
- (ii). Si  $i \in N$  y  $f_{ij} < q_{ij}$  o  $f_{ji} > 0$ , agréguese  $j$  a  $N$ .

Repítase (ii) hasta que no pueda agregarse vértice alguno a  $N$ . Pueden presentarse entonces dos casos: o bien  $t \in N$ , o bien  $t \notin N$ .

Caso 1. Si  $t \in N$  entonces, dada la construcción de  $N$ , existe una cadena  $C$  de  $s$  a  $t$  tal que  $f_{ij} < q_{ij}$ , para todo  $(i, j) \in F$ , y  $f_{ij} > 0$ , para todo  $(i, j) \in B$ ; entonces esta cadena es aumentante y por lo tanto puede construirse un flujo mejor. En efecto, sea  $q(C)$  la capacidad incremental de  $C$  y redefínanse  $f$  y  $v$  como sigue:

$$f_{ij} = \begin{cases} f_{ij} + q(C), & \text{para todo } (i, j) \in F \\ f_{ij} - q(C), & \text{para todo } (i, j) \in B \\ f_{ij} & \text{en otro caso.} \end{cases}$$

$$v = v + q(C).$$

Obsérvese que  $f$  es un flujo factible de valor  $v$ .

Constrúyase de nuevo el conjunto  $N$  con el procedimiento (i),  
(ii) utilizando este flujo de mayor valor.

Caso 2. Si  $t \notin N$  entonces  $(N, \bar{N})$  es una cortadura de  $R$ . Por construcción se tiene que  $f_{ij} = q_{ij}$ , para todo  $(i, j) \in (N, \bar{N})$ , y  $f_{ij} = 0$ , para todo  $(i, j) \in (\bar{N}, N)$ ; entonces:

$$v = \sum_{\substack{i \in N \\ j \in N}} f_{ij} - \sum_{\substack{i \in N \\ k \in N}} f_{ki} = \sum_{\substack{i \in N \\ j \in N}} q_{ij} = q(N, \bar{N})$$

Bajo el supuesto de que  $q_{ij}$  es entero, para todo  $(i, j) \in A$ , en el caso 1 se incrementa el flujo en al menos una unidad; por esta razón el flujo máximo se obtiene en un número finito de pasos. Con esto queda demostrado el teorema. ■

### 3.3 METODO DE SOLUCION Y JUSTIFICACION.

El método que se describirá para resolver el problema del flujo máximo es el algoritmo de Ford y Fulkerson y consiste, a grandes rasgos, en lo siguiente: Se inicia con cualquier flujo factible. Primeramente se etiqueta el vértice origen  $s$  con dos etiquetas  $[s, \infty]$  indicando que en este vértice se dispone de cualquier cantidad de flujo. Después, si  $j$  es un vértice etiquetado y puede enviarse flujo de  $j$  a  $i$ ,  $i$  recibe dos etiquetas  $[\pm j, f(i)]$ . La primera etiqueta será de la forma  $+j$  si  $i \in \Gamma^+(j)$ ; es decir, si puede aumentarse el flujo a través del arco  $(j, i)$ . Será de la forma  $-j$  si  $i \in \Gamma^-(j)$ ; es decir, si puede disminuirse el flujo a través del arco  $(i, j)$ . La segunda etiqueta es la cantidad de flujo que puede enviarse de  $j$  a  $i$  y se calculará por tanto como el mínimo entre  $f(j)$  y  $h$ , donde:

$$h = \begin{cases} q_{ji} - f_{ji}, & \text{si } i \in \Gamma^+(j) \\ f_{ij}, & \text{si } i \in \Gamma^-(j) \end{cases}$$

Debe observarse que si  $h = 0$  no puede enviarse flujo de  $j$  a  $i$  por lo cual no se etiquetará  $i$  con  $[\pm j, f(i)]$ .

Este proceso de asignación de etiquetas a vértices se repetirá mientras sea posible. Si el vértice destino  $t$  recibe etiquetas entonces, dado el modo de etiquetar, existe una cadena aumentante de  $s$  a  $t$  con capacidad incremental igual a  $f(t)$  y por tanto se procederá a determinar ésta, con la ayuda de la primera etiqueta, y se actualizará el flujo a través de ella. Si, por el contrario,  $t$  no recibe etiqueta alguna, entonces se habrá determinado el flujo máximo. Para justificar esto último considére-

se el conjunto de arcos que tienen extremo inicial etiquetado y extremo final no etiquetado; este conjunto forma una cortadura de capacidad igual al valor del último flujo definido y por lo tanto éste es máximo. En efecto, nótese la similitud de este algoritmo con la demostración del teorema flujo máximo-cortadura mínima. El conjunto  $N$  que se define durante esta demostración corresponde al conjunto de vértices etiquetados ya que se construyen de la misma manera. Por esta misma razón, la justificación de optimalidad y convergencia del algoritmo está dada por dicho teorema. Además, en vista de la definición de  $N$  (o de la manera de etiquetar vértices), el algoritmo servirá también en casos donde sea de interés determinar la cortadura de capacidad mínima.

Es importante observar que el algoritmo converge sólo si las capacidades para el flujo en los arcos son enteras; sin embargo, en casos donde las capacidades sean racionales, pueden transformarse éstas en enteras multiplicándolas por la potencia de 10 adecuada. De este modo, el algoritmo puede utilizarse también en estos casos.

## ALGORITMO DE FORD Y FULKERSON

Propósito. Determinar el flujo máximo entre origen y destino en una red  $R = [X, A, q]$

### Descripción

1. Iniciar con cualquier flujo factible  $f$
2. Etiquetar en origen  $s$  con  $[s, \infty]$
3. Elegir un vértice etiquetado y no examinado; sea  $j$  éste vértice y sean  $[± k, f(j)]$  sus etiquetas.
  - (i) A todo  $i \in \Gamma^+(j)$  que no esté etiquetado y tal que  $f_{ji} < q_{ji}$  asignar la etiqueta  $[+j, f(i)]$ , donde  $f(i) = \min \{f(j), q_{ji} - f_{ji}\}$ .
  - (ii) A todo  $i \in \Gamma^-(j)$  que no esté etiquetado y tal que  $f_{ij} > 0$  asignar la etiqueta  $[-j, f(i)]$ , donde  $f(i) = \min \{f(j), f_{ij}\}$

- Se dice ahora que el vértice  $j$  ha sido examinado -
4. Repetir el paso 3 hasta que suceda (i) o (ii):
  - (i) El vértice destino  $t$  no tiene etiqueta y todos los vértices etiquetados han sido examinados. Terminar ya que el flujo factible  $f$  es máximo.
  - (ii) El vértice  $t$  recibe etiqueta. Ir al paso 5.
5. Sea  $x = t$ .
  - (i) Si la etiqueta de  $x$  es de la forma  $[+z, f(x)]$  hacer
$$f_{zx} = f_{zx} + f(x)$$
  - (ii) Si la etiqueta de  $x$  es de la forma  $[-z, f(x)]$  hacer
$$f_{xz} = f_{xz} - f(x)$$
6. Si  $z = s$ , borrar todas las etiquetas y regresar al paso 2.  
Si  $z \neq s$ , hacer  $x = z$  y regresar al paso 5.

## Justificación del algoritmo

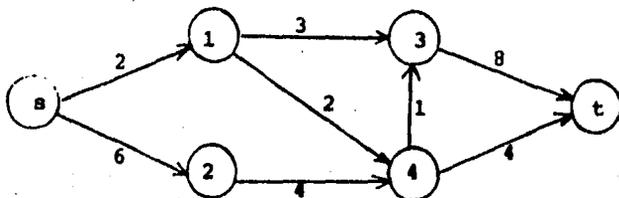
Como se había mencionado anteriormente, la justificación del algoritmo queda dada por el teorema de flujo máximo-cortadura mínima. ■

## Cortadura mínima

Al terminar de aplicar el algoritmo  $t$  no tiene etiqueta; luego, la cortadura de capacidad mínima está dada por el conjunto arcos:

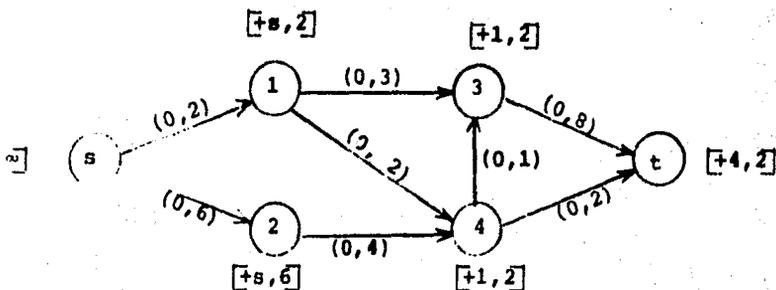
$(N, \bar{N})$ , donde  $N = \{x \in X \mid x \text{ tiene etiqueta}\}$

**Ejemplo 1.** Determinése el flujo máximo de  $s$  a  $t$  en la siguiente red mediante el algoritmo de Ford y Fulkerson. El número asociado a cada arco representa su capacidad.



Se aplicará el algoritmo utilizando inicialmente un flujo igual a cero a través de todos los arcos de la red.

**Iteración 1.** A continuación se muestran las etiquetas asignadas a los vértices durante esta iteración. Los números asociados a cada arco son el flujo a través de él y su capacidad.



Primeramente se etiquetó el vértice  $s$  con  $[s, \infty]$ , después se calcularon las otras etiquetas conforme a los pasos 3 y 4 del algoritmo.

Se elige el vértice  $s$  puesto que está etiquetado pero no examinado. Los sucesores  $i$  de  $s$  no etiquetados y tales que  $f_{si} < q_{si}$  son 1 y 2, por lo que:

1 recibe etiqueta  $[+s, \min \{\infty, 2\}]$

2 recibe etiqueta  $[+s, \min \{\infty, 6\}]$

El vértice  $s$  ha sido examinado.

Se elige el vértice 1 puesto que está etiquetado pero no examinado. Los sucesores  $i$  de 1 no etiquetados tales que

$f_{1i} < q_{1i}$  son 3 y 4, por lo que

3 recibe etiqueta  $[+1, \min \{2, 3\}]$

4 recibe etiqueta  $[+1, \min \{2, 2\}]$

El vértice 1 ha sido examinado.

Se elige el vértice 4.

El sucesor  $i$  de 4 no etiquetado y tal que  $f_{4i} < q_{4i}$  es  $t$ , de donde:

$t$  recibe etiqueta  $[+4, 2]$

El vértice 4 ha sido examinado.

Puesto que el vértice  $t$  recibió etiqueta existe una cadena aumentante de  $s$  a  $t$ . Se determina ésta y se actualiza el flujo a través de ella en los pasos 5 y 6 del algoritmo:

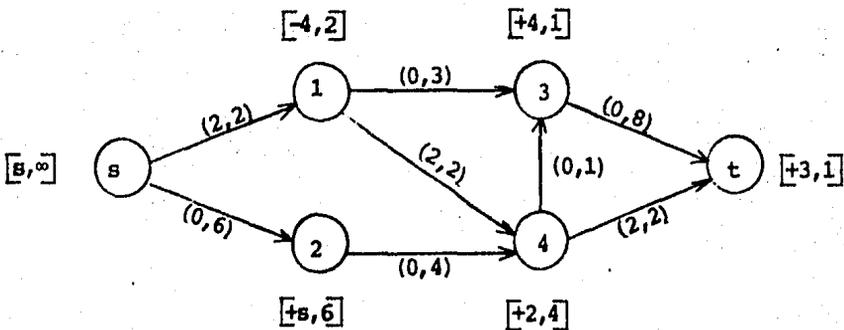
$x = t$  , su etiqueta es  $[+4, 2]$  , entonces  $f_{4t} = 2$

$x = 4$  , su etiqueta es  $[+1, 2]$  , entonces  $f_{14} = 2$

$x = 1$  , su etiqueta es  $[+s, 2]$  , entonces  $f_{s1} = 2$

Con este nuevo flujo de valor 2 se realiza otra iteración.

Iteración 2. Las etiquetas resultantes de esta iteración se muestran en la siguiente red:



De nuevo, se etiquetó s con  $[s, \infty]$ . Después:

Se elige el vértice s.

El vecino i de s no etiquetado y tal que  $f_{si} < q_{si}$  es 2, de donde:

2 recibe etiqueta  $[+s, \min\{\infty, 6\}]$

El vértice s ha sido examinado.

Se elige el vértice 2.

El único sucesor no etiquetado de 2 es 4 y  $f_{24} < q_{24}$  por lo que:

4 recibe etiqueta  $[+2, \min\{6, 4\}]$ .

El vértice 2 ha sido examinado.

Se elige el vértice 4.

El sucesor  $i$  no etiquetado de 4 tal que  $f_{4i} < q_{4i}$  es 3, de donde: 3 recibe etiqueta  $[+4, \{\min 4, 1\}]$ .

El predecesor de 4 no etiquetado es 1 y  $f_{14} > 0$ , de donde: 1 recibe etiqueta  $[-4, \{\min\{4, 2\}\}]$ .

El vértice 4 ha sido examinado.

Se elige el vértice 3.

El sucesor no etiquetado de 3 es  $t$  y  $f_{3t} < q_{3t}$ , de donde:  $t$  recibe etiqueta  $[+3, \{\min\{1, 8\}\}]$ .

El vértice 3 ha sido examinado.

Puesto que  $t$  recibió etiqueta existe una cadena aumentante de  $s$  a  $t$  con capacidad incremental igual a 1. Se determina ésta y se actualiza el flujo a través de ella:

$x = t$  , su etiqueta es  $[+3, 1]$  , entonces  $f_{3t} = 1$

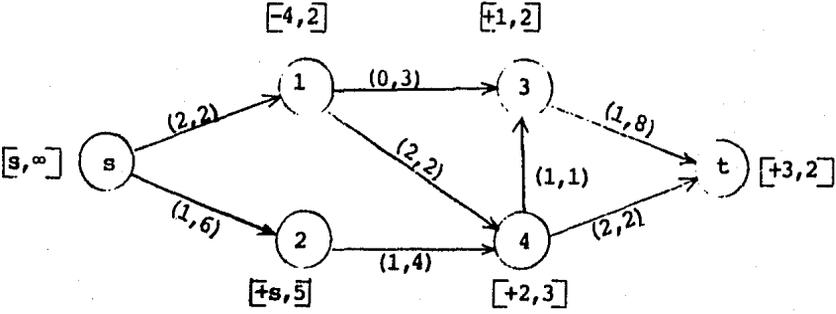
$x = 3$  , su etiqueta es  $[+4, 1]$  , entonces  $f_{43} = 1$

$x = 4$  , su etiqueta es  $[+2, 4]$  , entonces  $f_{24} = 1$

$x = 2$  , su etiqueta es  $[+s, 6]$  , entonces  $f_{s2} = 1$

Con este nuevo flujo factible de valor 3 se realiza otra iteración.

Iteración 3. De nuevo, las etiquetas asignadas se muestran en la siguiente red:



De nuevo, se etiquetó s con  $[s, \infty]$ . Luego:

Se elige el vértice s.

El sucesor de s no etiquetado es 2 y  $f_{s2} < q_{s2}$ , entonces: 2 recibe etiqueta  $[+s, \min\{\infty, 6-1\}]$ .

El vértice s ha sido examinado.

Se elige el vértice 2.

El sucesor no etiquetado de 2 es 4 y  $f_{24} < q_{24}$ , entonces: 4 recibe etiqueta  $[+2, \min\{5, 4-1\}]$ .

El vértice 2 ha sido examinado.

Se elige el vértice 4.

El predecesor no etiquetado de 4 es 1 y  $f_{14} > 0$ , luego: 1 recibe etiqueta  $[-4, \min\{3, 2\}]$ .

El vértice 4 ha sido examinado.

Se elige el vértice 1

El sucesor no etiquetado de 1 es 3 y  $f_{13} < q_{13}$ , por tanto 3 recibe etiqueta  $[+1, \min\{2,3\}]$ .

El vértice 1 ha sido examinado.

Se elige el vertice 3

El sucesor no etiquetado de 3 es t y  $f_{3t} < q_{3t}$ , por tanto t recibe etiqueta  $[+3, \min\{2,8-1\}]$ .

El vértice 3 ha sido examinado.

El vértice t recibió etiqueta por lo cual existe una cadena aumentante de capacidad incremental 2. Se determina ésta y se actualiza el flujo.

$x = t$ , su etiqueta es  $[+3, 2]$ , entonces  $f_{3t} = 1+2 = 3$

$x = 3$ , su etiqueta es  $[+1, 2]$ , entonces  $f_{13} = 2$

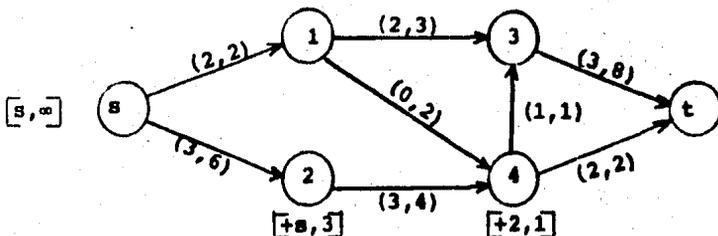
$x = 1$ , su etiqueta es  $[-4, 2]$ , entonces  $f_{14} = 2-2 = 0$

$x = 4$ , su etiqueta es  $[+2, 3]$ , entonces  $f_{24} = 1+2 = 3$

$x = 2$ , su etiqueta es  $[+s, 5]$ , entonces  $f_{s2} = 1+2 = 3$

Con este nuevo flujo factible de valor 5 se realiza otra iteración:

Iteración 4. Las etiquetas asignadas se muestran a continuación.



De nuevo la etiqueta de  $s$  es  $[s, \infty]$ , después:

Se elige el vértice  $s$

El sucesor  $i$  no etiquetado de  $s$  tal que  $f_{si} < q_{si}$  es  $2$ , por tanto:  
 $2$  recibe etiqueta  $[+s, \min(\infty, 6-3)]$ .

El vértice  $s$  ha sido examinado.

Se elige el vértice  $2$ .

El sucesor no etiquetado de  $2$  es  $4$  y  $f_{24} < q_{24}$ , de donde:  
 $4$  recibe etiqueta  $[+2, \min\{3, 4-3\}]$ .

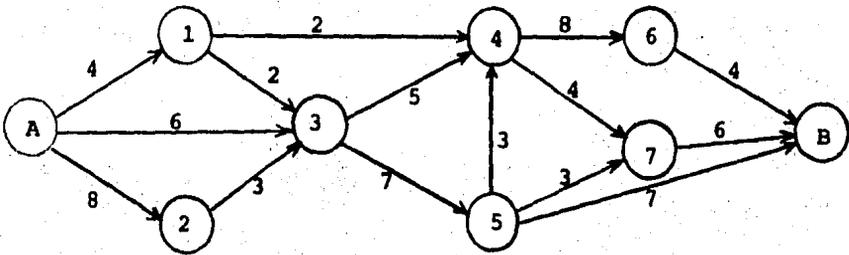
El vértice  $2$  ha sido examinado.

Se elige el vértice  $4$ .

Ningún vecino de  $4$  puede recibir etiqueta.

Todos los vértices etiquetados han sido examinados y  $t$  no recibió etiqueta. Por tanto, el último flujo es máximo. Por otro lado, el conjunto de vértices etiquetados es  $N = \{s, 2, 4\}$ ; por lo tanto, la cortadura mínima es  $(N, \bar{N}) = \{(s, 1), (4, 3) \text{ y } (4, t)\}$ . Obsérvese que su capacidad es  $q(N, \bar{N}) = 2+1+2 = 5$  que es igual al valor del flujo máximo.

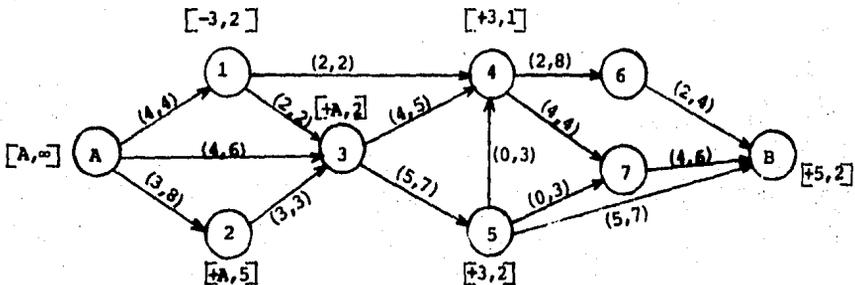
Ejemplo 2. Un gran número de personas viajan en automóvil de la ciudad A a la ciudad B. Las rutas posibles se muestran en la red siguiente. El departamento de policía de caminos desea construir suficientes casetas de inspección de tal manera que todo automóvil pase por al menos una de ellas en su trayectoria de A a B. El costo de construcción de las casetas varía según su localización; el costo asociado con cada tramo se proporciona en cada arco de la red (en millones de pesos). Determinése dónde deberán colocarse las casetas si se desea incurrir en el mínimo costo.



Obsérvese que, en términos de redes, se desea obtener un conjunto de arcos (tramos donde deberán construirse las casetas) de manera tal que si se eliminan esos arcos de la red ya no existan caminos de A a B (es decir, todo automóvil debe utilizar alguno o algunos de estos arcos en su trayectoria); este conjunto de arcos forma entonces una cortadura de la red. Por otro lado, se desea incurrir en el mínimo costo; debe determinarse entonces la cortadura de mínimo costo. Si se consideran los costos

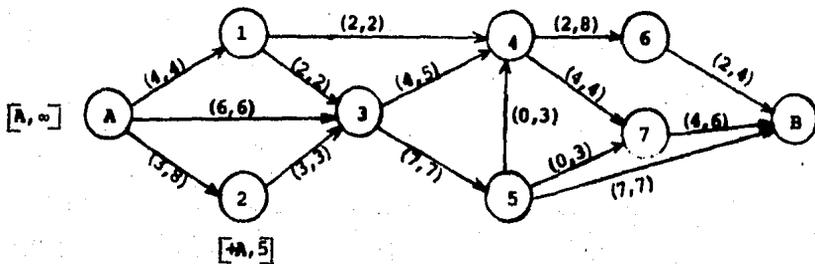
de los arcos como capacidades de un cierto flujo a través de ellos y se determina la cortadura de capacidad mínima, es claro que ésta corresponde a la de mínimo costo. Para ello se determinará el flujo máximo de A a B mediante el algoritmo de Ford y Fulkerson.

Iteración 1. Se aplicará el algoritmo con el flujo factible inicial de valor 11 definido en la siguiente red. Los números asociados a cada arco son el flujo a través de él y su capacidad. Las etiquetas asignadas a cada vértice durante esta iteración también se muestran en la red.



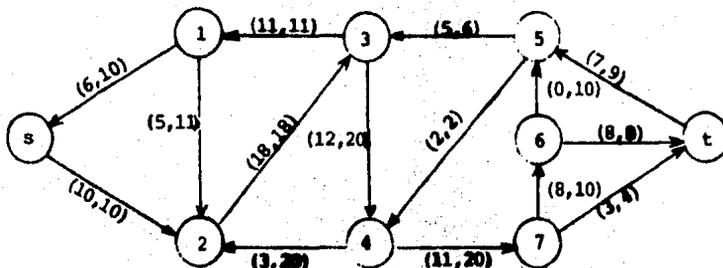
Puesto que  $f(B)=2$  existe una cadena aumentante de capacidad incremental igual a 2; ésta es: A, 3, 5, B. Actualizando el flujo a través de esta cadena se obtiene el definido en la siguiente red.

Iteración 2. De nuevo se asocia a cada vértice su correspondiente etiqueta.



En esta iteración el vértice B no recibió etiqueta; por lo tanto este flujo de valor 13 es el máximo. El conjunto de vértices no etiquetados es  $N = \{A, 2\}$  por lo que la cortadura de capacidad mínima es  $(N, \bar{N}) = \{(A, 1), (A, 3), (2, 3)\}$  de capacidad  $q(N, \bar{N}) = 4 + 6 + 3 = 13$ . Por tanto, la solución para el problema del departamento de policía de caminos es construir las casetas de inspección en los tramos (A, 1), (A, 3) y (2, 3) con un costo de construcción de 13 millones de pesos.

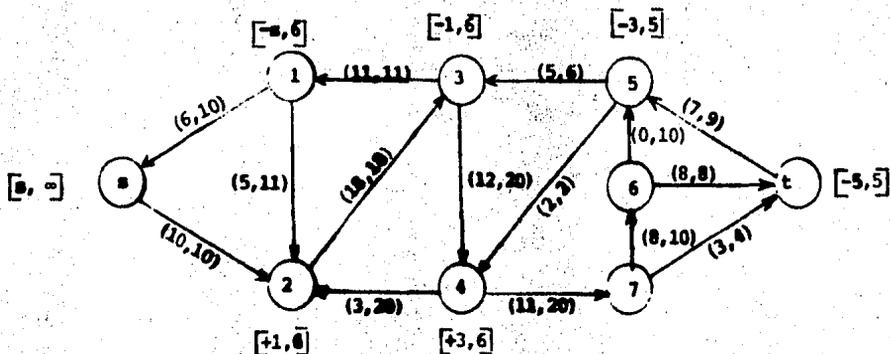
**Ejemplo 3.** Determinese el flujo máximo en la siguiente red a partir del flujo factible definido en ella:



Los números asociados a cada arco son el flujo a través de él y su capacidad.

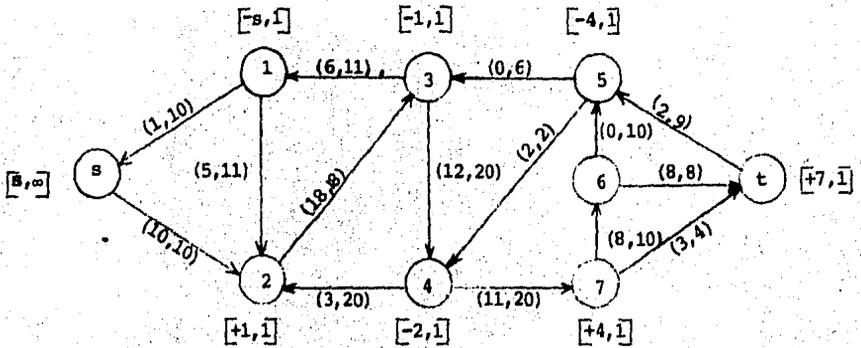
Obsérvese primeramente que el valor de este flujo es  $v=4$ .

**Iteración 1.** Las etiquetas asignadas a los vértices durante esta iteración se muestran en la siguiente red:



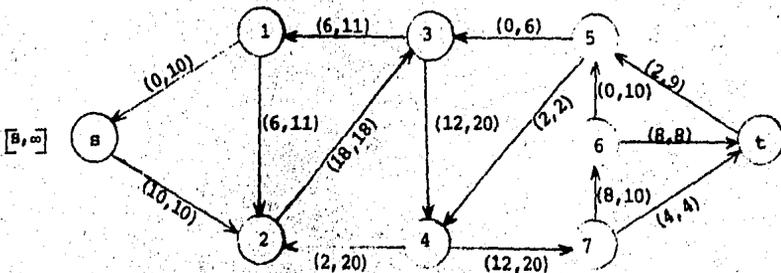
Puesto que  $f(t) = 5$  se ha determinado una cadena aumentante con capacidad incremental igual a 5 unidades de flujo, ésta es:  $s, 1, 3, 5, t$ . Actualizando el flujo se obtiene el definido en la siguiente red.

Iteración 2. De nuevo se asocia a cada vértice su correspondiente etiqueta:



Puesto que  $f(t)=1$  se ha determinado otra cadena aumentante con capacidad incremental igual a 1; ésta es:  $s, 1, 2, 4, 7, t$ . Actualizando el flujo se obtiene el definido en la siguiente red.

Iteración 3. De nuevo se asocia a cada vértice su etiqueta.



En esta iteración  $t$  no pudo ser etiquetado. Luego este flujo de valor 10 es máximo. Puesto que el conjunto de vértices etiquetados es  $N = \{s\}$ , la cortadura mínima está dada por:  $(N, \bar{N}) = \{(s, 2)\}$  de capacidad igual a 10.

### 3.4 VARIANTES DEL PROBLEMA

En esta parte del capítulo se analizan tres variantes del problema del flujo máximo en una red. Estas son:

- (a) Existen en la red varios orígenes o destinos.
- (b) Algunos vértices en la red tienen capacidad máxima para el flujo que pasa por ellos.
- (c) Los arcos de la red tienen cotas inferiores para el flujo a través de ellos.

Cada uno de estos problemas será reducido al problema tratado en las tres secciones anteriores por lo cual los métodos de solución variarán muy poco del algoritmo de Ford y Fulkerson.

#### (a) Caso donde existen varios orígenes o destinos.

Supóngase que la red  $R = [X, A, q]$  tiene  $m$  orígenes:  $s_1, s_2, \dots, s_m$ ,  $n$  destinos  $t_1, t_2, \dots, t_n$  y que puede enviarse flujo de cualquier origen a cualquier destino. El problema consiste en determinar el flujo máximo que puede enviarse, a través de la red, de todos los orígenes a todos los destinos. Supóngase que a la red  $R$  se asocia la red  $R' = [X', A', q']$  donde:

$$X' = XU\{s, t\}$$

$$A' = AU\{(s, s_i) \mid i=1, \dots, m\} \cup \{(t_j, t) \mid j=1, \dots, n\}$$

y

$$q'(i, j) = \begin{cases} q_{ij}, & \text{si } (i, j) \in A \\ \infty, & \text{si } i=s \text{ o } j=t. \end{cases}$$

Obsérvese que entonces el problema se reduce a determinar el flujo máximo de  $s$  a  $t$  en la red  $R'$ . En efecto, una vez que se determine dicho flujo en  $R'$ , bastará con definir en  $R$  el mismo flujo definido en los arcos de  $A$  en  $R'$ . Por tanto, para resolver este tipo de problemas, puede utilizarse el algoritmo de Ford y Fulkerson sin modificación alguna.

(b) Caso donde se permiten capacidades en vértices.

Sea  $R = [X, A, q, k]$  una red en la cual se han definido una función  $q$ , que asocia a cada arco la capacidad del flujo a través de él, y una función  $k$  que asocia a cada vértice la capacidad del flujo que puede pasar por él. Un flujo factible  $f$  en  $R$  es entonces una función, asociada a los elementos de  $A$ , que cumple los puntos (i) y (ii) de la sección 3.1 y además:

$$\sum_{j \in I^-(i)} f_{ji} \leq k(i), \text{ para todo } i \in X$$

Supóngase que se desea determinar el flujo máximo de  $s$  a  $t$  en la red  $R$ . Para esto, considérese una red  $R'$  construida de la siguiente manera: A cada vértice  $i$  de  $R$ , con capacidad finita, corresponde un arco  $(i^-, i^+)$  en  $R'$  con capacidad  $k(i)$ ; todos los arcos con extremo final  $i$  en  $R$  corresponden, en  $R'$ , a arcos de la forma  $(j, i^-)$  ó  $(j^+, i^-)$ , si  $j$  tiene capacidad finita; por último, todos los arcos con extremo inicial en  $R$  corresponden, en  $R'$ , a arcos de la forma  $(i^+, j)$  ó  $(i^+, j^-)$  si  $j$  tiene capacidad finita. Todos los arcos de  $R$  conservan su capacidad. Si  $R'$  está definida de esta manera se tiene que, a partir de cualquier flujo

definido en esta red, puede definirse otro del mismo valor en la red  $R$  y viceversa. En efecto, si  $f'$  es un flujo factible de valor  $v$  en  $R'$  y se define  $f$  como  $f_{ki} = f'_{ki^-}$  y  $f_{ij} = f'_{i+j}$ , para todo  $i \in X$  con capacidad finita, y  $f_{ij} = f'_{ij}$ , en otro caso, es fácil verificar que  $f$  es un flujo factible en  $R$ . Si  $i \in X$  tiene capacidad  $k(i)$  y  $V$  es el conjunto de vértices de  $R'$ , se tiene que:

$$\sum_{j \in X} f_{ij} - \sum_{k \in X} f_{ki} = \sum_{j \in V} f'_{i+j} - \sum_{k \in V} f'_{ki^-} =$$

$$\sum_{j \in V} f'_{i+j} - f'_{i-i^+} = \begin{cases} v & \text{si } i = s \\ 0 & \text{si } i \neq s, t \\ -v & \text{si } i = t \end{cases}$$

Por otro lado:

$$\sum_{j \in \Gamma^-(i)} f_{ji} = \sum_{j \in V} f'_{ji^-} = f'_{i-i^+} \leq k(i),$$

luego,  $f$  es un flujo factible de valor  $v$  en  $R$ . Determinar el flujo máximo en  $R$  es equivalente entonces, a determinarlo en  $R'$ . Por tanto, puede concluirse, de nuevo, que para resolver este tipo de problemas puede utilizarse el algoritmo de Ford y Fulkerson.

Por último, es importante observar que, en este caso, la cortadura de capacidad mínima de  $R'$  puede corresponder a un conjunto de arcos y vértices de  $R$ .

(c) Caso donde se permiten cotas inferiores en los arcos.

En la práctica es necesario, algunas veces, asociar una cota inferior al flujo a través de un arco. Por ejemplo, si la red representa un oleoducto, puede tenerse interés en enviar no menos de cierta cantidad de flujo puesto que no valdría la pena incurrir en costos de envío de una cantidad muy pequeña; en este caso se asocia como cota inferior la cantidad mínima de flujo deseada. Una cota inferior puede representar también una demanda por satisfacer.

Considérese una red  $R = [X, A, r, q]$  donde  $q$  es la función de capacidad de flujo asociada a los arcos de  $R$  y  $r$  es una función que asocia, a cada arco  $(i, j)$  de  $R$ , la cota inferior  $r_{ij}$  permitida para el flujo a través de  $(i, j)$ . En este caso se tiene un flujo factible  $f$  si se cumplen las ecuaciones de conservación de flujo y  $r_{ij} \leq f_{ij} \leq q_{ij}$ , para todo  $(i, j) \in A$ . Nótese que, de hecho, el problema tratado en las primeras tres secciones es un caso particular de éste ( $r_{ij} = 0$ , para todo  $(i, j) \in A$ ).

El primer problema que surge cuando se intenta maximizar el flujo del origen al destino en una red con estas restricciones es determinar un flujo factible inicial; claramente, en estos casos, un flujo igual a cero a través de todos los arcos no es factible.

Para determinar dicho flujo factible se utilizará la red  $R' = [XU\{s', t'\}, AUA'U\{(t, s)\}, q']$ , donde:

- Para todo  $(i,j) \in A$ , si  $r_{ij} \neq 0$ , se definen los arcos  $(s',j)$ ,  $(i,t') \in A'$
- $q'_{ij} = q_{ij} - r_{ij}$ , para todo  $(i,j) \in A$
- $q'_{ts} = 0$
- Para todo  $(i,j) \in A$ , si  $r_{ij} \neq 0$ , se define  $q'_{s',j} = q'_{i,t'} = r_{ij}$
- Las cotas inferiores para el flujo a través de los arcos de  $R'$  son todas 0.

Si para algún  $i \in X$  existen varios arcos de la forma  $(i,j)$  con  $r_{ij} \neq 0$  se define un sólo arco  $(i,t')$  con capacidad  $q'_{it'}$ , igual a la suma de las cotas inferiores de estos arcos. Análogamente se procede si existen varios arcos de la forma  $(k,i)$  con  $r_{ki} \neq 0$ , para algún  $i \in X$ .

Si el flujo máximo de  $s'$  a  $t'$ , en la red  $R'$ , tiene valor igual a la suma de las cotas inferiores de los arcos de  $R$ , puede demostrarse que existe un flujo factible en  $R$ . Esto se hará en el teorema que a continuación se enuncia. Este resultado se demuestra constructivamente proporcionando así una manera de obtener el flujo factible deseado.

**Teorema.** Sea  $F$  el flujo máximo de valor  $v'$  en  $R'$  y sea  $F_{ts}$  el flujo a través del arco  $(t,s)$ . Si  $v' = \sum_{(i,j) \in A} r_{ij}$  entonces existe un flujo factible de valor  $F_{ts}$  en  $R$ .

Demostración. Por ser  $F$  un flujo factible en  $R'$ , se tiene que

$$(i) \sum_{j \in \Gamma^+(i)} F_{ij} - \sum_{k \in \Gamma^-(i)} F_{ki} = \begin{cases} v', & \text{si } i=s' \\ 0, & \text{si } i \neq s', t' \\ -v', & \text{si } i=t' \end{cases}$$

$$(ii) 0 \leq F_{ij} \leq q_{ij}, \text{ para todo } (i,j) \in AUA'U\{(t,s)\}$$

Por otro lado, por ser  $v' = \sum_{(i,j) \in A} r_{ij}$  se tiene que el flujo a través de los arcos de la forma  $(s',j)$  y de la forma  $(i,t')$  es igual a la capacidad de dichos arcos.

El flujo  $f$  definido como  $f_{ij} = F_{ij} + r_{ij}$ , para todo  $(i,j) \in A$ , es factible de valor  $F_{ts}$  en  $R$ . Para probar esto se debe verificar que se cumplen:

$$(a) \sum_{j \in \Gamma^+(i)} f_{ij} - \sum_{k \in \Gamma^-(i)} f_{ki} = \begin{cases} F_{ts}, & \text{si } i=s \\ 0, & \text{si } i \neq s, t \\ -F_{ts}, & \text{si } i=t \end{cases}$$

$$(b) r_{ij} \leq f_{ij} \leq q_{ij}, \text{ para todo } (i,j) \in A$$

Primeramente se probará (a) para  $i \neq s, t$ :

$$\sum_j f_{ij} - \sum_k f_{kj} = \sum_{j \neq t'} F_{ij} + \sum_{j \neq t'} r_{ij} - \sum_{k \neq s'} F_{ki} - \sum_{k \neq s'} r_{ki} =$$

$$\sum_{j \neq t'} F_{ij} + F_{it'} - \sum_{k \neq s'} F_{ki} - F_{s'i} = \sum_j F_{ij} - \sum_k F_{ki} = 0 \text{ (por (i)).}$$

Ahora, si  $i=s$ :

$$\sum_j f_{sj} - \sum_k f_{ks} = \sum_{j \neq t} F_{sj} + \sum_{j \neq t} r_{sj} - \sum_{\substack{k \neq s \\ k \neq t}} F_{ks} - \sum_{\substack{k \neq s \\ k \neq t}} r_{ks} =$$

$$\sum_j F_{sj} - \sum_{k \neq t} F_{ks} = F_{ts} \quad (\text{por (i)}).$$

Análogamente se prueba la ecuación de conservación de flujo para  $t$ .

Para probar (b) se tiene que, de (ii):

$$0 + r_{ij} \leq F_{ij} + r_{ij} \leq q'_{ij} + r_{ij},$$

pero  $f_{ij} = F_{ij} + r_{ij}$  y  $q'_{ij} = q_{ij} - r_{ij}$ , por tanto:

$$r_{ij} \leq f_{ij} \leq q_{ij}.$$

Luego,  $f$  es un flujo factible de valor  $F_{ts}$  en  $R$  por lo cual el teorema queda demostrado. ■

Por otro lado, debe observarse no siempre existe un flujo factible, en redes con estas características. Se utilizará de nuevo la red  $R'$  para detectar cuándo no existe solución. Esto se hará en el siguiente teorema.

Teorema. Sea  $F$  el flujo máximo de  $s'$  a  $t'$ , de valor  $v'$  en  $R'$ . Si  $v' \neq \sum_{(i,j) \in A} r_{ij}$ , entonces no existe ningún flujo factible en  $R$ .

Demostración. La demostración del teorema se hará por contradicción. Supóngase que existe un flujo factible de valor  $v$  en  $R$ .

Entonces se cumplen:

$$(i) \quad \sum_j f_{ij} - \sum_k f_{ki} = \begin{cases} v, & \text{si } i=s \\ 0, & \text{si } i \neq s, t \\ -v, & \text{si } i=t \end{cases}$$

$$(ii) \quad r_{ij} \leq f_{ij} \leq q_{ij}, \quad \text{para todo } (i,j) \in A.$$

Definase el flujo  $F$  en  $R'$  como sigue:

$$F_{ij} = f_{ij} - r_{ij}, \text{ para todo } (i,j) \in A$$

$$F_{ts} = v$$

$$F_{it'} = \sum_j r_{ij}, \text{ donde } r_{ij} \text{ es la cota inferior de los arcos de la forma } (i,j) \in A$$

$$F_{s'j} = \sum_k r_{kj}, \text{ donde } r_{kj} \text{ es la cota inferior de los arcos de la forma } (k,j) \in A$$

Primeramente se mostrará que  $F$  es un flujo factible en  $R'$  de valor  $v' = \sum_{(i,j) \in A} r_{ij}$ . Para ello deberá verificarse que se cumplen:

$$(a) \sum_j F_{ij} = \sum_k F_{ki} = \begin{cases} v', & \text{si } i=s' \\ 0, & \text{si } i \neq s', t' \\ -v', & \text{si } i=t' \end{cases}$$

$$(b) 0 \leq F_{ij} \leq q'_{ij}, \text{ para todo } (i,j) \in A \cup A' \cup \{(t,s)\}$$

Sea  $i \neq s', t', s, t$ :

$$\sum_j F_{ij} - \sum_k F_{ki} = \sum_{j \neq t'} f_{ij} + F_{it'} - \sum_j r_{ij} - \sum_{k \neq s'} f_{ki} - F_{s'i} + \sum_k r_{ki}$$

Pero  $F_{it'} = \sum_j r_{ij}$  y  $F_{s'i} = \sum_k r_{ki}$ ; luego, la cantidad anterior es igual a:

$$\sum_{j \neq t'} f_{ij} - \sum_{k \neq s'} f_{ki} = 0 \quad (\text{por (1)})$$

Para  $i=s$ , se tiene:

$$\sum_j F_{sj} - \sum_k F_{ki} = \sum_{j \neq t'} f_{sj} + F_{st'} - \sum_j r_{sj} - \sum_{k \neq s'} f_{ks} - F_{s's} - F_{ts} + \sum_k r_{ks}$$

$$\sum_{j \neq t} f_{sj} - \sum_{\substack{k \neq s \\ k \neq t}} f_{ks} - v = v - v = 0. \quad (\text{por } (i)).$$

Análogamente se prueba que (a) se cumple para  $i=t$ .

Obsérvese que  $\Gamma^-(s') = \emptyset$  por lo que:

$$\sum_j F_{s',j} - \sum_k F_{ks'} = \sum_j F_{s',j} = \sum_{(i,j) \in A} r_{ij}$$

$$\text{Análogamente se prueba } \sum_j F_{t',j} - \sum_k F_{kt'} = - \sum_{(i,j) \in A} r_{ij}.$$

Para verificar (b) se tiene que, de (ii):

$$r_{ij} - r_{ij} \leq f_{ij} - r_{ij} \leq q_{ij} - r_{ij}, \quad \text{para todo } (i,j) \in A$$

Pero  $F_{ij} = f_{ij} - r_{ij}$  y  $q_{ij} = q_{ij} - r_{ij}$ , para todo  $(i,j) \in A$ ; luego:

$$0 \leq F_{ij} \leq q_{ij}, \quad \text{para todo } (i,j) \in A$$

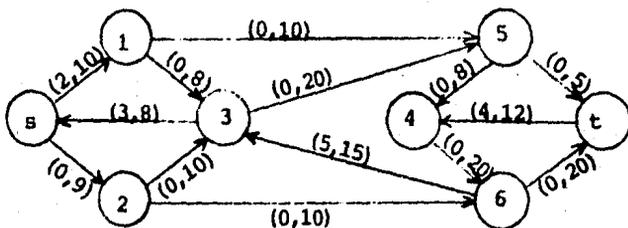
Por otro lado, el flujo definido para los arcos de la forma  $(i, t')$  y  $(s', j)$  es igual a su capacidad. Por último, la capacidad del arco  $(t, s)$  es infinita. Luego, (b) se cumple para todo  $(i, j) \in A \cup A' \cup \{(t, s)\}$ ; de donde,  $F$  es un flujo factible de valor  $\sum_{(i,j) \in A} r_{ij}$  en  $R'$ .

Nótese que el conjunto de arcos de  $R'$  con extremo inicial  $s'$  forman una cortadura de capacidad  $\sum_{(i,j) \in A} r_{ij}$  por lo cual  $F$  es un flujo máximo en  $R'$ . Esto es una contradicción a la hipótesis y por tanto se concluye la afirmación del teorema. ■

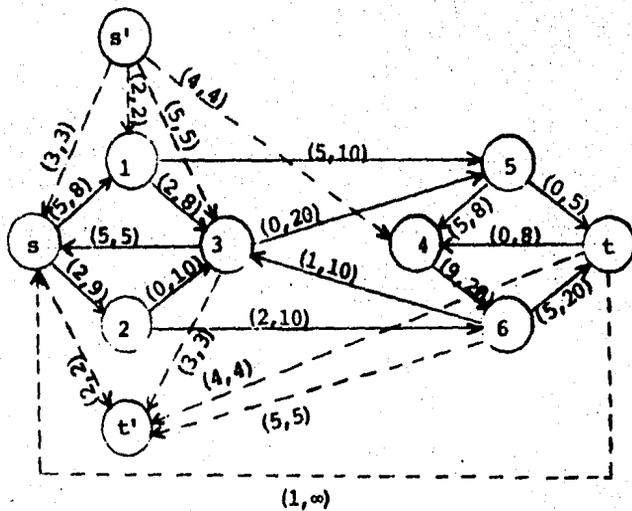
Una vez que se ha resuelto el problema de determinar un flujo factible en  $R$ , si existe, debe procederse a determinar si éste es máximo o no y, en caso de que no lo sea, construir otro flujo de valor mayor. Para esto debe observarse que una cadena  $C$ , de  $s$  a  $t$ , es aumentante, en este tipo de redes, si  $f_{ij} < q_{ij}$ , para todo  $(i,j) \in F$ , y  $f_{ij} > r_{ij}$ , para todo  $(i,j) \in B$ , puesto que en estos arcos debe decrementarse el flujo. En base a esto es claro que el algoritmo de Ford y Fulkerson puede aplicarse en estos casos con una sola modificación. La única modificación consiste en cuándo y cómo deben etiquetarse los predecesores de un vértice; es decir, el inciso (ii) del paso 3 deberá indicar:

3.(ii) A todo  $i \in \Gamma^-(j)$  que no esté etiquetado y tal que  $f_{ij} > r_{ij}$  asignar la etiqueta  $[-j, f(i)]$ , donde  $f(i) = \min\{f(j), f_{ij} - r_{ij}\}$ .

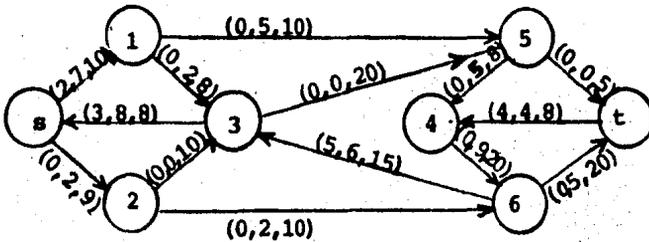
**Ejemplo 4.** Determinése el flujo máximo en la siguiente red R en donde los números asociados a cada arco  $(i,j)$  son  $r_{ij}$  y  $q_{ij}$ .



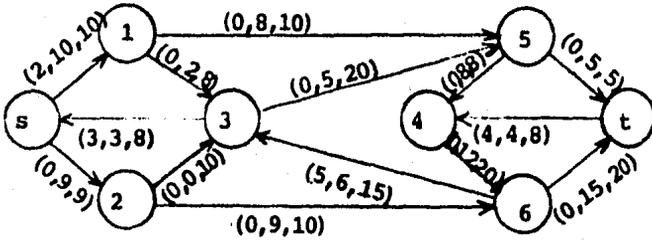
Primeramente se determinará un flujo factible en la red con ayuda de la red  $R'$  que se presenta a continuación:



El segundo número asociado a cada arco  $(i,j)$  es su capacidad  $q_{ij}$ ; el primero es el flujo  $F_{ij}$  definido. Puede verificarse fácilmente que  $F$  es un flujo máximo en  $R'$  de valor 14 que es igual a la suma de las cotas inferiores de los arcos de  $R$ . Luego, existe un flujo factible en  $R$  de valor  $F_{ts} = 1$ . En base a la demostración del teorema correspondiente este flujo es  $f_{ij} = F_{ij} + r_{ij}$ , para todo arco  $(i,j)$  de  $R$ :



Los números asociados a los arcos  $(i,j)$  de esta red son  $r_{ij}$ ,  $f_{ij}$  y  $q_{ij}$ , respectivamente. Utilizando el algoritmo de Ford y Fulkerson, con la modificación mencionada en esta sección, se determina la cadena aumentante  $s,3,5,t$  con capacidad incremental igual a 5 unidades de flujo. Una vez actualizado el flujo a través de los arcos de esta cadena, se determina la cadena  $s,2,6,t$  de capacidad incremental igual a 7. Por último se determina la cadena aumentante  $s,1,5,4,6,t$  de capacidad incremental 3. Por lo tanto el flujo máximo en la red es de valor 16 y se presenta como el segundo número asociado a los arcos de la siguiente red:



**FLUJO A COSTO MINIMO ENTRE ORIGEN Y DESTINO**

#### 4.1 DESCRIPCION DEL PROBLEMA

En el capítulo anterior se analiza el problema del flujo máximo entre origen y destino en una red  $R$  sin tomar en cuenta que puede tenerse un costo asociado al flujo a través de cada arco; este costo asociado al arco puede describir costos unitarios de transporte, almacenamiento, producción, etc.

Ahora se considera el problema de elegir el mejor flujo, bajo el criterio de su costo. En general, puede tenerse interés en elegir el mejor flujo, entre origen y destino, de cierto valor que no es necesariamente el máximo. Sea  $R = [X, A, q, c]$  una red con una función  $q$  de capacidad y una función  $c$  de costos por unidad de flujo asociadas a sus arcos. Se define como el costo de un flujo factible  $f$  a la cantidad  $\sum_{(i,j) \in A} c_{ij} f_{ij}$ , donde  $c_{ij}$  es el costo unitario del flujo a través del arco  $(i,j)$  y  $f_{ij}$  es la cantidad de flujo a través de  $(i,j)$ . Supóngase que se tiene interés en determinar un flujo factible de valor  $v$ , entre el origen y el destino, incurriendo en el menor costo posible; al flujo de menor costo se le llama flujo a costo mínimo de valor  $v$ .

Claramente, si la cantidad  $v$  de flujo requerido es mayor que el valor del flujo máximo el problema no tiene solución, por lo cual de aquí en adelante se supondrá que  $v$  no cumple esto.

Antes de exponer métodos de solución para el problema se presentarán algunos conceptos básicos que serán de gran utilidad.

## 4.2 CONCEPTOS BASICOS

Considérese la red de la figura 1. Supóngase que se requiere el flujo de valor  $v=5$  a costo mínimo. De alguna manera, que se rá explicada posteriormente, se ha determinado el flujo de valor 5 y de costo 48 que se muestra en la figura 2. Obsérvese que si se modifica el flujo a través del ciclo 1,2,3,1 se obtiene un flujo de menor costo con el mismo valor  $v=5$ . Dicha modificación consiste en incrementar, en una unidad, el flujo a través de los arcos (1,2), (2,3) y decrementar, en una unidad, el flujo a través del arco (1,3). Este último flujo, de costo 46, se muestra en la figura 3.

De aquí se concluye que, en general, es necesaria una herramienta que permita determinar si un flujo factible dado es de costo mínimo o no y, en caso de que no sea, proporcione una manera de construir un flujo mejor a partir de aquél con el que se cuenta. Esta herramienta está dada por el concepto de red marginal o incremental que a continuación se define.

Definición. Sea  $R = [\bar{X}, A, q, c]$  una red y sea  $f$  un flujo factible definido en  $R$ . La red marginal o incremental de  $R$ , con respecto al flujo  $f$ , es la red  $R'(f) = [\bar{X}, A_1 \cup A_2, q', c']$ , donde:

- $A_1 = \{(i, j) \in A \mid f_{ij} < q_{ij}\}$ ,
- $A_2 = \{(i, j) \mid (j, i) \in A \text{ y } f_{ji} > 0\}$ ,
- $q'$  describe la capacidad de los arcos de  $R'(f)$  de la siguiente manera:

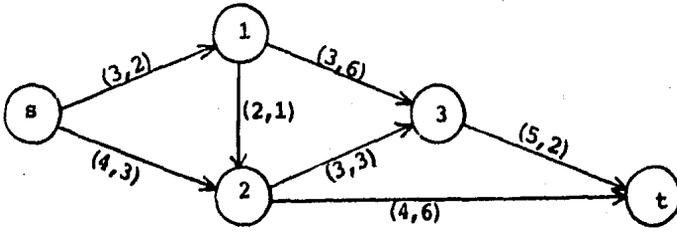


Figura 1.

Los números asociados a cada arco son (capacidad, costo)

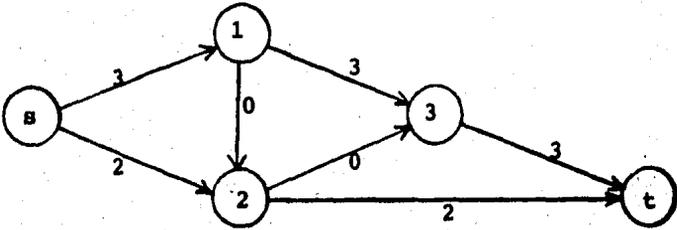


Figura 2.

El número asociado a cada arco es el flujo a través de él.

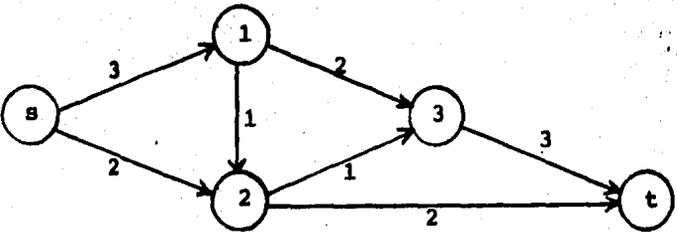


Figura 3.

El número asociado a cada arco es el flujo a través de él.

$$q'_{ij} = q_{ij} - f_{ij} , \text{ para todo } (i,j) \in A_1$$

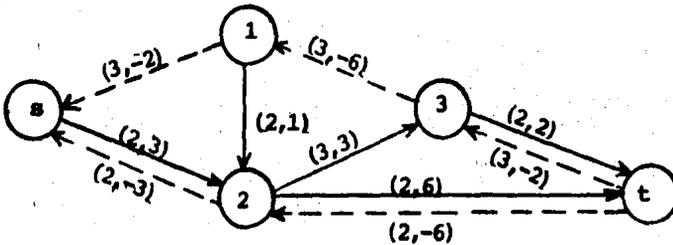
$$q'_{ij} = f_{ji} , \text{ para todo } (i,j) \in A_2$$

-  $c'$  describe el costo unitario del flujo a través de los arcos de  $R'(f)$  de la siguiente manera:

$$c'_{ij} = c_{ij} , \text{ para todo } (i,j) \in A_1$$

$$c'_{ij} = -c_{ji} , \text{ para todo } (i,j) \in A_2$$

La red marginal  $R'(f)$  de la red de la figura 1, con respecto al flujo definido en la figura 2 es:



Los números asociados a cada arco son, respectivamente, la capacidad y el costo unitario del flujo a través de él.

Obsérvese que, en general, un circuito en la red  $R'(f)$  corresponde a un ciclo en la red  $R$  con la característica de que es posible modificar el flujo a través de él. En efecto, cada arco  $(i,j) \in A_1$  corresponde a uno de  $A$  con  $f_{ij} < q_{ij}$  por lo cual el flujo puede aumentarse en este arco; cada arco  $(i,j) \in A_2$  corresponde a un arco  $(j,i) \in A$  con  $f_{ji} > 0$  por lo cual puede decrementarse el flujo a través de él.

Por otro lado, se obtiene un nuevo flujo factible del mismo valor  $v$  si se modifica en cierta cantidad  $d$  el flujo a través del ciclo de la siguiente forma:

$$f'_{ij} = \begin{cases} f_{ij} & , \text{ si } (i,j) \text{ no pertenece al ciclo} \\ f_{ij} + d & , \text{ si } (i,j) \in A_1 \text{ y pertenece al ciclo} \\ f_{ij} - d & , \text{ si } (j,i) \in A_2 \text{ y el arco } (i,j) \text{ pertenece al ciclo.} \end{cases}$$

Es fácil verificar que el flujo  $f'$  así definido es factible de valor  $v$ . Para ello se demostrará que:

$$\sum_j f'_{ij} - \sum_k f'_{ki} = \begin{cases} v, & \text{ si } i=s \\ 0, & \text{ si } i \neq s, t \\ -v, & \text{ si } i=t \end{cases}$$

Esto es claro si el vértice  $i$  no está en el ciclo. Ahora, si  $i$  es un vértice del ciclo, entonces tiene dos vecinos  $m$  y  $n$  en dicho ciclo. Si en el circuito correspondiente en  $R'(f)$  están los arcos  $(m,i)$  e  $(i,n)$  existen varios casos a considerar:

- (a)  $(m,i), (i,n) \in A_1$ . Esto implicaría que  $m$  es predecesor de  $i$  y  $n$  es sucesor de  $i$  en el ciclo.



- (b)  $(m,i), (i,n) \in A_2$ . Entonces  $m$  es sucesor de  $i$  y  $n$  es predecesor de  $i$  en el ciclo.



- (c)  $(m,i) \in A_1, (i,n) \in A_2$ . En este caso  $m$  y  $n$  son predecesores de  $i$  en el ciclo.



(d)  $(m,i) \in A_2, (i,n) \in A_1$ . Por tanto  $m$  y  $n$  son sucesores de  $i$  en el ciclo.



En el caso (a) se tiene:

$$\sum_j f'_{ij} - \sum_k f'_{ki} = \sum_{j \neq n} f'_{ij} + f'_{in} - \sum_{k \neq m} f'_{ki} - f'_{mi} =$$

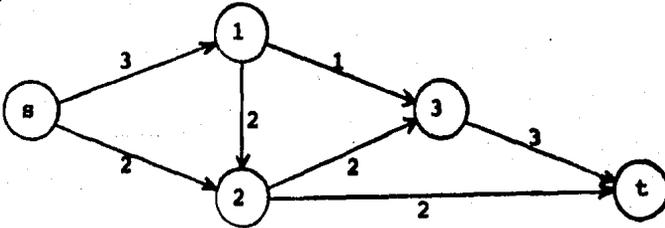
$$\sum_{j \neq n} f_{ij} + (f_{in} + d) - \sum_{k \neq m} f_{ki} - (f_{mi} + d) =$$

$$\sum_j f_{ij} - \sum_k f_{ji} = \begin{cases} v, & \text{si } i=s \\ 0, & \text{si } i \neq s, t \\ -v, & \text{si } i=t \end{cases}$$

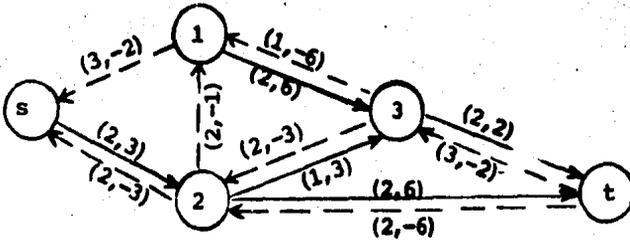
Análogamente se prueban los otros tres casos.

La red marginal indica entonces la manera de obtener otros flujos factibles del mismo valor. Más aún, indica el cambio en el costo al modificar dicho flujo. En efecto, incrementar una unidad de flujo a través del arco  $(i,j) \in A$  implica incrementar el costo  $c_{ij}$  unidades; decrementarlo implica decrementar el costo  $c_{ij}$  unidades y son precisamente  $c_{ij}$  y  $-c_{ij}$  los costos asignados a los arcos de  $R'(f)$  en el primer y segundo casos, respectivamente. De aquí que si se modifica el flujo, en una unidad, a través de un ciclo, el cambio total en el costo está dado por el costo del circuito correspondiente en  $R'(f)$ . Puede entonces concluirse que si existe algún circuito negativo en  $R'(f)$  conviene modificar el flujo a través del ciclo correspondiente mejorando así el costo. Además, puede verificarse que este comportamiento es lineal por lo que es deseable determinar la máxima cantidad,  $d$ , en que puede modificarse el flujo a través de un ciclo. Claramente,  $d$

debe ser tal que no rebase la capacidad de los arcos del circuito en  $R'(f)$  ya que estas capacidades representan el cambio posible en los arcos de  $R$ . Por tanto  $d = \text{Min} \{q'_{ij}, \text{ con } (i,j) \text{ en el circuito}\}$ . En el ejemplo de la figura 1, se tiene el circuito negativo 1,2,3,1, de costo -2, en la red marginal. Al modificar el flujo en una unidad a través del ciclo 1,2,3,1 (figura 2) el costo mejoró en 2 unidades. En este caso  $d = \text{Min} \{2, 3, 3\} = 2$  de donde se concluye que puede modificarse el flujo una unidad más a través del ciclo con un nuevo decremento de 2 unidades en el costo, es decir, un decremento total de 4 unidades.



Este flujo tampoco es de costo mínimo puesto que existe el circuito negativo 1,s,2,3,1.



En base al concepto de red marginal se enuncia el siguiente teorema en el cual se plasman los resultados discutidos anteriormente. Con él se propone, de hecho, un algoritmo para determinar el flujo a costo mínimo en una red.

Teorema. Sea  $f$  un flujo factible de valor  $v$  en la red  $R = [X, A, q, c]$ . Entonces el flujo  $f$  es de costo mínimo si y sólo si no existen circuitos negativos en la red marginal  $R'(f)$ .

Demostración. Sean  $c(f)$  el costo del flujo  $f$  y  $c[C|R'(f)]$  el costo de un circuito  $C$  en  $R'(f)$ .

Primeramente se demostrará, por contradicción, que si  $f$  es de costo mínimo entonces no existen circuitos negativos en  $R'(f)$ .

Para ello supóngase que existe algún circuito  $C$  en  $R'(f)$  tal que  $c[C|R'(f)] < 0$ . Se denotará con  $f + 1$  o  $(C)$  al flujo definido en  $R$  de la siguiente manera, donde  $K$  es el ciclo correspondiente a  $C$ .

$$f_{ij} = \begin{cases} f_{ij} & , \text{ si } (i,j) \notin K \\ f_{ij} + 1 & , \text{ si } (i,j) \in A_1 \cap C \\ f_{ij} - 1 & , \text{ si } (j,i) \in A_2 \cap C \end{cases}$$

Es sencillo verificar que  $f + 1$  o  $(C)$  es de valor  $v$ . El costo de este nuevo flujo es:

$$c(f + 1 \text{ o } (C)) = c(f) + c[C|R'(f)] < c(f),$$

lo cual contradice la hipótesis.

Para demostrar la suficiencia de la condición se utilizará un resultado que no se prueba ya que para ello se requiere de conceptos no expuestos en este trabajo (Referencia 4). Este resultado establece que un flujo  $f$  de valor  $v$  puede descomponerse como:  $f = 1$  o  $(P_1) + 1$  o  $(P_2) + \dots + 1$  o  $(P_{v-1}) + 1$  o  $(P_v) + 1$  o  $(C_1) + 1$  o  $(C_2) + \dots + 1$  o  $(C_k)$  donde  $P_i$  es un camino elemental de  $s$  a  $t$  en  $R(i, 1, \dots, v)$ ,  $C_j$  es un circuito elemental en

$R(j=1, \dots, k)$  y  $1 \circ (S)$  denota un flujo igual a una unidad en los arcos de  $S$  y a cero unidades en los arcos fuera de  $S$ .

Supóngase que  $c[\bar{C} | R'(f)] \geq 0$ , para todo circuito  $C$  en  $R'(f)$ , y que  $f^*(f^* \neq f)$  es el flujo a costo mínimo de valor  $v$  en  $R$ . Denótese con  $f^* - f$  el flujo que describe  $f^*_{ij} - f_{ij}$  unidades de flujo a través del arco  $(i, j) \in A$ . En base al resultado anteriormente mencionado,  $f^* - f$  puede descomponerse como:

$$f^* - f = 1 \circ (C_1) + 1 \circ (C_2) + \dots + 1 \circ (C_k),$$

donde  $C_j$  es un circuito en  $R$ , para  $j=1, \dots, k$ . Entonces

$$f^* = f + 1 \circ (C_1) + 1 \circ (C_2) + \dots + 1 \circ (C_k).$$

Puesto que este último flujo es factible, se tiene que cualquier suma  $f + 1 \circ (C_1) + \dots + 1 \circ (C_r)$  es factible para  $1 \leq r \leq k$ .

Considérese el flujo  $f + 1 \circ (C_1)$ , entonces:

$$c(f + 1 \circ (C_1)) = c(f) + c[\bar{C}_1 | R'(f)] \geq c(f)$$

Por otro lado, puede verificarse que:

$$c[\bar{C}_r | R'(f + 1 \circ (C_1))] \geq c[\bar{C}_r | R'(f)], \text{ para } r=2, \dots, k$$

Luego, para el flujo  $f + 1 o(C_1) + 1 o(C_2)$ , se tiene:

$$\begin{aligned}c(f + 1 o(C_1) + 1 o(C_2)) &= c(f + 1 o(C_1)) + c[C_2 | R'(f + 1 o(C_1))] \\ &\geq c(f + 1 o(C_1)) + c[C_2 | R'(f)] \\ &\geq c(f + 1 o(C_1)) \\ &\geq c(f)\end{aligned}$$

Continuando con este razonamiento se llega a  $c(f^*) \geq c(f)$  de donde  $f$  es un flujo a costo mínimo de valor  $v$ . ■

Anteriormente se ha supuesto que se cuenta con (o se determina) un flujo inicial cualquiera del valor  $v$  requerido. Existe un procedimiento alternativo para atacar el problema. Este consiste en considerar un flujo factible de valor  $u$  ( $u < v$ ) de costo mínimo e ir incrementando el valor del flujo, hasta obtener el valor  $v$ , de manera tal que siempre se conserve la optimalidad; es decir, que cada flujo que se obtenga sea de costo mínimo. Para lograr esto se utilizará de nuevo la red marginal. Es claro que, para incrementar el valor de un flujo dado, es necesario determinar cadenas aumentantes de  $s$  a  $t$  en la red. Nótese que un camino de  $s$  a  $t$  en la red marginal, con respecto a un flujo dado, corresponde a una cadena aumentante de  $s$  a  $t$  en la red original y viceversa; la capacidad incremental de una cadena será igual a la de la ruta correspondiente en la red marginal. Por tanto, para determinar una cadena aumentante se determinará una ruta en la red marginal y se enviará, a través de esta cadena, una cantidad igual a su capacidad incremental o igual a la diferencia entre el valor de flujo requerido y el valor del flujo actual. Pero este proceso debe garantizar la conservación de la optimalidad. Será necesario elegir la ruta más corta en la red marginal, para este propósito, como se propone en el siguiente teorema.

**Teorema.** Sea  $f$  un flujo factible, en la red  $R = [\bar{X}, A, q, c]$ , de valor  $v$  y de costo mínimo. Sea  $P^*$  la ruta más corta de  $s$  a  $t$  en  $R'(f)$  y sea  $P$  la cadena aumentante, en la red  $R$ , correspondiente a  $P^*$ . Entonces el flujo  $f'$ , resultante de aumentar una unidad de flujo a través de la cadena  $P$ , es un flujo de valor  $v+1$  y de costo mínimo.

**Demostración.** Primeramente obsérvese que  $f'$  es de valor  $v+1$ . Por otro lado, puesto que  $f$  es de costo mínimo,  $R'(f)$  no contiene circuitos negativos. Para demostrar que  $f'$  también es de costo mínimo, basta probar que la red marginal  $R'(f')$  tampoco contiene circuitos negativos. Las redes marginales  $R'(f)$  y  $R'(f')$  coinciden excepto, posiblemente, en los arcos de  $P^*$ . Considérese la partición, de los arcos de  $P$ , en los conjuntos  $F$  y  $B$  definidos en el tercer capítulo. Si  $f'_{ij} = f_{ij} + 1 < q_{ij}$ , para todo  $(i, j) \in F$  y  $f'_{ij} = f_{ij} - 1 > 0$ , para todo  $(i, j) \in B$  es claro que  $R'(f)$  y  $R'(f')$  contienen los mismos arcos y por tanto  $R'(f')$  no contiene circuitos negativos en este caso. Ahora, si existe algún  $(i, j) \in F$  tal que  $f'_{ij} = f_{ij} + 1 = q_{ij}$  entonces este arco, que pertenece a  $R'(f)$ , no pertenece a  $R'(f')$  y el arco  $(j, i)$  pertenece a esta última red. Supóngase que este arco pertenece a un circuito  $C$  de  $R'(f')$ ; sean los vértices de  $C: i, 1, 2, \dots, k, j, i$ . Puesto que  $P^*$  es una ruta más corta de  $s$  a  $t$  en  $R'(f)$ , entonces el arco  $(i, j)$  es una ruta más corta de  $i$  a  $j$  ya que pertenece a  $P^*$ . Por lo tanto, en la red  $R'(f)$  se tiene que  $c'_{ij} \leq c'_{i1} + c'_{12} + \dots + c'_{kj}$ ; luego  $c'(C) = c'_{i1} + c'_{12} + \dots + c'_{kj} - c'_{ij} \geq 0$  y por tanto  $C$  no es un circuito negativo. Análogamente se prueba que  $R'(f')$  no contiene circuitos negativos si existe algún arco  $(i, j) \in B$  tal que  $f'_{ij} = f_{ij} - 1 = 0$ . ■

### 4.3 METODOS DE SOLUCION

Como se mencionó en la sección anterior, el problema del flujo a costo mínimo en una red  $R = [X, A, q, c]$  puede resolverse de dos maneras. Una de ellas involucra la determinación de circuitos negativos en la red marginal y la otra involucra la determinación de rutas más cortas en esta red.

El primero de estos procedimientos será llamado algoritmo basado en la eliminación de circuitos negativos y algunas veces se hace referencia a él como algoritmo Primal puesto que empieza a aplicarse a partir de un flujo factible del valor  $v$  requerido y, en cada iteración, este flujo se mejora, sin modificar su valor, hasta alcanzar la optimalidad. El segundo procedimiento será llamado algoritmo basado en rutas más cortas; a veces, en la literatura, se hace referencia a él como algoritmo Dual ya que se aplica a partir de un flujo factible de algún valor menor del valor  $v$  requerido pero de costo mínimo; en cada iteración se incrementa el valor del flujo, siempre conservando la optimalidad, hasta alcanzar  $v$ . Enseguida se comentan y describen estos métodos.

#### a. Método basado en eliminación de circuitos negativos.

El primer paso de este método consiste en determinar un flujo factible del valor  $v$  requerido. Para ello puede utilizarse el algoritmo de Ford y Fulkerson con una modificación en el criterio de alto. Esta modificación consiste en lo siguiente: si, en el paso 5, la etiqueta  $f(t)$  de  $t$  más el valor  $v'$  del flujo actual

es mayor o igual que  $v$ , entonces se modifica el flujo en la cantidad  $v-v'$  y se termina con el flujo del valor  $v$ ; si éste no es el caso se realiza otra iteración. Obsérvese que si se determina el flujo máximo, utilizando esta modificación del algoritmo de Ford y Fulkerson, y el valor de éste es menor que  $v$ , entonces el problema no tiene solución. Una vez determinado el flujo de valor  $v$  se procede a "probar" la optimalidad. En base al primer teorema expuesto en la sección precedente, "probar" la optimalidad consiste en verificar si en la red marginal existen circuitos negativos o no; si no existen, el flujo es de costo mínimo; si existe alguno, éste será "eliminado". Dicho en otras palabras: si existe un circuito negativo en la red marginal, se determina su capacidad incremental y se modifica el flujo a través del ciclo correspondiente, en la red original, en esta cantidad para obtener un flujo de menor costo. Puesto que el nuevo flujo a través de alguno de los arcos del ciclo es igual a su capacidad o a cero, el circuito negativo no estará en la red marginal con respecto a este nuevo flujo; es decir, el circuito fue "eliminado". Este procedimiento será aplicado hasta que se hayan "eliminado" todos los circuitos negativos. En este caso el flujo de valor  $v$  actual es de costo mínimo. Nótese que, durante este procedimiento, es necesario determinar circuitos negativos; para determinarlos puede utilizarse el algoritmo de Floyd o el algoritmo general para determinar arborescencias de rutas más cortas si es que existe alguna raíz (estos algoritmos fueron presentados en el segundo capítulo). En efecto, recuérdese que durante el algoritmo de Floyd se determinan las rutas más cortas entre todo par de vértices, si existen, o se determina algún circuito nega-

ivo concluyéndose, en este caso, que no hay solución al problema. Es posible, entonces, emplear este método para detectar los circuitos negativos en la red marginal o para garantizar la existencia de éstos en el caso de que el algoritmo termine con las rutas más cortas. A continuación se presenta detalladamente el algoritmo basado en la "eliminación" de circuitos negativos cuya justificación queda dada por el primer teorema expuesto en la sección anterior.

ALGORITMO BASADO EN ELIMINACION DE CIRCUITOS NEGATIVOS.

Propósito. Determinar el flujo a costo mínimo de valor  $v$  en la red  $R = [X, A, q, c]$ .

Descripción

1. Determinese un flujo factible  $f$  de valor  $v$  mediante el algoritmo de Ford y Fulkerson.
2. Constrúyase la red marginal, con respecto a  $f$ ,  $R'(f)$ .
3. Mediante algún algoritmo de rutas más cortas, identifíquese algún circuito negativo en  $R'(f)$ . Si no existen circuitos negativos, terminar. El flujo actual  $f$  es el requerido; en otro caso, sea  $C$  el circuito negativo. Ir al paso 4.
4. Sea  $d = \min_{(i,j) \in C} \{q_{ij}\}$ 
  - (i) Para todo arco  $(i,j) \in A$  tal que  $(i,j) \in A_1 \cap C$ , actualizar.  
 $f_{ij} = f_{ij} + d$
  - (ii) Para todo arco  $(i,j) \in A$  tal que  $(j,i) \in A_2 \cap C$ , actualizar.  
 $f_{ij} = f_{ij} - d$Con este nuevo flujo ir a 2.

Ejemplo 1. Determinése el flujo a costo mínimo de valor 5 en la red de la figura 1 (sección 4.2).

Iteración 1. Se aplicará el algoritmo a partir del flujo, de costo 44, determinado en la sección 4.2. Este flujo se muestra nuevamente en la figura 4. La red marginal con respecto a este flujo se muestra en la figura 5. Utilizando el algoritmo de Floyd, se determina el circuito negativo 1,s,2,3,1 de costo -2. De aquí se tiene  $d = \text{Min} \{q'_{1s}, q'_{s2}, q'_{23}, q'_{31}\} = 1$ . Actualizando el flujo se obtiene el definido en la figura 6.

Iteración 2. La red marginal con respecto a este nuevo flujo se muestra en la figura 7. Al utilizar el algoritmo de Floyd se observa que la red marginal no contiene circuitos negativos por lo que el flujo de costo 42 definido en la figura 7 es el flujo a costo mínimo de valor 5.

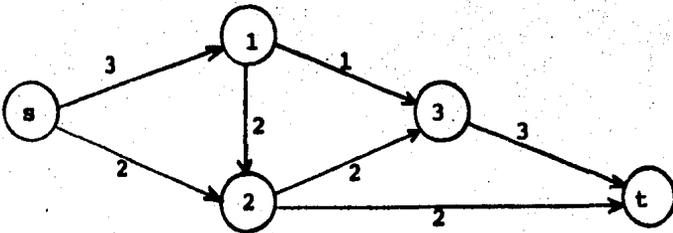


Figura 4.

Flujo de valor 5 de costo 44.

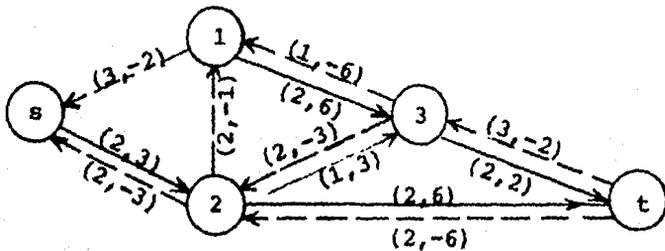


Figura 5.

Red marginal con respecto al flujo definido en la figura anterior.

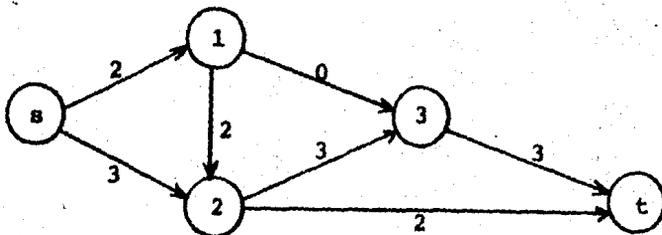


Figura 6.

Nuevo flujo de valor 5 de costo 4,2 (Flujo óptimo)

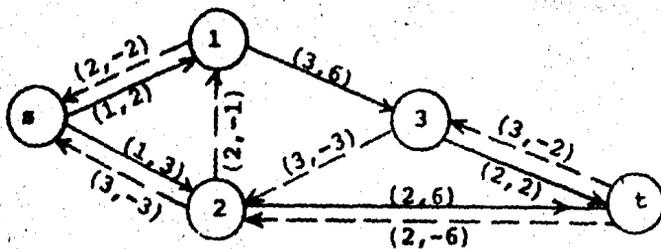
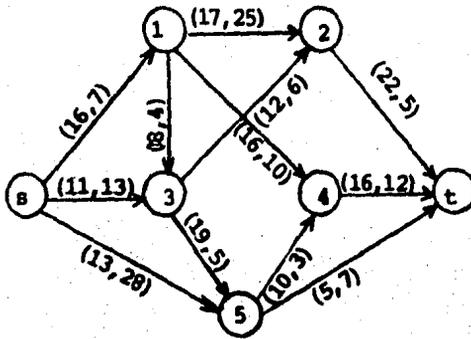


Figura 7.

Red marginal con respecto al flujo definido en la figura anterior.

Ejemplo 2. Determinese el flujo a costo mínimo de valor 20 en la siguiente red mediante el algoritmo basado en eliminación de circuitos negativos.



(capacidad, costo)

Iteración 1. Utilizando el algoritmo de Ford y Fulkerson se determina primero la cadena aumentante  $s, 1, 2, t$  de capacidad incremental igual a 16 unidades y después la cadena  $s, 5, t$  de capacidad incremental igual a 5 unidades por lo que a través de esta última cadena se envían sólo 4 unidades. El flujo inicial resultante se muestra en la figura 8(a). El costo de este flujo es 732. La red marginal con respecto a este flujo se muestra en la figura 8(b) en la cual se ha asociado a cada arco su capacidad y el costo unitario del flujo a través de él, respectivamente. Mediante el algoritmo de Floyd se detecta el circuito negativo  $3, 2, 1, 3$  de costo -15, de donde  $d = \min \{q'_{32}, q'_{21}, q'_{13}\} = 12$ . Actualizando el flujo se obtiene el definido en la figura 8(c) de costo  $732 - 15(12) = 552$ .

Iteración 2. La red marginal con respecto a este nuevo flujo se muestra en la figura 8(d). En ella se ha detectado el circuito negativo  $s, s, 3, 5$  de costo  $-10$ . Entonces  $d = \min \{q'_{5s}, q'_{s3}, q'_{35}\} = 4$ . Actualizando se obtiene el flujo de la figura 8(e) de costo  $552 - 10(4) = 512$ .

Iteración 3. La red marginal se muestra en la figura 8(f) en donde se identifica el circuito negativo  $t, 2, 1, 4, t$  de costo  $-8$ . Luego,  $d = \min \{16, 4, 16, 16\} = 4$ . Actualizando se obtiene el flujo de la figura 8(g) de costo  $512 - 8(4) = 480$ .

Iteración 4. La red marginal resultante se considera en la figura 8(h). En ésta se identifica el circuito negativo  $s, 3, 5, t, 4, 1, s$  de costo  $-4$ . De donde,  $d = \min \{7, 15, 1, 4, 4, 16\} = 1$ . El flujo que se obtiene es el correspondiente a la figura 8(i) de costo  $480 - 4 = 476$ .

Iteración 5. En la figura 8(j) se muestra la nueva red marginal. En ella se detecta un circuito negativo  $s, 1, 3, s$  de costo  $-2$ . Por tanto  $d = \min \{1, 6, 5\} = 1$ . El nuevo flujo, de costo  $476 - 2 = 474$ , se muestra en la figura 8(k).

Iteración 6. La nueva red marginal, figura 8(m), no contiene circuitos negativos por lo cual el flujo definido en la figura 8(k) es el de costo mínimo de valor 20.

Figura 8.

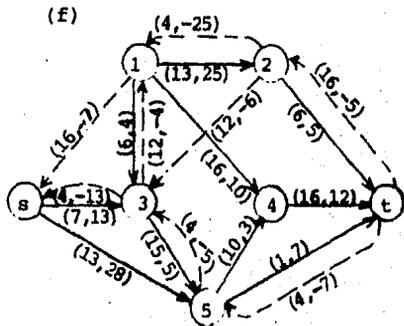
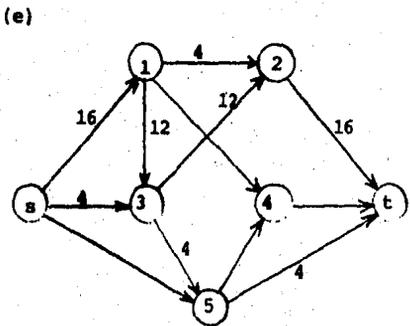
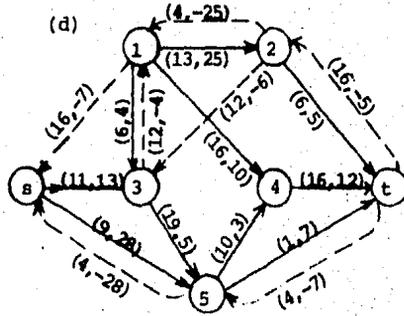
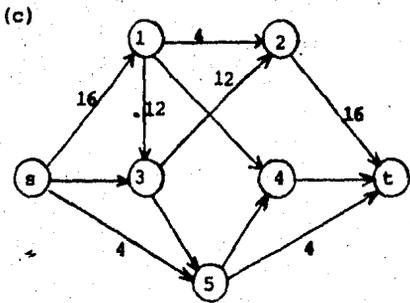
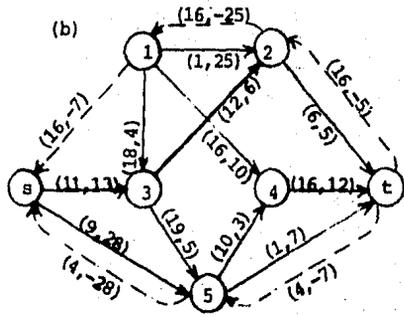
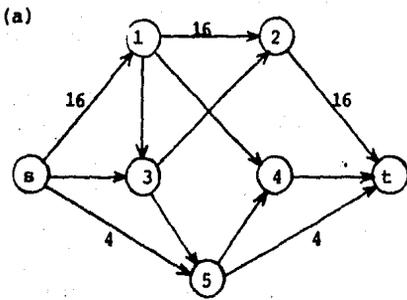
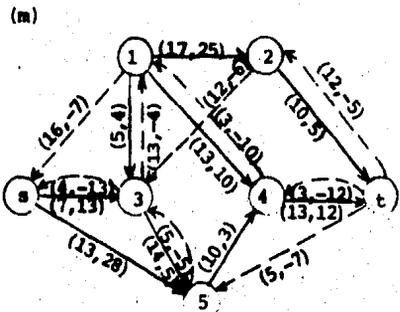
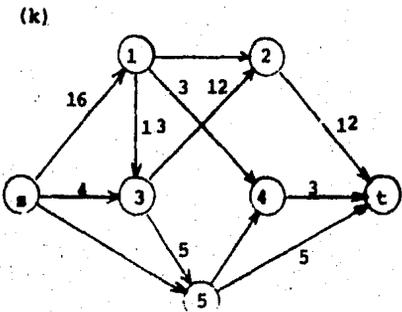
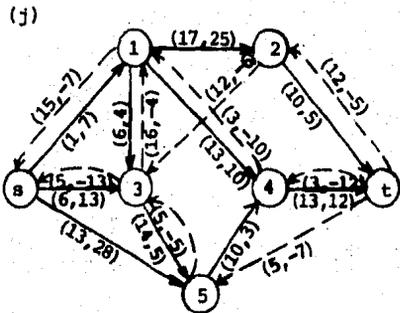
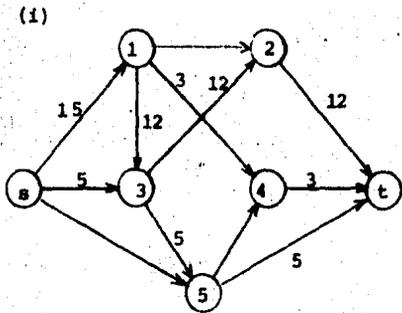
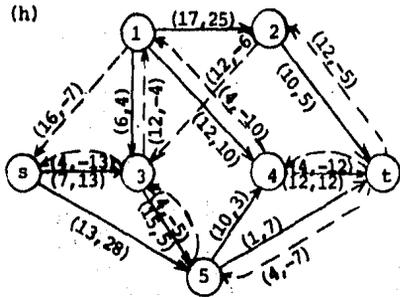
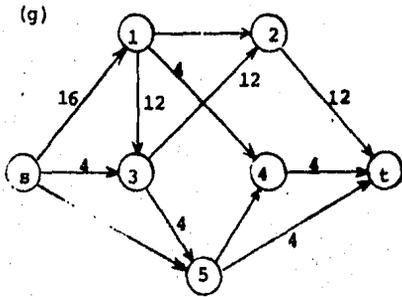


Figura 8. (Cont.)



b. Método basado en rutas más cortas.

Primeramente se determinará un flujo factible de valor  $v'$ , de costo mínimo, donde  $v'$  es menor que el valor  $v$  requerido. En general se utiliza  $v'=0$ . Para determinar este flujo deben eliminarse todos los circuitos negativos que existan en la red marginal. El siguiente paso consiste en determinar la ruta más corta  $P^*$ , de  $s$  a  $t$ , en la red marginal; de nuevo es posible utilizar, con este propósito, el algoritmo general para determinar arborescencias de rutas más cortas presentado en el segundo capítulo. A través de la cadena correspondiente a  $P^*$  se enviará la máxima cantidad posible de  $s$  a  $t$ ; el nuevo flujo así definido será de costo mínimo por el segundo teorema presentado en la sección 4.2. Se procederá de este modo, en cada iteración, hasta alcanzar el valor  $v$  requerido. Aquí hay que hacer algunas observaciones. Si en alguna iteración no existe ruta alguna de  $s$  a  $t$  en la red marginal y el valor  $v'$  del flujo actual  $f$  es menor que  $v$  entonces no existe ningún flujo del valor requerido ya que, al no existir ruta de  $s$  a  $t$  en la red marginal, no existen cadenas aumentantes de  $s$  a  $t$  en la red original y por tanto  $f$  es flujo máximo. Por otro lado, si  $v'$  más la capacidad incremental de la cadena encontrada es mayor que  $v$  entonces sólo será necesario enviar la cantidad  $v-v'$  a través de esta cadena para determinar el flujo deseado. En el algoritmo se denotará con  $f+k$  o  $(P^*)$  al flujo definido en  $R$  de la siguiente manera, donde  $P$  es la cadena correspondiente a  $P^*$ :

$$f_{ij} = \begin{cases} f_{ij} & , \text{ si } (i,j) \notin P \\ f_{ij} + k & , \text{ si } (i,j) \in A_1 \cap P^* \\ f_{ij} - k & , \text{ si } (j,i) \in A_2 \cap P^* \end{cases}$$

A continuación se describe este algoritmo cuya justificación queda dada por el segundo teorema de la sección 4.2.

Propósito. Determinar el flujo a costo mínimo de valor  $v$  en la red  $R = [\bar{X}, A, q, \bar{c}]$ .

Descripción

1. Determinese un flujo factible  $f$  de costo mínimo de valor 0 en  $R$ .
2. Constrúyase la red marginal, con respecto a  $f$ ,  $R'(f)$ .
3. Determinese la ruta más corta  $P^*$  de  $s$  a  $t$  en  $R'(f)$ .
4. Sea  $d = \text{Min}_{(i,j) \in P^*} \{q'_{ij}\}$
5. (i) Si el valor del flujo  $f + d$  o  $(P^*)$  es menor que  $v$ , actualizar  $f = f + d$  o  $(P^*)$  e ir a 2.
- (ii) Si el valor del flujo  $f + d$  o  $(P^*)$  es igual a  $v$ , actualizar  $f = f + d$  o  $(P^*)$  y terminar. En este caso  $f$  es el flujo a costo mínimo de valor  $v$ .
- (iii) Si el valor del flujo  $f + d$  o  $(P^*)$  es mayor que  $v$  y  $v'$  es el valor de  $f$ , actualizar  $f = f + (v-v')o(P^*)$  y terminar ya que  $f$  es el flujo requerido.

Ejemplo 3. Determinése el flujo a costo mínimo de valor 20 en la red del ejemplo 2 utilizando el método basado en rutas más cortas.

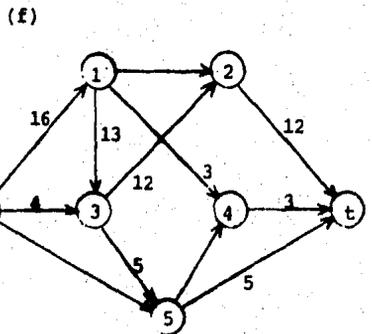
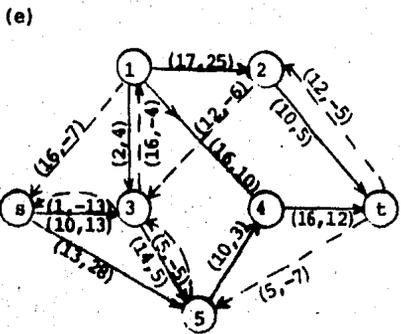
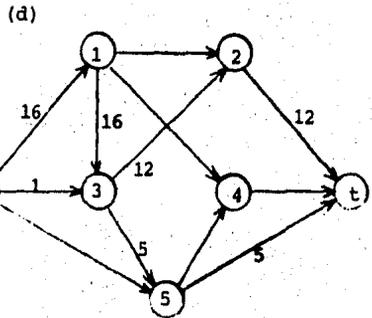
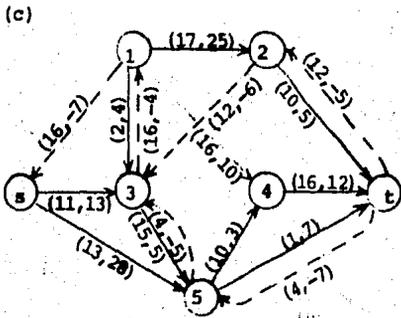
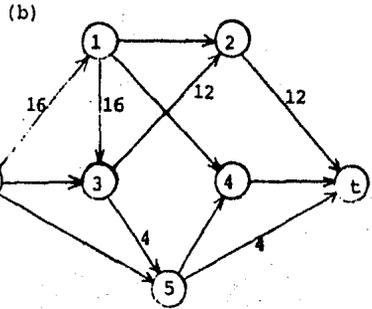
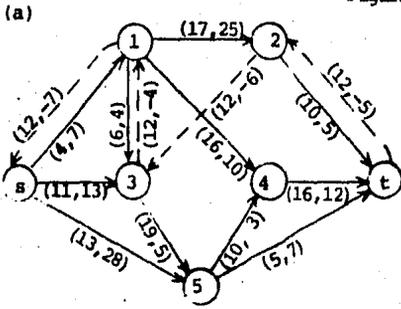
Iteración 1. Si se define flujo igual a 0 a través de todos los arcos de la red se obtiene un flujo de valor 0 de costo mínimo puesto que la red marginal, con respecto a este flujo, coincide con la original y no existen en ella circuitos negativos. Utilizando el algoritmo de Dijkstra se determina la ruta más corta  $P^*: s, 1, 3, 2, t$ , de costo 22. En este caso  $d = \min\{16, 18, 12, 2\} = 12$ . Entonces se actualiza el flujo  $f = f + 12$  o ( $P^*$ ); es decir, se aumentan 12 unidades al flujo a través de los arcos  $(s, 1)$ ,  $(1, 3)$ ,  $(3, 2)$  y  $(2, t)$ .

Iteración 2. La red marginal se muestra en la figura 9(a). Mediante el algoritmo general para determinar arborescencias de rutas más cortas, se obtiene la ruta  $P^*: s, 1, 3, 5, t$  de costo 23. Luego,  $d = \min\{4, 6, 19, 5\} = 4$ . Se actualiza  $f = f + 4$  o ( $P^*$ ) definiéndose, de este modo, el flujo definido en la figura 9(b).

Iteración 3. La figura 9(c) corresponde a la nueva red marginal.  $P^*$  resulta ser:  $s, 3, 5, t$  de costo 25, de donde  $d = \min\{11, 15, 1\} = 1$ . El flujo  $f + 1$  o ( $P^*$ ) se muestra en la figura 9(d).

Iteración 4. La red marginal es la mostrada en la figura 9(e).  $P^*$  es:  $s, 3, 1, 4, t$  de costo 31. De aquí  $d = \min\{10, 16, 16, 16\} = 10$ . Puesto que el valor del flujo actual es  $12 + 4 + 1 = 17$ , sólo se envía la cantidad  $20 - 17 = 3$  a través de esta última cadena; es decir, se actualiza  $f = f + 3$  o ( $P^*$ ). De este modo se obtiene el flujo a costo mínimo deseado que se muestra en la figura 9(f).

Figura 9.



**JJO A COSTO MINIMO CON OFERTAS EN TODOS LOS VERTICES**

**CAPITULO 8**

## 5.1 DESCRIPCION DEL PROBLEMA

Considérese el siguiente problema: Supóngase que en ciertos centros de oferta se dispone de una cantidad limitada de un artículo; este artículo es demandado en ciertos centros de demanda. El costo unitario de transporte del artículo entre cada par de centros es conocido. Se desea transportar el artículo de los centros de oferta a los centros de demanda incurriendo en el menor costo posible suponiendo que la oferta total del artículo es igual a la demanda total del mismo. A este problema puede asociarse la red  $R = [X, A, c, b]$  donde:

- $X$  representa el conjunto de centros de oferta y demanda
- si  $i, j \in X$ ,  $(i, j) \in A$  sólo si es posible transportar el artículo del centro  $i$  al centro  $j$ .
- $c: A \rightarrow \mathbb{R}$  donde, para  $(i, j) \in A$ ,  $c(i, j)$  = costo en el que se incurre por transportar una unidad del artículo del centro  $i$  al centro  $j$ .
- $b: X \rightarrow \mathbb{R}$  donde, para  $i \in X$ ,  $b(i)$  es la cantidad disponible del artículo en el centro  $i$  (si  $b(i) > 0$ ) o la cantidad demandada del artículo en el centro  $i$  (si  $b(i) < 0$ ).

El problema consiste entonces en determinar la cantidad  $x_{ij}$  del artículo a transportar a través del arco  $(i, j)$ , para todo  $(i, j) \in A$ , de tal manera que se incurra en el mínimo costo total de transporte. Obsérvese que, puesto que la oferta total es igual a la demanda total, se tiene que  $\sum_{i \in X} b(i) = 0$ . (Si esto no se cumple, se agrega un vértice  $f$  con oferta  $\sum_{i \in X} b(i) \delta - \sum_{i \in X} b(i)$ ).

En general, a la cantidad  $x_{ij}$  se le llama flujo a través del arco  $(i,j)$  y al problema se le conoce con el nombre de flujo a costo mínimo en la red  $R$ . Por comodidad se denotará  $c(i,j)$  con  $c_{ij}$ , para todo  $(i,j) \in A$ , y  $b(i)$  con  $b_i$ , para todo  $i \in X$ . A esta última cantidad  $b_i$  se le llamará oferta del vértice  $i$ , entendiéndose como demanda una oferta negativa.

El problema de flujo a costo mínimo en una red  $R$  puede plantearse como el problema de programación lineal (P) en donde  $x_{ij}$  es la cantidad de flujo a través del arco  $(i,j) \in A$ .

$$(P): \quad \text{Minimizar } z = \sum_{(i,j) \in A} c_{ij} x_{ij}$$

sujeto a:

$$\sum_{j \in F^+(i)} x_{ij} - \sum_{k \in F^-(i)} x_{ki} = b_i, \text{ para todo } i \in X$$

$$x_{ij} \geq 0, \text{ para todo } (i,j) \in A.$$

Las restricciones del problema se derivan del hecho de que la cantidad total de flujo que sale de un vértice  $i$  debe ser igual a la cantidad que llega a él más la oferta de este vértice. Por otro lado, obsérvese que la matriz de restricciones  $M$  del problema tiene una estructura muy especial. En efecto, cada columna de la matriz tiene exactamente dos elementos distintos de 0: un 1 y un -1. Las componentes asociadas a los vértices  $i$  y  $j$ , de la columna de  $M$  correspondiente a la variable  $x_{ij}$ ,

son 1 y -1 respectivamente puesto que  $x_{ij}$  es el flujo a través del arco  $(i,j)$ ; por esta misma razón, el resto de las componentes son 0. La matriz  $M$  es entonces la matriz de incidencia de la gráfica  $[X,A]$  puesto que cada variable  $x_{ij}$  está asociada al arco  $(i,j) \in A$ . El problema (P) puede por tanto expresarse en forma matricial como:

$$\begin{aligned}
 (P) \quad & \text{Min } z = cx \\
 & \text{s.a.} \\
 & \quad Mx = b \\
 & \quad x \geq 0
 \end{aligned}$$

donde:

- $c$  es el vector  $1 \times m$  de costos unitarios de flujo
- $x$  es el vector  $m \times 1$  de flujos
- $b$  es el vector  $n \times 1$  de ofertas
- $M$  es la matriz  $n \times m$  de incidencia de  $[X,A]$
- $n$  es el número de vértices de la red
- $m$  es el número de arcos de la red

Para resolver el problema de programación lineal (P) puede utilizarse el algoritmo simplex; sin embargo, gracias a la estructura de  $M$ , este algoritmo puede simplificarse en estos casos. Esta simplificación del algoritmo simplex recibe el nombre de método simplex especializado en redes. Antes de exponer este método se analizarán las propiedades de las soluciones básicas de (P) en la siguiente sección.

## 5.2 CARACTERIZACION Y PROPIEDADES DE BASES.

El primer problema que surge cuando se resuelve el problema (P), planteado en la sección anterior, es la identificación de una base de la matriz de restricciones M. En esta sección se pretende caracterizar dichas bases como árboles expandidos, con una cierta característica que se mencionará posteriormente, de la red R  $[X, A, c, q]$ . Para probar esto se utilizarán las propiedades de los árboles expuestas en el primer capítulo, además de otros resultados que a continuación se exponen.

**Definición.** Sea  $C = \{i_1, a_1, i_2, \dots, a_{k-1}, i_k\}$  una cadena de  $i_1$  a  $i_k$ , representada con la secuencia de vértices y arcos que la forman. La orientación de la cadena C es un vector  $O(C) \in R^k$  donde:

$$O_j(C) = \begin{cases} 1 & \text{si } a_j = (i_j, i_{j+1}) \\ -1 & \text{si } a_j = (i_{j+1}, i_j) \end{cases}$$

En términos de la primera sección del tercer capítulo la  $j$ -ésima componente de  $O(C)$ , asociada al arco  $a_j$ , es 1 si  $a_j \in F$  o es -1 si  $a_j \in B$ . De aquí en adelante se utilizarán  $M_j$  para denotar la columna de la matriz de incidencia M de una gráfica asociada al arco  $a_j$ ,  $n$  y  $m$  para denotar el número de vértices y arcos, respectivamente, de la gráfica y  $e^i$  para denotar al vector de dimensión  $n$  cuyas componentes son 0 excepto la  $i$ -ésima que es 1. En base a la anterior definición y a las últimas aclaraciones se tiene la siguiente proposición:

Proposición 1. Sea  $C = \{i_1, a_1, i_2, \dots, a_{k-1}, i_k\}$  una cadena en una gráfica con matriz de incidencia  $M$ . Entonces:

$$\sum_{j=1}^{k-1} O_j(C)M_j = e^{i_1} - e^{i_k}$$

Demostración. Sea  $a_j$  cualquier arco de  $C$ . Se tienen entonces dos casos:

(a)  $a_j = (i_j, i_{j+1})$ . En este caso  $O_j(C) = 1$  y  $M_j = e^{i_j} - e^{i_{j+1}}$ , entonces

$$O_j(C)M_j = e^{i_j} - e^{i_{j+1}}.$$

(b)  $a_j = (i_{j+1}, i_j)$ . En este caso  $O_j(C) = -1$  y  $M_j = e^{i_{j+1}} - e^{i_j}$ , entonces

$$O_j(C)M_j = e^{i_j} - e^{i_{j+1}}.$$

De (a) y (b) se tiene:

$$\sum_{j=1}^{k-1} O_j(C)M_j = e^{i_1} - e^{i_2} + e^{i_2} - e^{i_3} + \dots + e^{i_{k-1}} - e^{i_k} = e^{i_1} - e^{i_k}.$$

con lo que queda probada la proposición. ▀

Intuitivamente este resultado puede interpretarse del siguiente modo: "Ir" del vértice  $i_1$  al vértice  $i_k$  a través de la cadena  $C$  es lo mismo que "ir" de  $i_1$  a  $i_k$  directamente a través del arco  $(i_1, i_k)$ ; es decir, este último arco (cuya columna en la matriz de incidencia puede expresarse como  $e^{i_1} - e^{i_k}$ ) puede escribirse, si existe, como "combinación lineal" de los arcos de la cadena  $C$ .

Obsérvese que las bases de  $M$  corresponden a un conjunto de arcos de la gráfica correspondiente. Este conjunto de arcos no puede formar ciclos; para probar esto se demostrará, en el corolario 3, que las columnas de  $M$  asociadas con un ciclo son linealmente dependientes para lo cual la proposición 1 será de utilidad.

Proposición 2. Sea  $C = \{i_1, a_1, \dots, a_{k-1}, i_k\}$  un ciclo en una gráfica con matriz de incidencia  $M$ . Entonces:

$$\sum_{j=1}^{k-1} O_j(C)M_j = 0$$

Demostración. Puesto que  $C$  es un ciclo se tiene que  $i_1 = i_k$ . Entonces  $e^{i_1} - e^{i_k} = 0$  con lo que queda demostrada la proposición. ■

Corolario 3. Sea  $C = \{i_1, a_1, \dots, a_{k-1}, i_k\}$  un ciclo en una gráfica con matriz de incidencia  $M$ . Entonces  $\{M_j | a_j \in C\}$  es un conjunto de vectores linealmente dependientes.

Demostración. Se exhibe una combinación lineal de estos vectores igual al vector 0 en la proposición 2. ■

Una vez que se ha mostrado que las bases no pueden corresponder a arcos que forman ciclo, se probará que un árbol puede corresponder a parte de una base.

Proposición 4. Sea  $M$  la matriz de incidencia de una gráfica y sea  $T = [X', A']$  un árbol con  $r$  vértices de la gráfica. Supóngase que  $r \geq 2$ . Entonces  $\{M_j | a_j \in A'\}$  es un conjunto de vectores linealmente independientes.

Demostración. Por inducción se tiene que, si  $m$  es el número de arcos de  $T$ :

Para  $r=2$ ,  $m=1$  por lo que es válida la afirmación.

Supóngase válida la afirmación para  $r=k-1$ .

Ahora se probará para  $r=k$ .

Supóngase que, para  $r=k$ , los vectores del conjunto  $\{M_j | a_j \in A'\}$  son linealmente dependientes. Entonces existen  $v_1, v_2, \dots, v_k \in \mathbb{R}$  tales que:

$$\sum_{j=1}^k v_j M_j = 0, \text{ con } v_i \neq 0, \text{ para alguna } i \in \{1, \dots, k\}$$

Por otro lado, por la proposición 4 del primer capítulo, el árbol  $T$  tiene al menos dos vértices pendientes. Sea  $p$  uno de estos vértices y sea  $a_w$  el único arco adyacente a  $p$  en  $T$ . Sea  $M_{pj}$  la componente de  $M_j$  correspondiente al vértice  $p$ . Puesto que en  $T$  el grado de  $p$  es 1, se tiene que:

$$M_{pj} = \begin{cases} 0 & \text{si } j \in \{1, \dots, k\}, j \neq w \\ +1 & \text{si } j = w \end{cases}$$

De aquí:

$$\sum_{j=1}^k v_j M_{pj} = v_w M_{pw} + \sum_{j \neq w} v_j M_{pj} = v_w M_{pw} = 0$$

luego,  $v_w = 0$ . Ahora, por la proposición 5 del primer capítulo, la gráfica  $T' = [X' - \{p\}, A' - \{w\}]$  es un árbol con  $k-1$  vértices; las columnas de  $M$  asociadas con los arcos de  $T'$  son  $\{M_j | a_j \in A' - \{w\}\}$ ; estos vectores son linealmente dependientes ya que

$$\sum_{j=1}^k v_j M_j = 0, \text{ con } v_i \neq 0, \text{ para alguna } i \in \{1, \dots, k\}, i \neq w$$

Esto contradice la hipótesis de inducción y por tanto se concluye que la afirmación es válida para toda  $r$ . ■

Otro punto importante a considerar es el rango de la matriz  $M$ . Es claro que esta matriz no tiene rango completo puesto que sumando sus renglones se obtiene el vector  $0$ . De hecho, el rango de  $M$  es exactamente  $n-1$ . Esto se prueba en la siguiente proposición.

Proposición 5. Sea  $M$  la matriz de incidencia de una gráfica conexa con  $n$  vértices. Entonces el rango de  $M$  es  $n-1$ .

Demostración. Puesto que  $M$  no es de rango completo, entonces es menor o igual que  $n-1$ . Por otro lado, por el corolario 6 del primer capítulo, existe un árbol expandido  $T$  de la gráfica. Entonces, por la proposición anterior, las  $n-1$  columnas de  $M$  asociadas con los arcos de  $T$  son linealmente independientes. Por lo tanto se concluye que el rango de esta matriz es mayor o igual que  $n-1$ . Luego, dicho rango es exactamente  $n-1$ . ■

El siguiente resultado garantiza que el conjunto de arcos, de la gráfica en cuestión, asociados con  $n-1$  columnas linealmente independientes de  $M$  forman un árbol expandido.

Proposición 6. Sea  $M$  la matriz de incidencia de una gráfica  $G = [X, A]$  con  $n$  vértices. Sea  $A' \subset A$  tal que  $\{M_j \mid a_j \in A'\}$  es un conjunto de vectores linealmente independientes y supóngase que  $A'$  consta de  $n-1$  arcos. Entonces la gráfica  $T = [X, A']$  es un árbol expandido de  $G$ .

Demostración. Puesto que los vectores de este conjunto son linealmente independientes entonces, por el corolario 3, los arcos asociados con ellos no forman ciclo. Entonces la gráfica parcial  $T$  de  $G$  es acíclica y tiene  $n-1$  arcos; luego, por el teorema 3 del primer capítulo,  $T$  es un árbol expandido de  $G$ . ■

Ahora que se han establecido estos resultados se cuenta con las herramientas necesarias para caracterizar las soluciones básicas del problema de programación lineal (P). Para resolver este problema se completará el rango de la matriz de restricciones  $M$  agregando una variable artificial  $x_h$  ( $h \in \{1, \dots, n\}$ ). Reformulando (P) se obtiene:

$$(P) \quad \text{Min } z = cx$$

s.a.

$$Mx + x_h e^h = b$$

$$x \geq 0$$

$$0 \leq x_h \leq 0$$

La matriz de restricciones es ahora  $[M \ e^h]$  la cual se probará enseguida, es de rango completo.

Proposición 7. Sea  $M$  la matriz de incidencia de una gráfica  $G = [X, A]$  con  $n$  vértices y sea  $T = [X, A']$  un árbol expandido de  $G$ . Sea  $h \in \{1, \dots, n\}$ . Entonces:

$$\Omega = (M_j \mid a_j \in A') \cup \{e^h\} \text{ expande } \mathbb{R}^n.$$

Demostración. Basta demostrar que los vectores de la base canónica en  $\mathbb{R}^n$ :  $e^1, e^2, \dots, e^n$  pueden expresarse como combinación lineal de elementos de  $\Omega$ . Esto es claro para  $e^h$ . Sea  $p \in \{1, \dots, n\}$  tal que  $p \neq h$ . Puesto que  $T$  es un árbol, existe una única cadena entre los vértices  $p$  y  $h$  en  $T$ . Sea  $C = \{p=i_1, a_1, \dots, a_{k-1}, i_k=h\}$  esta cadena. De la proposición 1 se tiene:

$$\sum_{j=1}^{k-1} O_j(C)M_j = e^p - e^h,$$

entonces

$$e^p = \sum_{j=1}^{k-1} O_j(C)M_j + e^h,$$

de donde se concluye la validez de la afirmación. ■

Es importante contar con una interpretación gráfica del vector  $e^h$  para poder caracterizar las bases del problema. Para ello se definen los siguientes conceptos.

Definición. Un arco raíz en una gráfica  $G$  es un arco que tiene extremo inicial pero no tiene extremo final. Al vértice inicial de un arco raíz se le llama vértice enraizado. Si  $h$  es un vértice enraizado a  $G$  se le llama gráfica enraizada en el vértice  $h$ .

Obsérvese que la columna de la matriz de incidencia asociada con un vértice enraizado  $h$  es el vector canónico  $e^h$ .

En la proposición 7 se prueba que las columnas de la matriz de incidencia  $[M e^h]$  de una gráfica enraizada en el vértice  $h$ , asociadas con un árbol expandido enraizado en  $h$ , forman una base del problema. El inverso también es válido y se prueba en la siguiente proposición:

Proposición 8. Sea  $M$  la matriz de incidencia de una gráfica  $G = [X, A]$  y sea  $h \in X$ . Supóngase que  $G$  se enraiza en  $h$ . Si  $\Omega$  es una base para  $[M e^h]$  entonces  $e^h \in \Omega$  y  $T = [X, A']$ , donde  $A' = \{a_j \in A \mid M_j \in \Omega\}$  es un árbol expandido de  $G$ .

Demostración. Por la proposición 7 se tiene que  $[M e^h]$  tiene rango completo; pero por la proposición 5 la matriz  $M$  no tiene rango completo. Luego, se concluye que  $e^h \in \Omega$ . De aquí que  $A'$  corresponde a  $n-1$  columnas linealmente independientes de  $M$ ; por lo tanto, de la proposición 6 se concluye que  $T = [X, A']$  es un árbol expandido de  $G$  con lo que queda demostrada la proposición. ■

De este modo, las bases del problema han quedado completamente caracterizadas como árboles expandidos enraizados de la gráfica. Se resume esto en el siguiente teorema.

Teorema 9. Sea  $M$  la matriz de incidencia de una gráfica  $G = [X, A]$ . Sea  $a_h$  un arco raíz en el vértice  $h$ . Las únicas bases para  $[M e^h]$  son  $e^h$  junto con un conjunto de columnas de  $M$  que corresponden a un árbol expandido de  $G$ .

Demostración. Queda dada por las proposiciones 7 y 8. ■

Por último se probará una propiedad muy importante de las bases del problema. Esta propiedad será muy útil para el cálculo de los valores de las variables básicas y de las variables duales.

Proposición 10. Sea  $M$  la matriz de incidencia de una gráfica  $G = [X, A]$  y sea  $a_h$  un arco raíz en el vértice  $h$ . Sea  $B$  una base de  $[M \ e^h]$ ; entonces  $B$  es triangularizable.

Demostración. Sean  $T = [X, A^*]$  el árbol expandido enraizado de  $G$  correspondiente a  $B$  y sean  $h$  y  $a_h$  el vértice enraizado y el arco raíz, respectivamente. Por la proposición 4 del primer capítulo se afirma que  $T$  tiene, al menos, dos vértices pendientes; sea  $p_1$  uno de ellos ( $p_1 \neq h$ ) y sea  $a_1$  el único arco adyacente a él. Obsérvese que el renglón de  $B$  correspondiente a  $p_1$  contiene un sólo elemento distinto de 0 (el correspondiente a  $a_1$ ); entonces, si se intercambian el primer renglón y la primera columna por el renglón de  $p_1$  y la columna de  $a_1$ , se obtiene:

$$\begin{bmatrix} \pm 1 & & 0 & & 0 \\ & & & & \\ & & & & \\ \pm e^r & & & & B_1 \\ & & & & \end{bmatrix}$$

donde  $r$  es el otro extremo de  $a_1$ . Sea  $T_1 = [X - \{p_1\}, A^* - \{a_1\}]$ ; entonces, por la proposición 5 del primer capítulo,  $T_1$  es un árbol con  $n-1$  vértices. Si  $n-1=1$ , entonces  $B_1$  es una matriz  $1 \times 1$  y por tanto  $B$  es triangularizable. Supóngase que  $n-1 > 1$ ; sea  $p_2$  un vértice pendiente de  $T_1$  y sea  $a_2$  el único arco adyacente a él.

Obsérvese que el renglón de  $B_1$  correspondiente a  $p_2$  tiene un sólo elemento distinto de 0 (el correspondiente a  $a_2$ ); intercambiando el segundo renglón y la segunda columna de B por el renglón de  $p_2$  y la columna de  $a_2$  se obtiene:

$$\begin{bmatrix} \underline{+1} & 0 & 0 & 0 \\ & \underline{+1} & 0 & 0 \\ \underline{+e^r} & & & \\ & \underline{+e^s} & & B_2 \end{bmatrix}$$

donde  $s$  es el otro extremo de  $a_2$ . Este procedimiento se repite exactamente  $n$  veces. Luego, B es triangularizable. Obsérvese también que el último renglón y la última columna de la matriz B triangularizada corresponden a  $h$  y  $a^h$  respectivamente.

Una vez que se han caracterizado las bases y se ha probado que son triangularizables, se cuenta con lo necesario para presentar el método de solución.

### 5.3 METODO SIMPLEX ESPECIALIZADO EN REDES.

Primeramente se exponen los pasos generales del método primal simplex para después analizar la especialización de cada uno en el caso de redes: Primero se encuentra una solución básica factible inicial; después se calculan los coeficientes de costo reducido para probar la optimalidad de la solución. Si esta solución no es óptima se procede a cambiarla determinando la variable que entra a la base y la que sale de ella. Con esta nueva solución básica factible se realiza otra iteración.

Puesto que, en el caso del problema de flujo a costo mínimo en una red, las bases corresponden a árboles expandidos enraizados de ésta, cambiar de base significará cambiar de árbol. Más aún, dadas las características de las bases, no será necesario calcular la inversa de éstas; debido a que son triangularizables se podrán calcular eficientemente los valores de las variables básicas y los coeficientes de costo reducido. El problema de determinar una solución básica inicial se tratará más adelante.

#### Cálculo de los valores de las variables básicas.

Sean  $R = [X, A, c, b]$  una red en la cual se desea determinar el flujo a costo mínimo. Sea  $B$  una base de  $[M \ e^h]$  y sea  $x_B$  el vector  $n \times 1$  de las variables básicas correspondientes a  $B$ ; es decir,  $x_B$  es el vector de flujos a través de los arcos del árbol expandido  $T$  enraizado en  $h$  asociado con  $B$ . Entonces se sabe que  $Bx_B = b$ . Aprovechando que  $B$  es triangularizable, este sistema se resuelve fácilmente por substitución. Sin embargo,

la solución puede inferirse directamente sobre la red. En efecto, un renglón de la matriz B triangularizada corresponde a un vértice pendiente p del árbol T' resultante de eliminar de T todos los vértices correspondientes a renglones anteriores al de p junto con los arcos adyacentes a ellos. Por tanto, existe un sólo arco w adyacente a p en T'; puede verificarse fácilmente que el valor de la variable básica  $x_w$  (flujo a través del arco w) es:

$$x_w = b_p + \sum_{j \in \Gamma^-(p)} x_{jp} - \sum_{\substack{i \in \Gamma^+(p) \\ i \neq k}} x_{pi} \quad \text{si } w = (p, k) ;$$

o es el recíproco de esta cantidad si  $w=(k,p)$ . Obsérvese que la componente  $(p,w)$  de la matriz es igual a +1 en el primer caso y a -1 en el segundo caso.

### Cálculo de los coeficientes de costo reducido

El siguiente punto a considerar es el cálculo del vector  $\bar{c} = c_B B^{-1} M - c$  (donde  $c_B$  es el vector de costos de  $x_B$ ) de coeficientes de costo reducido. Primeramente se calculará el vector  $w = c_B B^{-1} c \in \mathbb{R}^n$  que es el vector de variables duales (del problema (P)) también conocido como vector de potenciales en los vértices. En efecto, el problema dual de (P) es:

$$(D) \quad \text{Max } z' = wb$$

s.a.

$$w [M \quad e^h] \leq [c, 0]$$

en el cual es sencillo comprobar que  $w = c_B B^{-1}$  es factible si  $\bar{c} \leq 0$  (es decir, si B es base óptima para (P)).

Puesto que la columna de B, correspondiente al arco (i,j) del árbol T expandido enraizado en h asociado a B, puede expresarse como  $e^i - e^j$ , para calcular  $w = c_B B^{-1}$  será más fácil resolver el sistema  $wB = c_B$ . En este sistema se tiene una ecuación por cada variable básica. La ecuación correspondiente a la variable básica  $x_{ij}$ , equivalentemente al arco (i,j) de T, es:

$$w_i - w_j = c_{ij}$$

y la ecuación correspondiente al arco raíz  $a_h$  es:

$$w_h = 0$$

Entonces este sistema puede resolverse fácilmente sobre la red, asociando a cada vértice la correspondiente variable dual.

Se empieza calculando  $w_h = 0$ ; es decir, el potencial del vértice enraizado h. De aquí se calcula el potencial de un vértice, extremo de algún arco del árbol T, cuando se haya calculado el potencial del otro extremo del arco a través de la relación  $w_i - w_j = c_{ij}$ . Una vez calculadas las variables duales se procede al cálculo de  $\bar{c}$ . La componente (i,j) de este vector es:

$$\bar{c}_{ij} = w(e^i - e^j) - c_{ij} = w_i - w_j - c_{ij};$$

obsérvese que para las variables básicas (es decir, para los arcos de T),  $\bar{c}_{ij} = 0$ . Basta entonces calcular  $\bar{c}_{ij}$  para las variables no básicas (es decir, para los arcos que no pertenecen a T).

Si  $\bar{c}_{ij} \leq 0$ , para toda variable  $x_{ij}$  no básica, se tiene que la solución actual es óptima; en caso contrario se procede a cambiar de solución básica factible.

#### Determinación del arco que entra a la base.

Los arcos del árbol T expandido enraizado en h son llamados arcos básicos; los arcos que no pertenecen a T son llamados arcos no básicos. Determinar la variable que entra a la base significa entonces determinar el arco no básico que debe ser considerado como parte del nuevo árbol expandido enraizado. Este arco puede ser cualquiera para el cual  $\bar{c}_{ij}$  sea mayor que 0.

#### Determinación del arco que sale de la base.

Al agregar el arco que entra a la base al árbol T, se forma un único ciclo. Para obtener un nuevo árbol debe entonces eliminarse dicho ciclo. Entonces, determinar la variable que sale de la base significa determinar el arco básico del ciclo que debe ser eliminado. Si un arco se convierte en básico aumentará el valor del flujo a través de él; luego, el flujo a través de los arcos del ciclo formado deberá ser reajustado para conservar la factibilidad. Aquel arco para el cual el nuevo flujo sea igual a 0 es el que debe dejar la base. Si C es la cadena (única en T) de x a y que forma ciclo con el arco (x,y) que entra a la base y D es la cantidad que aumenta el flujo a través de éste arco, es claro que el nuevo flujo a través de esos arcos de C es:

$$x_j + D \quad , \quad \text{si } O_j(C) = -1$$

$$x_j - D \quad , \quad \text{si } O_j(C) = +1$$

donde  $D \leq x_j$ , para todo arco  $j \in C$  tal que  $O_j(C) = +1$ .

El valor máximo de  $D$  es por tanto:

$$\min \{x_j \mid j \text{ es un arco de } C \text{ y } O_j(C) = -1\}$$

El arco que se elimina de  $T$  es entonces aquél para el cual  $x_j = D$ , cuando  $D$  toma su máximo valor. De este modo, se determina un nuevo árbol  $T'$  expandido enraizado en  $h$  y se calculan los nuevos valores de las variables básicas sumando o restando  $D$  a los arcos que formaban parte del árbol  $T$  según sea su orientación.

## ALGORITMO SIMPLEX ESPECIALIZADO EN REDES

Propósito: Determinar el flujo a costo mínimo en la red

$R = [X, A, c, b]$ .

### Descripción

1. Determínese  $T = [X, A']$  un árbol expandido de  $R$  enraizado en  $h$ , correspondiente a la solución básica factible  $x$  del problema. Sea  $B$  la matriz triangular asociada a  $T$  y sea  $x_B$  el vector de flujos a través de los arcos básicos. Calcúlese las componentes de  $x_B$  resolviendo el sistema  $Bx_B = b$  sobre la red como se indicó anteriormente.
2. Calcúlese las variables duales resolviendo el sistema  $w_h = 0$  y  $w_i - w_j = c_{ij}$ , para todo arco  $(i, j) \in A'$ .
3. Calcúlese los coeficientes de costo reducido  $\bar{c}_{ij} = w_i - w_j - c_{ij}$ , para todo arco  $(i, j) \notin A'$ .
  - (i) Si  $\bar{c}_{ij} \leq 0$  para todo  $(i, j) \notin A'$ , entonces la solución dada por  $T$  es óptima. Terminar.
  - (ii) Si  $\bar{c}_{ij} > 0$  para algún  $(i, j) \notin A'$ , ir a 4.
4. Determínese el arco que entra a la base. Este es cualquier arco  $(r, s)$  tal que  $\bar{c}_{rs} > 0$ .
5. Determínese el arco que sale de la base. Sea  $C$  la única cadena que une  $r$  y  $s$  en el árbol  $T$ . Calcúlese:

$$D = \min \{x_j \mid O_j(C) = 1\}$$

Sea  $k \in A'$  tal que  $O_k(C) = 1$  y  $x_k = D$ . Entonces  $k$  es el arco que sale de la base.

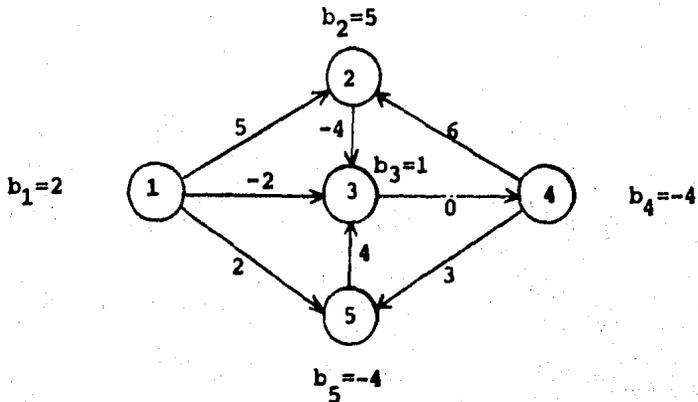
6. Actualícese la solución. Sea  $T' = [X, \{A' - \{(u,v)\} \cup \{(r,s)\}\}]$  el nuevo árbol expandido de  $R$  enraizado en  $h$ .

Defínase:

$$x_i = \begin{cases} x_i + D & , \text{ si } O_i(C) = -1 \quad \text{o } i=(r,s) \\ x_i - D & , \text{ si } O_i(C) = 1 \\ x_i & , \text{ si } i \notin C \end{cases}$$

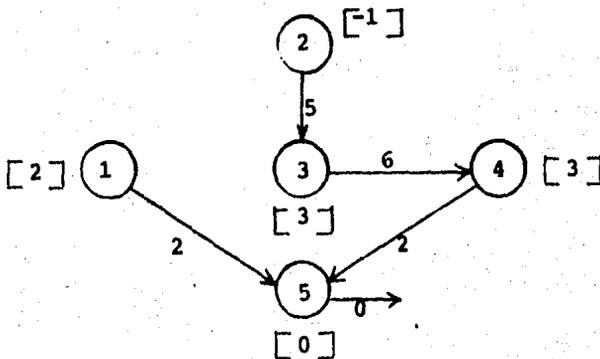
Con este nuevo flujo regrésese a 2.

Ejemplo 1. Determínese el flujo a costo mínimo en la siguiente red mediante el algoritmo simplex especializado en redes. Considérense como arcos básicos iniciales: (1,5), (2,3), (3,4) y (4,5).



El número asociado a cada arco es el costo unitario del flujo a través de él.

Iteración 1. Si se enraiza la red en el vértice 5, se obtiene el siguiente árbol  $T=(X,A')$  correspondiente a la primera base.



El número asociado a cada arco es el valor del flujo a través de él y el asociado a cada vértice es su potencial. El flujo a través de cada arco se calculó de la siguiente manera:

Los vértices 1 y 2 son pendientes por lo cual

$$x_{15} = b_1 = 2 \quad \text{y} \quad x_{23} = b_2 = 5$$

Al eliminar los vértices 1 y 2 y los arcos (1,5) y (2,3), en el árbol resultante, el vértice 3 es pendiente, entonces

$$x_{34} = b_3 + x_{23} = 5 + 1 = 6$$

Al eliminar el vértice 3 y el arco (3,4) el vértice 4 resulta pendiente, luego:

$$x_{45} = b_4 + x_{34} = -4 + 6 = 2$$

y finalmente  $x_5 = b_5 + x_{15} + x_{45} = -4 + 2 + 2 = 0$ .

Los potenciales de los vértices se calcularon:

$$w_5 = 0$$

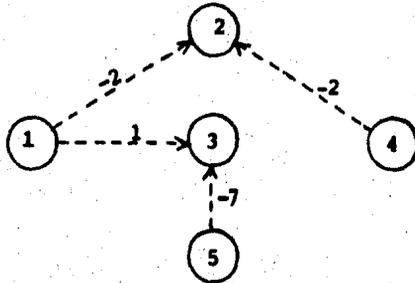
$$w_4 = w_5 + c_{45} = 0 + 3 = 3$$

$$w_1 = w_5 + c_{15} = 0 + 2 = 2$$

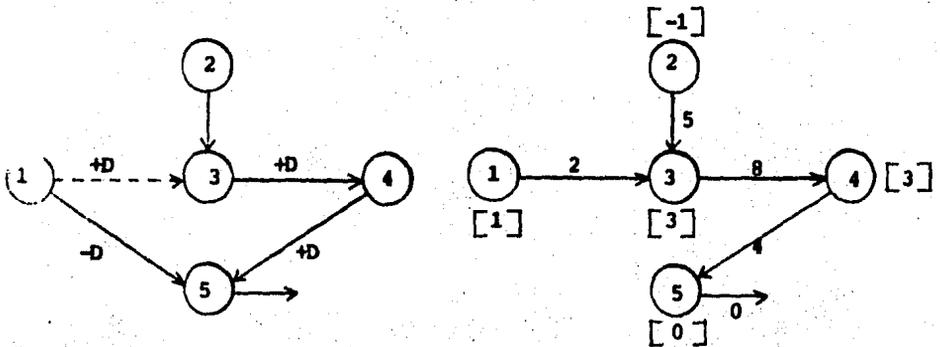
$$w_3 = w_4 + c_{34} = 3 + 0 = 3$$

$$w_2 = w_3 + c_{23} = 3 - 4 = -1$$

En la siguiente figura se asocia a cada arco no básico su coeficiente de costo reducido.

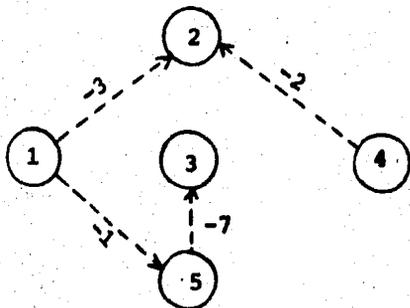


Puesto que  $\bar{c}_{13} = 1 > 0$  entonces el arco (1,3) entra a la base. Al agregar a T este arco se forma un ciclo con la cadena 1,5,4,3, que une sus extremos. En este caso  $D = \min(x_{15}) = 2$  puesto que éste es el único arco con orientación igual a 1. Por tanto este arco es el que sale de la base.



Al agregar el arco (1,3) y eliminar el arco (1,5) se obtiene el árbol de la figura anterior, se actualiza el flujo como se indica en la primera figura puesto que la orientación de la cadena 1,5,4,3, es el vector (1,-1,-1). Este flujo actualizado se asocia a cada arco del árbol resultante.

Iteración 2. También se asocia a cada vértice del árbol anterior su potencial que fué calculado resolviendo el sistema  $w_5=0$  y  $w_i - w_j = c_{ij}$ , para todo arco (i,j) básico. A continuación se asocia a cada arco su correspondiente coeficiente de costo reducido.



Puesto que todos los coeficientes de costo reducido son menores que 0, la solución actual es la óptima; es decir, el flujo a costo mínimo en la red está dado por:

$$x_{12}=0, x_{13}=2, x_{15}=0, x_{23}=5, x_{34}=8, x_{42}=0, x_{45}=4 \text{ y } x_{53} = 0.$$

### Solución inicial.

Para determinar una solución inicial para el problema de flujo a costo mínimo en la red  $R = [X, A, c, b]$  con  $n$  vértices, se utilizará una especialización del método de las dos fases. Para ello agréguese un nuevo vértice  $f$  a la red y nuevos arcos de la forma  $(i, f)$ , si  $b_i \geq 0$ , y  $(f, i)$ , si  $b_i < 0$ . Considérese la red  $R' = [XU\{f\}, AUA', c', b']$  donde:

$$A' = \{(i, f) \mid b_i \geq 0\} \cup \{(f, i) \mid b_i < 0\}$$

$$c'_{ij} = \begin{cases} 1, & \text{si } (i, j) \in A' \\ 0, & \text{si } (i, j) \in A \end{cases}$$

$$b'_i = \begin{cases} b_i, & \text{si } i \in X \\ 0 & \text{si } i = f \end{cases}$$

Entonces la primera fase del método consiste en resolver el problema de flujo a costo mínimo en la red  $R'$ . Una solución básica para este nuevo problema corresponde a un árbol expandido, en raizado en algún vértice, de la gráfica  $[XU\{f\}, AUA']$ . Es fácil verificar que los arcos del conjunto  $A'$  junto con algún arco raíz, forman una base factible. Los valores de los flujos a través de los arcos de  $A'$  son:

$$x_{if} = b_i, \text{ para todo } i \in X \text{ tal que } b_i \geq 0$$

$$x_{fi} = -b_i, \text{ para todo } i \in X \text{ tal que } b_i < 0$$

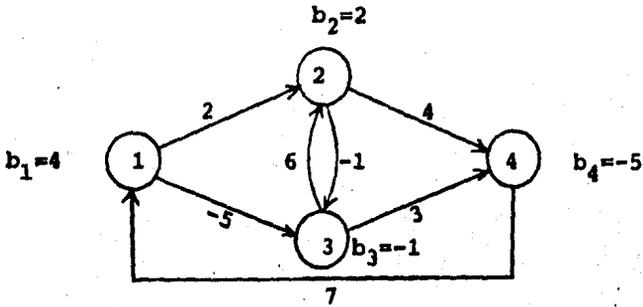
y el flujo a través del arco raíz es igual a 0.

Una vez determinada la solución óptima para la red  $R'$  pueden presentarse dos casos:

Caso a. La solución óptima tiene costo 0. En este caso un árbol óptimo  $T'$  consta de  $n-1$  arcos de la red original, un arco raíz y un arco de  $A'$ , donde el flujo a través de este último es igual a 0. Los  $n-1$  arcos de la red  $R$ , junto con algún arco raíz, forman una base factible del problema original. En efecto, el vértice  $f$  es pendiente en  $T'$  y el único arco adyacente a él es el que pertenece a  $A'$ , entonces al eliminarlos de  $T'$  se obtiene un árbol  $T$  expandido de  $R$ . El flujo definido en los arcos de  $T$  es factible en  $R$  ya que  $b'_f = 0$ . Con el árbol  $T$ , enraizado en algún vértice, se inicia la segunda fase del método que consiste en determinar el flujo a costo mínimo en la red  $R$ .

Caso b. La solución óptima tiene costo distinto de 0. En este caso el árbol óptimo  $T'$  contiene algún arco de  $A'$  a través del cual el flujo es distinto de 0. Claramente, cuando esto sucede, no existe solución factible alguna para el problema original y por tanto se da por terminado el método.

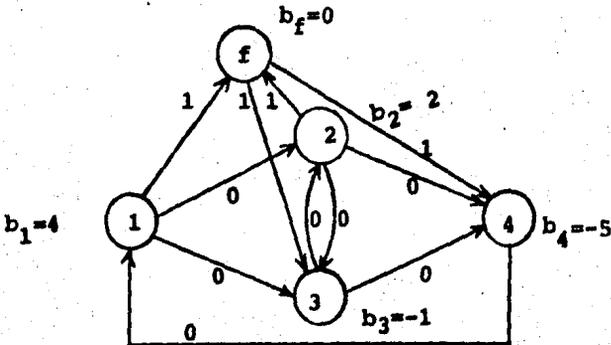
Ejemplo 2. Determinese el flujo a costo mínimo en la siguiente red utilizando el método simplex de las dos fases especializado en redes.



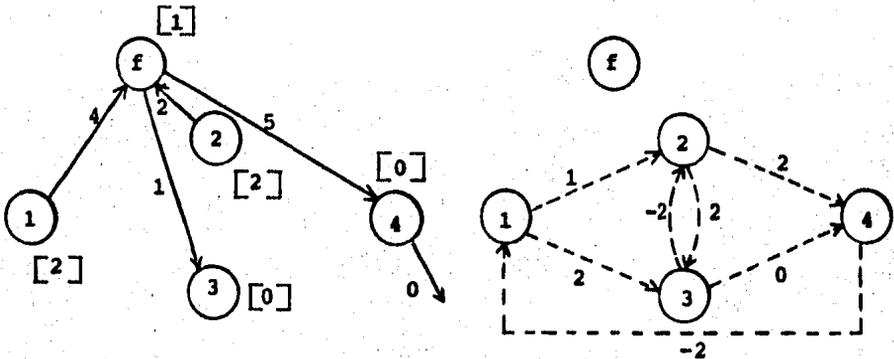
El número asociado a cada arco es el costo unitario del flujo a través de él.

Primera Fase.

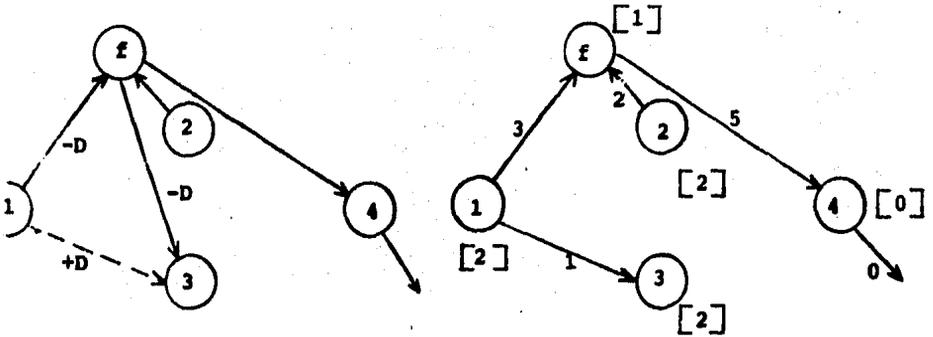
Iteración 1. Puesto que no se cuenta con una base inicial factible se construye la red  $R'$  que a continuación se muestra con el costo unitario de flujo asociado a cada arco.



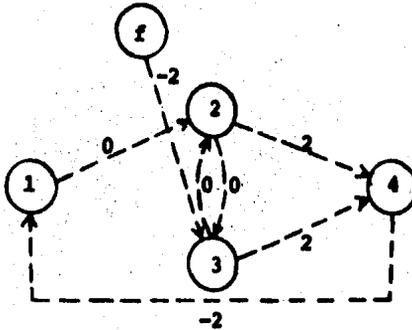
El árbol básico factible inicial en la primera fase, si se enraiza la red en el vértice 4, es la primera red de la siguiente figura. En él se asocia cada arco el flujo a través de él y a cada vértice su potencial. En la segunda red se asocia a cada arco su coeficiente de costo reducido.



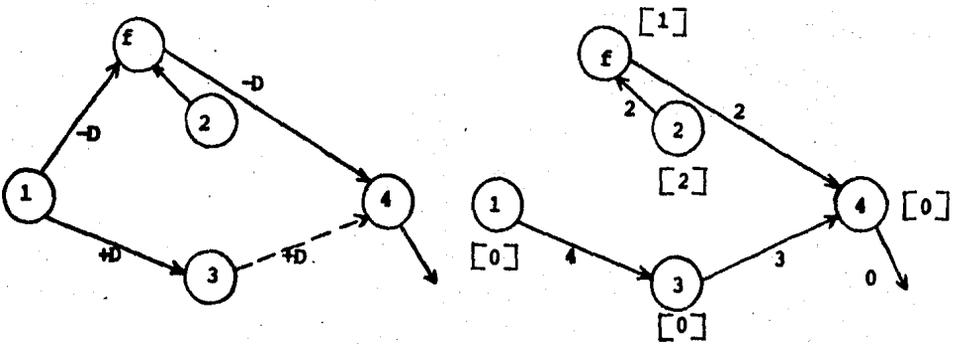
Puesto que  $(1,2)$ ,  $(1,3)$ ,  $(2,3)$  y  $(2,4)$  tienen coeficiente de costo reducido mayor que 0, cualquiera de estas variables puede entrar a la base. Elíjase el arco  $(1,3)$  para entrar a la base. Entonces la cadena que forma ciclo con este arco es 1, f, 3 con orientación  $(1,1)$  por lo que  $D = \min\{x_{1f}, x_{f3}\} = 1$ ; de donde el arco que sale de la base es  $(f,3)$ . Se actualiza el flujo como se muestra en la primera red de la siguiente figura obteniéndose el definido en el nuevo árbol básico que se muestra enseguida.



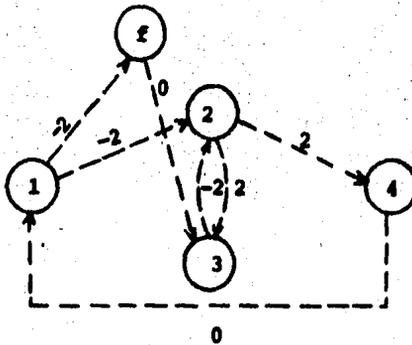
Iteración 2. En el árbol anterior se ha asociado a cada vértice la correspondiente variable dual. De nuevo, se calculan los coeficientes de costo reducido de los arcos no básicos.



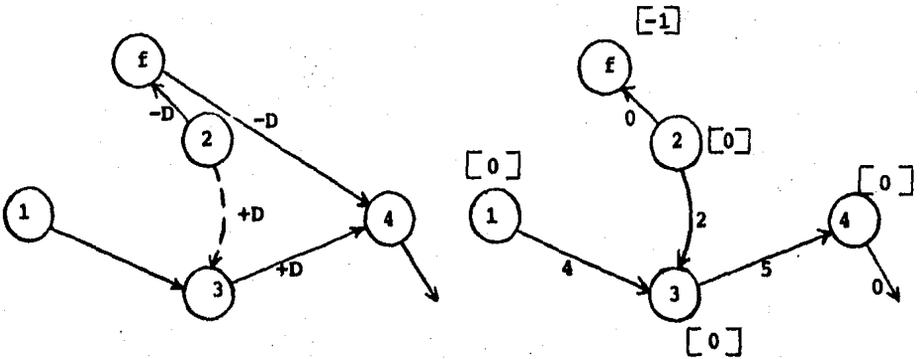
Se elige el arco (3,4) para que entre a la base puesto que "rompe" la optimalidad. La cadena con la que forma ciclo este arco es 3,1,f,4 de orientación (-1,1,1); de donde  $D = \min(x_{1f}, x_{f4}) = 3$  y por tanto (1,f) sale de la base. Actualizando el flujo como se indica en la primera red de la siguiente figura se obtiene el definido en el árbol resultante.



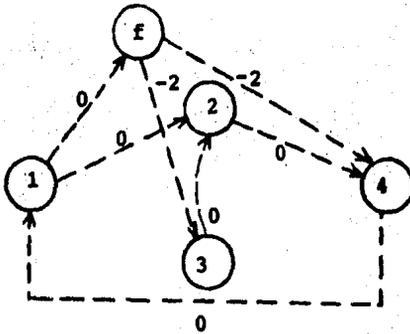
Iteración 3. En el último árbol se muestra, junto a cada vértice  $i$ , la variable dual  $w_i$ . Los coeficientes de costo reducido son:



Elijase  $(2,3)$  para que entre a la base. La cadena con la que forma ciclo es  $2, f, 4, 3$  de orientación  $(1, 1, -1)$ ; luego  $D = \min(x_{2f}, x_{f4}) = 2$  y por tanto  $(2, f)$  o  $(f, 4)$ , sale de la base. Actualizando el flujo se obtiene el definido en el nuevo árbol básico.



Iteración 4. Los coeficientes de costo reducido de los arcos no básicos, calculados en base a las variables duales que se muestran en el último árbol son:

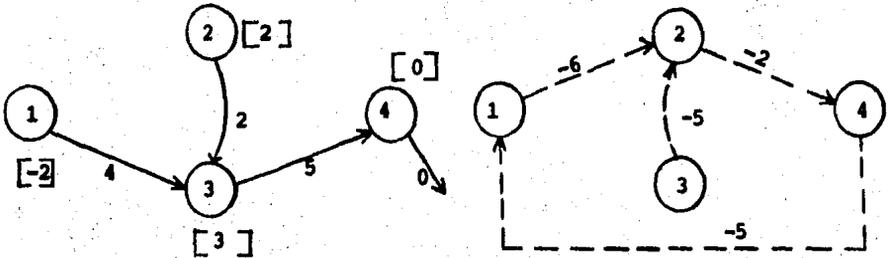


Puesto que  $\bar{c}_{ij} \leq 0$ , para todo  $(i,j)$  no básico, el último árbol es óptimo. Puesto que el costo del flujo en  $R'$  es 0 entonces existe solución factible para el problema en la red original  $R$ .

Segunda fase.

Iteración 1. Eliminense el vértice  $f$  y el arco  $(2, f)$  del árbol óptimo resultante en la primera fase. Entonces el árbol que se obtiene es una base factible para el problema original.

A continuación se muestran variables duales y los coeficientes de costo reducido de los arcos no básicos.



La solución actual es óptima ya que  $\bar{c}_{ij} \leq 0$ , para todo  $(i, j)$  no básico.

#### 5.4 METODO SIMPLEX ESPECIALIZADO EN REDES CON VARIABLES ACOTADAS.

En base a los conceptos expuestos en las secciones precedentes resulta sencillo extender el método simplex especializado en redes para el caso donde el flujo a través de cada arco es tá acotado superior e inferiormente. El problema de flujo a costo mínimo en la red  $R = [X, A, r, q, c, b]$ , donde  $r$ ,  $q$  y  $c$  son los vectores de capacidades inferior, superior y costo unitario del flujo a través de los arcos de  $R$  respectivamente y  $b$  es el vector de ofertas de los vértices de  $R$ , puede plantearse como el problema de programación lineal (P) siguiente:

$$\begin{aligned} \text{(P):} \quad & \text{Min } z = cx \\ & \text{s.a.} \\ & Mx = b \\ & r \leq x \leq q \end{aligned}$$

donde  $x$  es el vector de flujos a través de los arcos de  $R$ . Este problema puede resolverse utilizando el método simplex para variables acotadas. Procediendo del mismo modo que en el caso de flujos no acotados se tiene que si se agrega una variable artificial  $x_h$  o, equivalentemente, se enraiza la red en un vértice  $h$ , las bases para el problema transformado son  $n-1$  columnas linealmente independientes junto con el vector  $e^h$ ; es decir, de nuevo las bases están caracterizadas por árboles expandidos enraizados de la red  $R$ . A continuación se analiza la especialización en redes de cada uno de los pasos del método simplex para variables acotadas.

Solución inicial. Como en el caso de variables no acotadas, se empezará con variables básicas artificiales. Puesto que los arcos originales son todos no básicos, se define el flujo a través de éstos igual a su cota inferior o a su cota superior. Se calcula  $b'_i = b_i - \sum_j x_{ij} + \sum_k x_{ki}$ , para todo  $i \in X$ .

De esta manera se definen los arcos  $(i, f)$  con flujo  $b'_i$ , si  $b'_i \geq 0$ , y  $(f, i)$  con flujo  $-b'_i$ , si  $b'_i < 0$ , donde  $f$  es el vértice que se ha añadido a  $R$ . Defínanse costos iguales a 1 para los arcos artificiales y a 0 para los arcos originales.

#### Cálculo de las variables básicas.

Los valores del flujo a través de los arcos básicos se calculan de la misma manera que en el caso no acotado utilizando el vector  $b'$  para reflejar los valores de los flujos a través de los arcos no básicos. Se procede de esta manera ya sea en la primera fase o en la segunda.

#### Cálculo de los coeficientes de costo reducido.

Las variables duales y los coeficientes de costo reducido se calculan exactamente de la misma manera que en el caso no acotado. Sin embargo, el criterio de optimalidad en presencia de cotas es:

$$\text{Si } \bar{c}_{ij} = w_i - w_j - c_{ij} \leq 0, \text{ para toda } x_{ij} = r_{ij}$$

$$\text{y } \bar{c}_{ij} = w_i - w_j - c_{ij} \geq 0, \text{ para toda } x_{ij} = q_{ij},$$

entonces la solución actual es óptima; en otro caso se procede a determinar una solución básica factible de menor costo.

Determinación del arco que entra a la base.

Según el criterio de optimalidad, el arco que entra a la base es cualquiera con flujo igual a su cota inferior tal que  $\bar{c}_{ij} > 0$ , o con flujo igual a su cota superior tal que  $\bar{c}_{ij} < 0$ .

Determinación del arco que sale de la base.

Sea  $w = (r, s)$  el arco que entra a la base. Sea  $C$  la única cadena que une  $r$  con  $s$  en el árbol  $T$  correspondiente a la base actual. Supóngase que  $x_w = r_w$  y supóngase que se define un nuevo flujo:

$$x_j + D, \text{ si } O_j(C) = -1$$

$$x_j - D, \text{ si } O_j(C) = 1$$

$$x_w + D, \text{ para el arco } w.$$

Este flujo resulta ser factible si  $D \leq q_j - x_j$ , para todo  $j \in A$  tal que  $O_j(C) = -1$ ,  $D \leq x_j - r_j$ , para todo  $j \in A$  tal que  $O_j(C) = 1$  y  $D \leq q_w - x_w$ . Esto se debe a que el flujo a través de cada arco debe tener un valor entre sus cotas inferior y superior. El valor máximo de  $D$  es por tanto  $\text{Min}\{D_1, D_2, q_w - x_w\}$ , donde  $D_1 = \text{Min}\{q_j - x_j \mid j \in A, O_j(C) = -1\}$  y  $D_2 = \text{Min}\{x_j - r_j \mid j \in A, O_j(C) = 1\}$ .

El arco que deja la base es entonces aquél para el cual el flujo alcance su cota inferior o su cota superior. Nótese que en este caso el flujo a través del arco que entra a la base puede cambiar de su cota inferior a la superior o viceversa manteniéndose, en este caso, la misma base. Para actualizar los valores de las variables se procede igual que en el caso de variables no acotadas.

Supóngase ahora que  $x_w = q_w$ . Entonces, al entrar  $w$  a la base, su valor disminuye. Para construir un flujo factible debe considerarse:

$$x_j - D, \quad \text{si } O_j(C) = -1$$

$$x_j + D, \quad \text{si } O_j(C) = 1$$

$$x_w - D, \quad \text{para el arco } w,$$

donde el máximo valor que puede tomar  $D$  es  $\text{Min}\{D_1, D_2, x_w - r_w\}$ , si  $D_1 = \text{Min}\{x_j - r_j \mid j \in A, O_j(C) = -1\}$  y  $D_2 = \text{Min}\{q_j - x_j \mid j \in A, O_j(C) = 1\}$ . Entonces el arco que deja la base es aquél para el cual el nuevo flujo sea igual a su cota inferior o a su cota superior. Para actualizar los valores de las variables básicas se procede de manera análoga que en el caso anterior.

Propósito. Determinar el flujo a costo mínimo en la red  $R = [X, A, r, q, c, b]$  con  $n$  vértices y cotas inferiores  $r$  y superiores  $q$  para el flujo a través de sus arcos.

Descripción.

- 1.- Considérese una solución básica factible y el árbol  $T = [X, A']$  enraizado en  $h$  correspondiente. Sea  $B$  la base triangularizada asociada a  $T$ . Calcúlese los valores de los flujos a través de los arcos de  $T$  de la siguiente manera. Si  $p$  es un vértice pendiente del árbol resultante de eliminar de  $T$  los vértices correspondientes a renglones de  $B$  anteriores al de  $p$  junto con sus arcos adyacentes y  $w$  es el único arco adyacente a  $p$ :

$$x_w = b_p + \sum_j x_{j p} - \sum_{i \neq k} x_{p i}, \quad \text{si } w = (p, k),$$

o  $x_w$  igual al recíproco de esta cantidad si  $w = (k, p)$ .

- 2.- Calcúlese los valores de las variables duales resolviendo el sistema  $w_h = 0$  y  $w_i - w_j = c_{ij}$ , para todo arco  $(i, j) \in A'$ .

- 3.- Calcúlese los coeficientes de costo reducido  $\bar{c}_{ij} = w_i - w_j - c_{ij}$  para todo arco  $(i, j) \notin A'$ .

(i) Si  $\bar{c}_{ij} \leq 0$ , para todo  $(i, j) \notin A$  tal que  $x_{ij} = r_{ij}$ ,  
 y  $\bar{c}_{ij} \geq 0$ , para todo  $(i, j) \notin A$  tal que  $x_{ij} = q_{ij}$ ,  
 entonces la solución actual es óptima.

(ii) En otro caso, ir a 4.

4.- Sean  $F_1 = \{(i,j) \in A \mid x_{ij} = r_{ij} \text{ y } \bar{c}_{ij} > 0\}$  y

$F_2 = \{(i,j) \in A \mid x_{ij} = q_{ij} \text{ y } \bar{c}_{ij} < 0\}$ .

Determinese como arco que entra a la base cualquier

$(r,s) \in F_1 \cup F_2$  y asígnese:

$$d = \begin{cases} 1 & \text{si } (r,s) \in F_1 \\ -1 & \text{si } (r,s) \in F_2 \end{cases}$$

5.- Calcúlese

$$D_1 = \text{Min } \{x_j - r_j \mid O_j(C) = d\}, \quad D_2 = \text{Min } \{q_j - x_j \mid O_j(C) = -d\}$$

$$D_3 = \begin{cases} q_{rs} - x_{rs}, & \text{si } d = 1 \\ x_{rs} - r_{rs}, & \text{si } d = -1 \end{cases}$$

y 
$$D = \text{Min } \{D_1, D_2, D_3\}$$

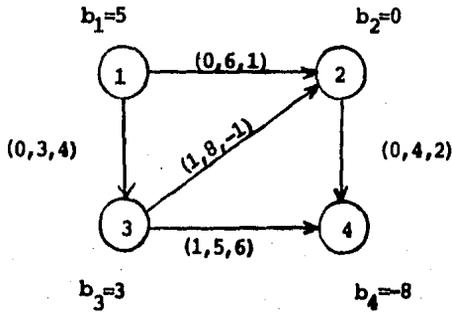
El arco que sale de la base es aquél para el cual  $x_u + O_u(C) \cdot D \cdot d$  es igual a  $r_u$  o a  $q_u$ . Si  $x_{rs} + d \cdot D$  es igual a  $r_{rs}$  o  $q_{rs}$  se mantiene la misma base.

6.- Actualícese los valores de las variables básicas como sigue:

$$x_k = \begin{cases} x_k + D \cdot d, & \text{si } k = (r,s) \\ x_k + O_k(C) \cdot D \cdot d, & \text{para todo } k \in C \\ x_k, & \text{si } k \notin C, k \neq (r,s) \end{cases}$$

y regrésese al paso 2.

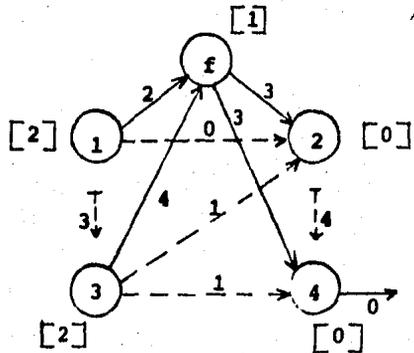
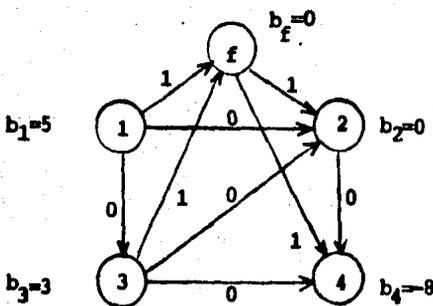
**Ejemplo 3.** Determinese, mediante el método simplex especializado en redes, el flujo a costo mínimo en la siguiente red.



Los números asociados a cada arco son la cota inferior, la cota superior y el costo unitario del flujo a través de él. Puesto que no se cuenta con una solución inicial se utiliza el método de las dos fases.

**Primera Fase.**

**Iteración 1.** A continuación se muestra la red  $R'$ , con los costos unitarios de flujo asociados a sus arcos, y el primer árbol factible (formado por las líneas continuas).



Para determinar el sentido de los arcos adyacentes a  $f$  se definieron los valores de los flujos a través de los arcos originales iguales a sus cotas inferiores o superiores; después se calculó:

$$\hat{b}_1 = b_1 - \sum_j x_{1j} + \sum_k x_{k1} = 5 - 3 = 2$$

$$\hat{b}_2 = b_2 - \sum_j x_{2j} + \sum_k x_{k2} = 0 - 4 + 1 = -3$$

$$\hat{b}_3 = b_3 - \sum_j x_{3j} + \sum_k x_{k3} = 3 - 1 - 1 + 3 = 4$$

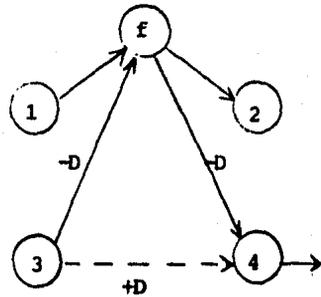
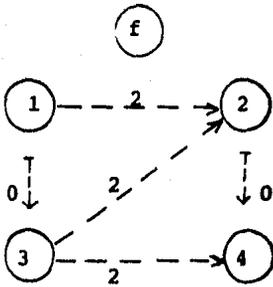
$$\hat{b}_4 = b_4 - \sum_j x_{4j} + \sum_k x_{k4} = -8 + 1 + 4 = -3$$

Puesto que  $\hat{b}_1$  y  $\hat{b}_3$  son mayores que 0, entonces se tienen los arcos (1,f) y (3,f) con flujo 2 y 4 respectivamente. Las cantidades  $\hat{b}_2$  y  $\hat{b}_4$  son menores que 0 por lo que se definen los arcos (f,2) y (f,4) con flujo igual a 3 en ambos. La base inicial está formada entonces por estos cuatro arcos junto con un arco raíz. En este primer árbol básico se ha asociado a cada vértice su potencial que fué calculado resolviendo el sistema  $w_4=0$  y  $w_i - w_j = c_{ij}$  para los arcos (i,j) básicos.

En la primera red de la siguiente figura se asocia a cada arco su coeficiente de costo reducido. En este caso se tiene:

$$F_1 = \{(1,2), (3,2), (3,4)\}; \quad F_2 = \emptyset$$

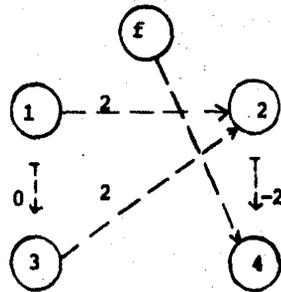
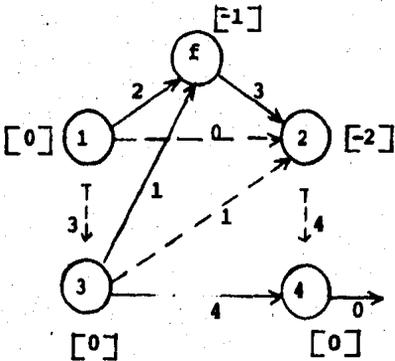
Se elige (3,4) para meter a la base por lo que  $d=1$ . Al agregar este arco al árbol básico se forma un ciclo con la cadena 3,f,4.



Se calcula:  $D_1 = \min \{x_{3f}, x_{f4}\} = 3$  ,  $D_2 = \infty$ ,

$D_3 = q_{34} - x_{34} = 5-1 = 4$  ,  $D = \min \{3, \infty, 4\} = 3$ .

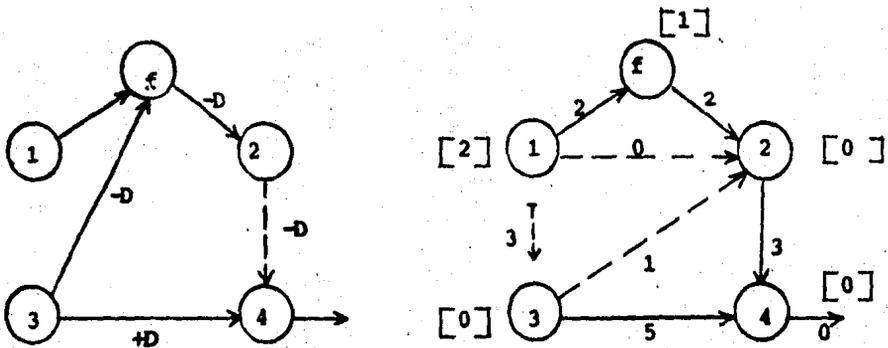
Entonces el arco que sale de la base es  $(f, 4)$ . Se actualizan los valores de los flujos a través de los arcos básicos, como se muestra en la figura anterior, obteniéndose el árbol de la siguiente figura (marcado de nuevo con líneas continuas).



Iteración 2. En el árbol básico de la figura anterior se ha asociado a cada vértice la correspondiente variable dual; en la segunda red de esta figura se ha asociado a cada arco su coeficiente de costo reducido. En este caso resulta:

$$F_1 = \{(1,2), (3,2)\} \quad ; \quad F_2 = \{(2,4)\}$$

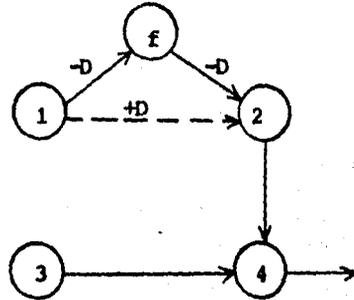
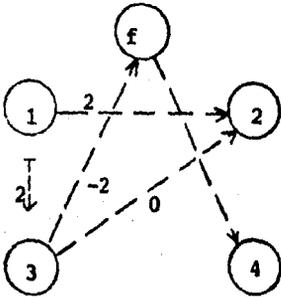
Se elige (2,4) para meter a la base;  $d=-1$ . Se forma ciclo con la cadena 2,f,3,4.



Se calcula  $D_1 = \min \{x_{f2}, x_{3f}\} = 1$  ;  $D_2 = \min \{5-4\} = 1$ ,  
 $D_3 = 4 - 0 = 4$  ;  $D = \min \{1,1,4\} = 1$ .

De donde el arco que sale de la base es (3,f). Se actualiza la base como se indica en la primera red de la figura anterior, obteniéndose el segundo árbol de dicha figura. (Obsérvese que esta solución es degenerada).

Iteración 3. Las variables duales se han asociado a los vértices del último árbol básico. A continuación se muestran los coeficientes de costo reducido de los arcos no-básicos.



De la primera red de la figura anterior se observa que:

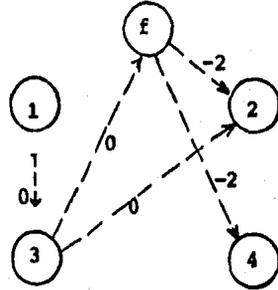
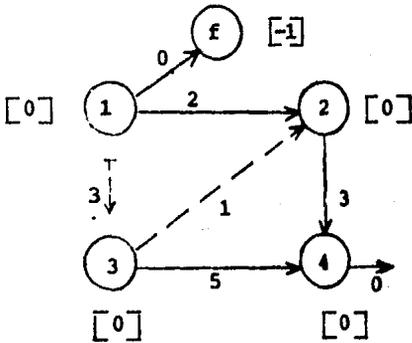
$$F_1 = \{(1,2)\} ; F_2 = \emptyset$$

Por lo que (1,2) entra a la base formándose un ciclo con la cadena 1,f,2. En este caso  $d=1$ , entonces:

$$D_1 = \text{Min} \{x_{1f}, x_{f2}\} = 2 ; D_2 = \infty$$

$$D_3 = 6 - 0 = 6 , \quad ; D = \text{Min} \{2, \infty, 6\} = 2$$

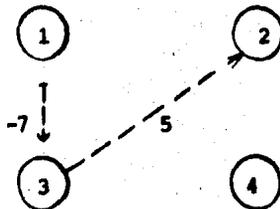
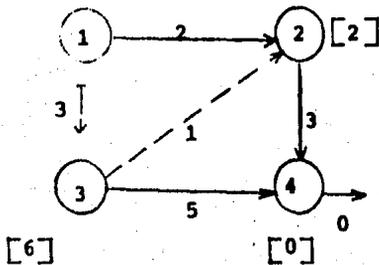
Por lo tanto pueden salir de la base (1,f) o (f,2). Elíjase (f,2). Se actualiza la base como se ilustra en la figura anterior resultando el siguiente árbol básico:



Los coeficientes de costo reducido, mostrados en la figura anterior, son menores o iguales que cero, para los arcos no básicos con flujo igual a su cota inferior, y mayores o iguales que cero, para los arcos no básicos con flujo igual a su cota superior. Luego, la solución actual es óptima. Por otro lado, como el flujo óptimo en  $R'$  es de costo cero, existe una solución factible para el problema en la red original.

Segunda fase.

Iteración 1. Eliminando el vértice  $f$  y el arco  $(1,f)$  se obtiene el siguiente árbol básico factible:



En base a los coeficientes de costo reducido de los arcos no básicos se tiene:

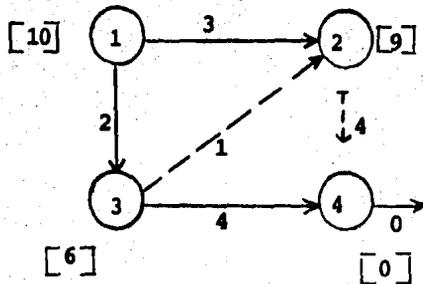
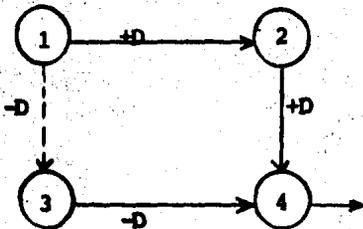
$$F_1 = \{(3,2)\} ; \quad F_2 = \{(1,3)\}.$$

Elijase (1,3) para meter a la base; entonces  $d = -1$ . Se forma ciclo con la cadena 1,2,4,3. Por tanto:

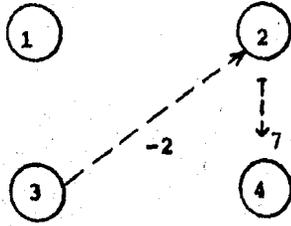
$$D_1 = \text{Min} \{x_{34} - r_{34}\} = 4 \quad , \quad D_2 = \text{Min} \{q_{12} - x_{12}, q_{24} - x_{24}\} = 1$$

$$D_3 = x_{13} - r_{13} = 3 \quad , \quad D = \text{Min} \{4, 1, 3\} = 1$$

El arco que sale de la base es entonces (2,4) ya que el flujo a través de él alcanza su cota superior. Actualizando el flujo como se ilustra en la primera red de la siguiente figura se obtiene el árbol que se muestra a continuación.



Iteración 2. Los coeficientes de costo reducido de los arcos no básicos, calculados en base a los potenciales calculados en la figura anterior, son:



Puesto que los coeficientes de costo reducido son menores o iguales que cero para los arcos no básicos con flujo igual a su cota inferior y mayores o iguales que cero para los arcos no básicos con flujo igual a su cota superior, se concluye que el árbol básico actual es óptimo.

**CONCLUSIONES**

## CONCLUSIONES

Como puede observarse, este trabajo ha sido enfocado a la presentación de resultados teóricos y algoritmos. Se hizo patente la importancia de los resultados establecidos en el desarrollo de los métodos de solución. También quedó demostrado, en algunos casos, que es posible utilizar estos métodos de solución en variantes de los problemas analizados, lo cual refuerza su importancia en la práctica. Además, se hizo notar que los cinco problemas están relacionados de alguna manera y, en el caso del problema del flujo a costo mínimo con ofertas en todos los vértices, se observó la importancia de la programación lineal en los modelos de redes. Sin embargo, sería interesante estudiar las relaciones de estos problemas básicos con otros como el de asignación, el de acoplamiento de cardinalidad máxima, el de transporte, etc. También podrían llevarse a cabo estudios comparativos de los métodos de solución o podría profundizarse más en las relaciones existentes con los modelos de programación lineal. De hecho, por ejemplo, el método general presentado para la determinación de la arborescencia de rutas más cortas es un método simplex especializado; las etiquetas que se asignan a los vértices, durante la aplicación del algoritmo, resultan ser los recíprocos de las variables duales del programa lineal correspondiente. También se establecen, implícitamente, relaciones de dualidad en los problemas de flujo máximo y del flujo a costo mínimo. Con esto queda demostrado que el campo de estudio de los modelos de redes es muy amplio y variado. Aún quedan problemas abiertos en esta interesante rama de las matemáticas.

## A. ELEMENTOS DE TEORIA DE GRAFICAS

El objetivo de este anexo es presentar los conceptos básicos de teoría de gráficas que son utilizados durante la exposición de los problemas de redes tratados en este trabajo.

Una gráfica es una pareja de conjuntos  $[X, A]$ , donde  $X$  es un conjunto de puntos llamados vértices o nodos y  $A$  es un conjunto de líneas que unen todos o algunos de los vértices. Se denota con  $G = [X, A]$ .

Si los elementos de  $A$  tienen una dirección, representada con una flecha, se llaman arcos y se dice que la gráfica  $G$  es dirigida u orientada. Si no tienen dirección se llaman aristas y  $G$  es no dirigida.

Un arco puede representarse como la pareja  $(i, j)$ , donde  $i, j \in X$  son los vértices que une dicho arco. Si  $a = (i, j) \in A$ ,  $i$  es el vértice o extremo inicial de  $a$  y  $j$  es el vértice o extremo final de  $a$ .

Sea  $G = [X, A]$  una gráfica dirigida y sea  $i \in X$ . Se llama sucesor de  $i$  a todo vértice  $j \in X$  tal que existe  $(i, j) \in A$ . Se llama predecesor de  $i$  a todo vértice  $j \in X$  tal que existe  $(j, i) \in A$ . También se definen las funciones  $\Gamma^+, \Gamma^- : X \rightarrow p(X)$  (conjunto potencia de  $X$ ), donde, para  $i \in X$  :

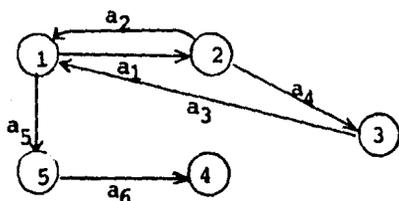
$$\Gamma^+(i) = \{\text{sucesores de } i\} = \{j \in X \mid (i, j) \in A\} \text{ y}$$

$$\Gamma^-(i) = \{\text{predecesores de } i\} = \{j \in X \mid (j, i) \in A\}.$$

# **ELEMENTOS DE TEORIA DE GRAFICAS**

**APENDICE**

Como ejemplo considérese la siguiente gráfica dirigida:



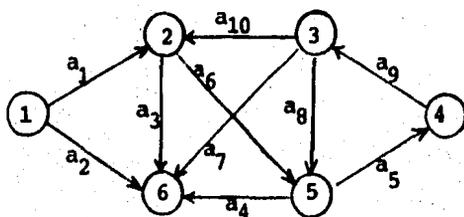
Puede verificarse fácilmente que:  $\Gamma^+(1) = \{2, 5\}$ ,  $\Gamma^+(2) = \{1, 3\}$ ,  $\Gamma^+(4) = \emptyset$ ,  $\Gamma^-(1) = \{2, 3\}$ ,  $\Gamma^-(5) = \{1\}$ , etc.

También puede definirse, tanto para gráficas dirigidas como no-dirigidas, el siguiente concepto. Un vértice  $j$  es vecino de un vértice  $i$  si existe la arista  $(i, j) \in A$ . Si la gráfica es dirigida  $j \in X$  es vecino de  $i \in X$  si  $j$  es predecesor o sucesor de  $i$ . Algunas veces se dice que  $i$  y  $j$  son adyacentes. Se define la función  $\Gamma: X \rightarrow \mathcal{P}(X)$  como  $\Gamma(i) = \{\text{vecinos de } i\}$  para  $i \in X$ . En la gráfica anterior se tiene  $\Gamma(1) = \{2, 3, 5\}$ ,  $\Gamma(4) = \{5\}$ , etc. Obsérvese que en gráficas dirigidas se cumple  $\Gamma(i) = \Gamma^+(i) \cup \Gamma^-(i)$ , para  $i \in X$ .

De nuevo considérese una gráfica  $G = [X, A]$  dirigida. Se definen los siguientes conceptos en ella.

Un camino es una secuencia de arcos en la cual el vértice final de uno es el vértice inicial del que le sigue en la secuencia. Un camino puede representarse con la secuencia de arcos que lo forman o por la secuencia de vértices extremos de estos arcos o por la secuencia alternada de vértices y arcos. Si  $a_1 = (i_1, i_2)$  es el primer arco del camino y  $a_q = (i_q, i_{q+1})$  es el último se dice que el camino es un camino de  $i_1$  a  $i_{q+1}$ . Un camino simple es un camino  $a_1, a_2, \dots, a_q$  tal que  $a_i \neq a_j$  si  $i \neq j$ .

Un camino elemental es un camino  $i_1, i_2, \dots, i_q$  (secuencia de vértices) tal que  $i_j \neq i_k$  si  $j \neq k$ . Dicho en otras palabras, un camino simple es uno en el cual no se repiten arcos y uno elemental es uno que no repite vértices. Obsérvese que todo camino elemental es también simple. Como ejemplo considérese la siguiente gráfica.



La secuencia  $a_5, a_9, a_8, a_5$  es un camino de 5 a 4. También puede representarse:  $5, a_5, 4, a_9, 3, a_8, 5, a_5, 4$  o  $5, 4, 3, 5, 4$ . Este camino no es ni simple ni elemental. El camino  $a_6, a_5, a_9, a_8, a_4$  es simple y no elemental. El camino  $a_6, a_5, a_9$  es camino simple y elemental.

Enseguida se define un concepto válido para gráficas dirigidas y no dirigidas. Una cadena es una secuencia de aristas (o arcos)  $a_1, a_2, \dots, a_q$  donde toda  $a_i$  está conectada a  $a_{i-1}$  por un extremo y a  $a_{i+1}$  por el otro. Si la cadena es de cardinalidad dos, estas dos aristas deben estar conectadas por uno de sus extremos. En la gráfica anterior  $a_1, a_{10}, a_9, a_5$  (o  $1, 2, 3, 4, 5$ ) es una cadena que une los vértices 1 y 5. Análogamente que como fueron definidos

camino simple y camino elemental se definen cadena simple y cadena elemental.

Hay algunos caminos y cadenas especiales por lo que reciben un nombre particular. Un circuito es un camino  $a_1, a_2, \dots, a_q$  donde el vértice final de  $a_q$  y el inicial de  $a_1$  coinciden; es decir, un circuito es un camino "cerrado". En la gráfica anterior  $a_5, a_9, a_8$  es un circuito. Un circuito simple es un circuito con todos sus arcos distintos. Un circuito elemental es un circuito con todos sus vértices, excepto el inicial, distintos. Un ciclo es una cadena  $i_1, i_2, \dots, i_q$  donde  $i_1 = i_q$ . En la gráfica anterior  $a_4, a_8, a_7$  es un ciclo.

Se conoce como grados de un vértice a lo siguiente. Sea  $G = [X, A]$  una gráfica dirigida y sea  $i \in X$ . El grado exterior de  $i$  es el número de arcos que tienen a  $i$  como vértice inicial. Se denota  $g^+(i)$ . El grado interior de  $i$  es el número de arcos que tienen a  $i$  como extremo final. Se denota  $g^-(i)$ . Obsérvese que  $g^+(i)$  es igual a la cardinalidad de  $\Gamma^+(i)$  y  $g^-(i)$  es igual a la cardinalidad de  $\Gamma^-(i)$ . Además, es fácil verificar que, si  $m$  es el número de arcos y  $n$  es el número de vértices de  $G$ , entonces:

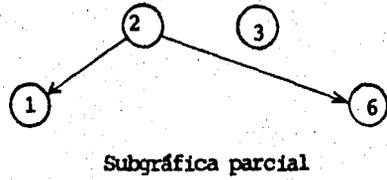
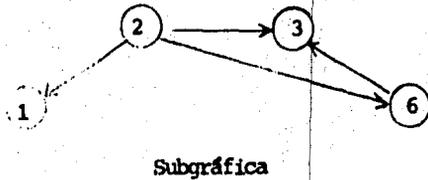
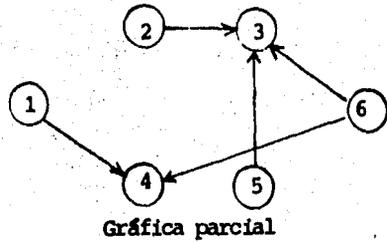
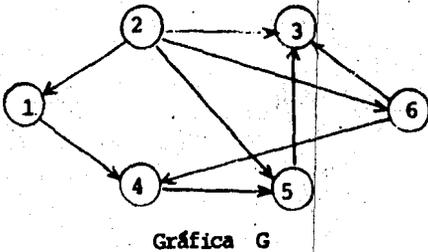
$$\sum_{i=1}^n g^+(i) = \sum_{i=1}^n g^-(i) = m$$

El grado de  $i \in X$ , es el número de arcos que tienen a  $i$  como uno de sus extremos. Se denota  $g(i)$ .

En gráficas no dirigidas se define el grado de  $i \in X$  como la cardinalidad de  $\Gamma(i)$ .

En la gráfica anterior se tiene que  $g^+(6)=0$ ,  $g^-(6)=4$ ,  $g(6)=4$ .

Existen ciertos subconjuntos de gráficas que son de utilidad y a continuación se definen. Sea  $G = [X, A]$  una gráfica. Una gráfica parcial de  $G$  es la gráfica  $G_p = [X, A_p]$ , donde  $A_p \subset A$ ; es decir, es una gráfica constituida por todos los vértices y algunas aristas o arcos de  $G$ . Una subgráfica de  $G$  es la gráfica  $G_s = [X_s, A_s]$ , donde  $X_s \subset X$  e  $(i, j) \in A_s$  si y sólo si  $i, j \in X_s$  e  $(i, j) \in A$ ; es decir, es una gráfica que consta de un subconjunto de los vértices de  $G$  y todas las aristas o arcos de  $G$  que unen los vértices de tal subconjunto. Como ejemplo considérese la siguiente gráfica, una gráfica parcial de ella, una subgráfica de  $G$  y una subgráfica parcial de  $G$  (subgráfica de la gráfica parcial).



Un concepto utilizado muy frecuentemente es el de conexidad.

Una gráfica  $G = [X, A]$  es conexa si para todo par de vértices  $i, j \in X$  existe una cadena que los une. Todas las gráficas anteriores, excepto la última, son conexas.

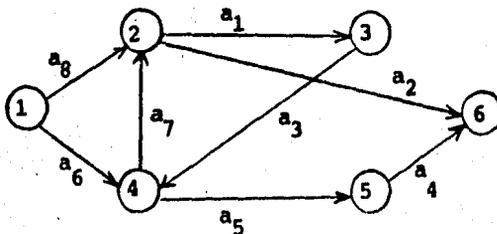
Las gráficas pueden ser representadas algebraicamente por medio de matrices. Sea  $G = [X, A]$  una gráfica dirigida. La matriz de adyacencia de  $G$  es una matriz  $B$  cuadrada  $n \times n$ , donde  $n$  es el número de vértices de  $G$ , de componentes.

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

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

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

Si se considera la gráfica siguiente, sus matrices de adyacencia e incidencia son  $B$  y  $M$ , respectivamente.





Por último se define el concepto de red. Una red  $R = [X, A, f]$  es una gráfica ponderada; es decir, es una gráfica  $[X, A]$  con una función real  $f$  definida sobre sus arcos o aristas o sobre sus vértices o sobre ambos.

## B I B L I O G R A F I A

1. Bazaraa, M.S. y Jarvis, J.J. "Programación Lineal y Flujo en Redes". Limusa. 1981.
2. Busaker, R.G. y Saaty, T.L. "Finite Graphs and Networks: An introduction with applications". Mc Graw-Hill. 1965.
3. Carré, B. "Graphs and Networks". Oxford University Press. 1979.
4. Christofides, N. "Graph Theory: An algorithmic Approach". Academic Press. 1975.
5. Hillier, F. S. y Lieberman, G.J. "Introduction to Operations Research". Holden-Day, Inc. 1980.
6. Hu, T.C. "Integer Programming and Network Flows". Addison-Wesley. 1969.
7. Jensen, P.A. y Barnes, J.W. "Network Flow Programming". John Wiley & Sons. 1980.
8. Kennington, J.L. y Helgason, R.V. "Algorithms for Network Programming". John Wiley & Sons. 1980.
9. Minieka, E. "Optimization Algorithms for Networks and Graphs". Marcel Dekker. 1978.

10. Phillips, D.T.; Ravindran, A. y Solberg, J.J. "Operations Research: Principles and Practice". John Wiley & Sons. 1976.
11. Prawda, J. "Métodos y Modelos de Investigación de Operaciones". (Vol. 1). Limusa. 1979.
12. Sakarovitch, M. "Introduction a l'etude des Graphes". Université Scientifique et Médicale de Grenoble. 1975.
13. Sakarovitch, M. "Optimization dans les réseaux". Université Scientifique et Médicale de Grenoble. 1977.