

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
FACULTAD DE INGENIERÍA



DIVISIÓN DE INGENIERÍA MECÁNICA E INDUSTRIAL



# TEORÍA DE REDES

Idalia Flores De La Mota



# TEORÍA DE REDES

Idalia Flores De La Mota



Acrobat Reader  
Haz Click

FLORES De La Mota, Idalia

*Teoría de redes*

Universidad Nacional Autónoma de México,

Facultad de Ingeniería, 2023, 307 p.

ISBN: 978-607-30-8297-6

---

## Teoría de redes

Primera edición electrónica

de un ejemplar (24 MB) en formato PDF

Publicado en línea en noviembre de 2023

D.R. © 2023, Universidad Nacional Autónoma de México,  
Avenida Universidad 3000, Col. Universidad Nacional Autónoma de México,  
Ciudad Universitaria, Delegación Coyoacán, C.P. 04510, México, CDMX.

FACULTAD DE INGENIERÍA

<http://www.ingenieria.unam.mx/>

ISBN de la obra impresa: 968-36-9076-9

ISBN de la obra digital: 978-607-30-8297-6

Esta edición y sus características son propiedad de la Universidad  
Nacional Autónoma de México. Prohibida la reproducción  
o transmisión total o parcial por cualquier medio  
sin la autorización escrita del titular de los derechos patrimoniales.

Hecho en México.

---

UNIDAD DE APOYO EDITORIAL

Cuidado de la edición: María Cuairán Ruidíaz  
Diseño y formación editorial: Nismet Díaz Ferro

### Idalia Flores De La Mota

La Dra. Idalia Flores de la Mota es matemática por la Facultad de Ciencias de la UNAM. Estudió la Maestría y el Doctorado en Investigación de Operaciones en la Facultad de Ingeniería de la UNAM. Se graduó de la maestría con mención honorífica y obtuvo la medalla Gabino Barreda por el mejor promedio de su generación. Desde 1987 forma parte del personal académico de la Facultad de Ingeniería, y cuenta con una trayectoria académica de más de treinta años. Actualmente es Profesora Titular “B” Definitiva de tiempo completo en el Departamento de Sistemas. Desde su ingreso en el Posgrado en Ingeniería en Sistemas ha impartido varias materias, relacionadas con la investigación de operaciones, ingeniería industrial, matemáticas e investigación. Ha publicado apuntes para las materias que ha impartido, ha sido coordinadora y autora de Cuadernillos de Divulgación sobre Investigación de Operaciones, publicados por la Facultad de Ingeniería, con apoyo de proyectos PAPIIT y PAPIIME. Ha publicado artículos en revistas internacionales y libros sobre Simulación y Optimización. Ha sido árbitro de diferentes revistas internacionales.

Como parte de las actividades del Departamento de Investigación de Operaciones e Ingeniería Industrial de la División de Ingeniería Mecánica e Industrial de la Facultad de Ingeniería UNAM, nos hemos propuesto el desarrollo de una serie de apuntes que sirvan de apoyo a los diferentes cursos que se imparten y, desde luego, como material de referencia o de lectura para quienes así lo requieran.

En dichos apuntes se busca mantener un alto nivel, de manera que contribuyan a una sólida formación teórica del alumno, pero al mismo tiempo se intenta acercarlo a las aplicaciones de tales conocimientos, como es en síntesis el objetivo central del posgrado de ingeniería.

Estos apuntes constan de ocho capítulos, en los cuales se desarrolla cada uno de los temas con unas notas históricas al final de cada uno.

Como autora de estos apuntes quiero expresar mi agradecimiento a Alejandro Zárate y Mayela Luna por su colaboración en la elaboración de figuras y revisión de la redacción. A la Unidad de Apoyo Editorial, en particular a la Maestra María Cuairán Ruidíaz y a la Lic. Nismet Díaz Ferro por el trabajo de revisión y comentarios, así como por el diseño y formación que hicieron mejorar la calidad de este trabajo. Asimismo, a la Lic. Patricia García Naranjo titular de la UDAE por su apoyo en la publicación de este texto.

# PRÓLOGO

La Teoría de Redes es un área dentro de la Investigación de Operaciones cuyo origen está vinculado a la Programación Lineal. El desarrollo de la computación ha provocado una mayor variedad y velocidad de los algoritmos propios de esta teoría y en general en la teoría del diseño de algoritmos. Su versatilidad y la representación gráfica de los problemas hace de esta una de las áreas más usadas en el campo de las aplicaciones, cuyo rango es tan amplio que abarca la química, la física, redes de computadoras, muchas ramas de la ingeniería, manufactura, política pública y sistemas sociales, tráfico urbano, telecomunicaciones y transporte, por citar solo algunos. El problema central de resolver un problema de flujo a costo mínimo conlleva a que los problemas de flujo máximo, ruta más corta, asignación, transporte y circulación se puedan ver ligados y formando parte de este problema central.

Estos apuntes tienen como objetivo ser un apoyo didáctico para la clase, así como un complemento de la bibliografía sugerida para el curso. Ya que por su amplitud no es posible cubrir todos los tópicos de redes que están contenidos en un libro de texto, se da una visión sencilla y general del tema. Los grandes temas que se abordan son flujo máximo y corte mínimo, ruta más corta, árbol de expansión mínima, ruta crítica y por último el problema de flujo a costo mínimo. Se exponen diferentes algoritmos de un mismo problema para compararlos y ver su complejidad computacional, así como una variedad de ejemplos desarrollados paso a paso para facilitar su comprensión. Finalmente se expone una serie de notas históricas al final de cada capítulo, con el objetivo de que el lector tenga una visión más amplia sobre cada tema.



# ÍNDICE

Sobre la autora .....	II
Presentación .....	III
Prólogo .....	IV

## Formulación de problemas de redes

1.1 Introducción .....	1
1.2 Tipos básicos de problemas de redes .....	8
1.3 Redes de tránsito urbano .....	9
1.4 Planteamiento de modelos de redes .....	10
1.5 El problema de la ruta más corta .....	11
1.6 El problema de flujo máximo .....	14
1.7 El árbol de expansión mínima .....	16
1.8 Planteamiento de modelos de redes como problemas de programación lineal .....	23
1.9 Notas históricas .....	40
1.10 Ejercicios propuestos .....	41

## Conceptos básicos de redes

2.1 Introducción .....	44
2.2 Representación matricial de una red .....	44
2.3 Redes especiales .....	47
2.4 Flujo y divergencia en una red .....	51
2.5 Circulaciones .....	58
2.6 Potenciales y tensiones .....	59
2.7 Flujos óptimos y potenciales .....	64

2.8	Notas históricas . . . . .	65
2.9	Ejercicios propuestos . . . . .	68

## CAP. 3

**Trayectorias y cortes**

3.1	Introducción . . . . .	70
3.2	El problema de determinar una trayectoria . . . . .	74
3.3	Algoritmo de enrutamiento . . . . .	79
3.4	Aplicaciones a conexidad. . . . .	89
3.5	Notas históricas . . . . .	91
3.6	Ejercicios propuestos . . . . .	92

## CAP. 4

**El problema de flujo máximo-corte mínimo**

4.1	Introducción . . . . .	94
4.2	Principio fundamental de divergencia . . . . .	94
4.3	Problema de flujo máximo en una red . . . . .	100
4.4	El problema de factibilidad de flujo . . . . .	112
4.5	Descripción del algoritmo de etiquetado de Ford Fulkerson. . . . .	123
4.6	Complejidad computacional . . . . .	128
4.7	Notas históricas . . . . .	129
4.8	Ejercicios propuestos . . . . .	132

## CAP. 5

**Árbol de expansión mínima**

5.1	Introducción . . . . .	134
5.2	Árbol de expansión mínima. . . . .	143
5.3	Algoritmo Kruskal . . . . .	145
5.4	Algoritmo de Prim . . . . .	150
5.5	Algoritmo de Sollin. . . . .	155
5.6	Análisis de sensibilidad . . . . .	161
5.7	Notas históricas . . . . .	165
5.8	Ejercicios propuestos . . . . .	167

**Problemas de ruta más corta**

6.1	Introducción . . . . .	169
6.2	Caracterización de una arborescencia . . . . .	173
6.3	Métodos de solución y justificación. . . . .	177
6.4	Problema de ruta más corta con programación dinámica. . . . .	203
6.5	Complejidad computacional de los algoritmos. . . . .	211
6.6	Notas históricas . . . . .	213
6.7	Ejercicios propuestos . . . . .	215

**Ruta crítica (PERT-CPM)**

7.1	Introducción . . . . .	217
7.2	Diferencias entre PERT y CPM. . . . .	219
7.3	Terminología de PERT/CPM. . . . .	221
7.4	Estructura de red . . . . .	223
7.5	Elaboración de la red . . . . .	225
7.6	Más sobre las actividades ficticias . . . . .	229
7.7	Análisis de una red PERT/CPM . . . . .	231
7.8	Problemas con incertidumbre o PERT . . . . .	247
7.9	Notas históricas . . . . .	254
7.10	Ejercicios propuestos . . . . .	257

**Problema de flujo a costo mínimo**

8.1	Introducción . . . . .	261
8.2	Formulación de programación lineal . . . . .	267
8.3	Algoritmo Simplex para redes con capacidades en los arcos. . . . .	272
8.4	Algoritmo Simplex para redes . . . . .	290
8.5	Notas históricas . . . . .	301
8.6	Ejercicios propuestos . . . . .	302

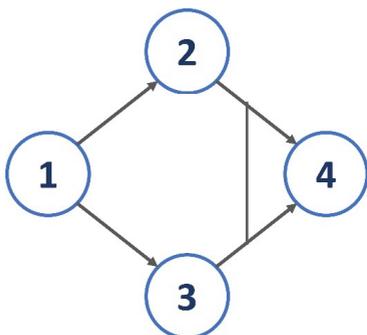
	Bibliografía. . . . .	305
--	-----------------------	-----

# Formulación de problemas de redes

## 1.1 INTRODUCCIÓN

En problemas de optimización donde se tiene una función objetivo y restricciones sean lineales o no, algunas veces es posible tener una representación visual del problema porque involucra el uso de una red. Por ejemplo, si se quiere localizar un servicio para darlo de mejor manera a los clientes, o se están buscando rutas para entrega de materiales. De esto se desprenden problemas que se consideran clásicos en lo que se conoce como teoría de redes. Es importante entonces hacer notar que se cuenta no solo con la representación en redes sino también con algoritmos propios que explotan las características de la red, así como la geometría de esta. El hecho de que se pueda contar con la representación hace que el problema sea más fácil de entender y en muchos casos de encontrar soluciones más rápidamente. En este capítulo se verán ejemplos de formulación de problemas en redes y cómo se pueden también formular como problemas de programación lineal para considerar esta herramienta como otra manera de modelarlos y resolverlos.

Cualquier red está formada por tres componentes (1) nodos, (2) arcos y (3) flujo en los arcos como se ilustra en la figura 1.1.



**Figura 1.1**  
Representación  
de una red

Los círculos son los nodos y están unidos por arcos, observe que en la figura hay dos tipos de arcos, dirigidos y no dirigidos. Un arco dirigido es aquel sobre el cual puede moverse el flujo en una sola dirección, y uno no dirigido es aquel sobre el cual puede moverse el flujo en cualquier sentido. En la figura 1.1, el arco que une los nodos 1 y 2 es un arco dirigido y el que une los nodos 2 y 3 es no dirigido.

Por lo general los nodos se numeran, tal como se ha mostrado en la figura 1.1 y los arcos se denotan por los nodos que unen. Por ejemplo, el arco que une los nodos 1 y 2 se identifica como el arco  $j_1 \sim (1,2)$ .

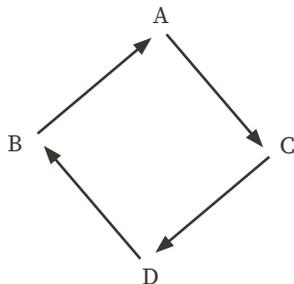
El flujo que pasa de un nodo a otro a través de un arco es un factor desconocido en la red y se le denota como  $X_{ij}$  para el flujo entre los nodos  $i$  y  $j$ .

El flujo de una red puede constar de muchos bienes o productos distintos. Unos cuantos ejemplos serían: gas natural en un gasoducto, distribución de artículos de mayoristas a detallistas o entre fábricas y almacenes. El costo unitario del flujo para cada arco se denota como

$C_{ij}$  para los nodos  $i$  y  $j$ . En el caso de una red carretera, las ciudades son los nodos y las rutas de transporte entre las ciudades son los arcos. El costo por automóvil para cada ruta es el costo del flujo. En algunos problemas pueden existir capacidades para cada arco que limiten la cantidad de flujo.

Puede entonces definirse una red como un conjunto de nodos, arcos y flujos que pasan de un nodo a otro a través de los arcos.

Un buen dibujo es mejor que una larga lectura (o una imagen dice más que mil palabras). El lenguaje de las gráficas trata de poner esta idea a tal efecto. De hecho, muy a menudo es un reflejo natural que nos hace abstraer una situación dada, dibujando en una hoja de papel puntos que pueden representar individuos.



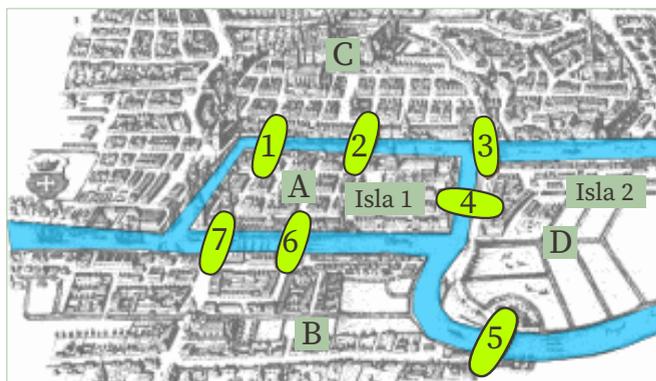
**Figura 1.2**  
Representación  
en imágenes

Lugares	
<input type="text"/>	A
<input type="text"/>	B
<input type="text"/>	C
<input type="text"/>	D
<input type="text"/>	E
<input type="text"/>	F
<input type="text"/>	G
<input type="text"/>	H

Y así conectando con líneas o flechas que simbolizan una cierta relación. Esta representación de la realidad por figuras tiene las siguientes ventajas:

1. De esta forma se expresa la estructura profunda de una situación dada.
2. Desde un punto de vista práctico, muestra una visión completa del problema, lo que representa una valiosa guía para la intuición y el razonamiento.

El primer documento en gráficas “*solutio problematis ad geometriam situs pertinentis*”, lo escribió Euler en 1736. Comenzó con una curiosidad matemática: cómo puede una persona caminar una y solo una vez a través de cada uno de los siete puentes de Königsberg. Dichos puentes se muestran en la figura siguiente:



**Figura 1.3**  
Los puentes  
de Königsberg<sup>1</sup>

Desde 1946 la teoría de las gráficas ha sido ampliamente desarrollada bajo la inspiración de muchos especialistas en investigación de operaciones, motivados por problemas actuales. En forma paralela, un esfuerzo importante de síntesis ha prevalecido.

Es a partir de 1960, que, ayudados por la aparición de las primeras computadoras electrónicas, se ha visto una explosión real de investigación y de aplicaciones.

---

<sup>1</sup> Tomado de <https://es.quizclub.com/trivia/quien-soluciono-el-problema-matematico-de-los-puentes-de-koenigsberg/>

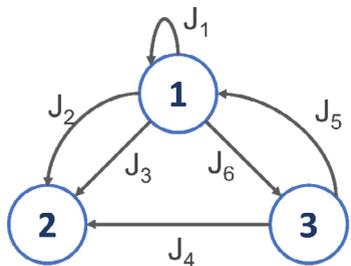
Una red  $G$  consiste en 3 componentes: Un conjunto  $N$  de nodos (vértices o puntos) que se representan con círculos pequeños, un conjunto  $A$  de arcos (líneas, flechas o cadenas) que se representan con flechas, la dirección de estas da una orientación al arco.

Y una función  $\rho : A \rightarrow N \times N$  que asigna a cada  $j \in A$  una pareja  $(i, i') \in N \times N$  tal que  $i \neq i'$  se supone  $N \neq \emptyset$ . Si  $i = i'$  se trata de un rizo.

## E

## Ejemplo 1.1

Representación en red donde se tienen tres nodos y seis arcos.



Los arcos  $j$  tal que  $\rho(j) = (i, i)$  como ya se ha dicho anteriormente se denominarán rizos.

Si no existen arcos paralelos es normal hacer

$$j \sim (i_1, i_2) \quad \text{o bien} \quad (i_1, i_2)$$

Y se dice que  $i_1$  es el nodo inicial de  $j$ , e  $i_2$  es el nodo final. También se dice que  $i_1, i_2$  son nodos adyacentes y que  $j$  incide en  $i_1, i_2$ .

$$N = \{1, 2, 3\}$$

$$A = \{J_1, J_2, J_3, J_4, J_5, J_6\}$$

$$\rho : A \rightarrow N \times N$$

O bien

$$\rho(J_1) = (1, 1)$$

$$\rho(J_2) = (1, 1)$$

$$\rho(J_3) = (1, 1)$$

$$\rho(J_4) = (1, 1)$$

$$\rho(J_5) = (1, 1)$$

$$\rho(J_6) = (1, 1)$$

Una red donde no existen arcos en paralelo se conoce como una gráfica dirigida o digráfica y el conjunto  $A$  se identifica como un subconjunto de  $N \times N$ . El término gráfico se usa para una estructura similar donde los arcos no están orientados.

## Ejemplo 1.2

El gerente de programación de la General Motors, de la región oriental de México, está interesado en elaborar un plan semanal para enviar automóviles de su puerto de entrada a diversas distribuidoras regionales. Para elaborar el plan, ha recopilado datos sobre costos de transporte por automóvil en todo el país, necesidades mensuales de automóviles de cada distribuidor y llegadas mensuales de automóviles a cada puerto de entrada. Los automóviles se pueden enviar en forma directa a cada distribuidor o puede enviarse un conjunto de ellos a un distribuidor, descargar algunos, y enviar el resto a algún otro distribuidor. En el mapa de la figura 1.4 se muestran la localización de los distribuidores y de los puertos de entrada.



**Figura 1.4**  
Los puentes  
de Könisberg<sup>2</sup>

<sup>2</sup> Tomado de <https://realestatemarket.com.mx/infraestructura-y-construccion/27568-impulsar-el-sur-sureste-de-mexico>

En la tabla 1.1 y en la figura 1.5 se muestran en pesos por automóvil, los costos de transporte entre los puertos de entrada, las ciudades de transbordo y las ciudades que son destinos finales. En los casos en los que no existe relación directa entre un par de ciudades no se muestran los costos.

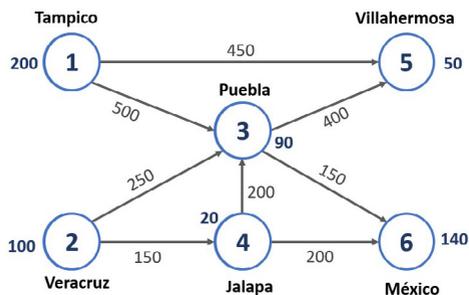
**TABLA 1.1** Costo entre ciudades

DE: CIUDAD	A: CIUDAD			
	Cd. Mx.	Villahermosa	Puebla	Jalapa
Tampico	-	\$450	\$500	-
Veracruz	-	-	\$250	\$150
Puebla	\$150	\$400	-	-
Jalapa	\$200	-	\$200	-

### Uso de la representación en redes

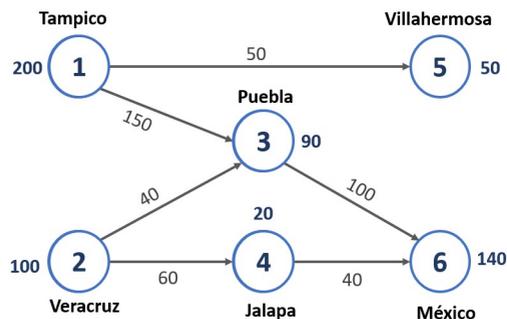
Este problema consiste en determinar la forma de enviar los automóviles de manera que se minimicen los costos, y al mismo tiempo, que se satisfagan las necesidades de todos los distribuidores.

En la red de la figura 1.5 los círculos representan a cada uno de los puertos de entrada y los distribuidores y las flechas, las relaciones entre ellos. Los números que están fuera de los círculos son las disponibilidades y las demandas según sea el caso de cada ciudad.



**Figura 1.5**  
Red de distribución

El gerente necesita determinar cuántos automóviles se deben enviar de Tampico y Veracruz a cada una de las otras ciudades, para que sean vendidos allí o transferidos a alguna otra ciudad. Se da cuenta de que no es difícil encontrar un plan de envíos que satisfaga las necesidades de todas las ciudades, como se muestra en la figura 1.6, pero ¿será este el plan más económico?



**Figura 1.6**  
Plan de envío

## 1.2 TIPOS BÁSICOS DE PROBLEMAS DE REDES

Se consideran cuatro tipos importantes de problemas de redes:

1. El de transbordo
2. El de la ruta más corta
3. El de flujo máximo
4. El de transporte

El problema de la General Motors es un ejemplo típico de problema de transbordo, puesto que se tienen disponibilidades y/o demandas y algunas ciudades que al mismo tiempo pueden recibir y enviar automóviles a otras. Los otros tipos de problemas se ilustran más adelante. Algo común a estos es que los cuatro se pueden plantear en forma similar a

los que se denominan problemas de flujo de costo mínimo; pero antes, es necesario analizar un poco la terminología de los problemas de redes.

### 1.3 REDES DE TRÁNSITO URBANO

El sistema de transporte consiste en redes que representan los modos disponibles (automóviles, autobuses, etc.) La descripción de la red es una abstracción de lo que actualmente existe en el campo, y como tal, no incluye todas las calles locales o las calles colectoras en el área. Se desarrolla una descripción de la red para representar los viajes en automóvil y camión, con una reseña separada para el transporte público, si se incluye este medio. Estas descripciones pueden abarcar la geometría del sistema de transporte.

La geometría de la red incluye la numeración de las intersecciones (llamadas nodos para propósitos de asignación). La numeración de los nodos permite identificar los segmentos entre ellos (llamados tramos). En las redes de transporte se pueden identificar grupos de tramos por los que pasan rutas específicas (llamadas líneas). Esta descripción geométrica de la red de transporte muestra todas las posibles vías por las que puedan realizarse los viajes entre puntos del área.

En la descripción de la red, se identifican los centroides de zona (centros de actividad); que son también conocidos como nodos de oferta. Estos se conectan a los nodos por medio de los tramos imaginarios denominados correctores de centroides. Los centroides se utilizan como los puntos en los cuales se efectúa la “carga” a la red.

Una vez que la red de transporte se ha descrito en términos de la manera como se pueden conectar los puntos, es necesario cuantificar la facilidad con la que se realizan estas conexiones.

Las velocidades de viaje y la capacidad de un tramo en una vía rápida probablemente serían mayores que en una calle arterial; esta diferencia de nivel de servicio debe cuantificarse e incluirse como parte de la descripción del sistema de transporte.

Para la descripción de la red vehicular, se deben reunir puntos específicos para determinar el nivel de servicio en cada tramo. Estos requerimientos incluyen aspectos físicos tales como la longitud del tramo y el número de carriles, ya sea que el tipo de instalación bajo consideración sea vía rápida, arterial, etc., así como la localización del tramo en el área urbana.

El costo de recorrido en un tramo específico se conoce como impedancia y tiene implícitas las mediciones de tiempo y costo.

## 1.4 PLANTEAMIENTO DE MODELOS DE REDES

### Problemas de transbordo

Si un problema de redes se refiere a la minimización de los costos del flujo de algún producto entre nodos, en donde cada nodo puede ser un punto de abastecimiento, un punto de demanda, o ambos, entonces se considera que el problema de redes es un problema de transbordo.

El problema de la General Motors es un excelente ejemplo de un problema de transbordo. Existen tres tipos de nodos en un problema de transbordo. Si un nodo actúa al mismo tiempo como receptor y emisor de flujo se le denomina nodo de transbordo. En la figura 1.5 los nodos 3 y 4 son nodos de transbordo. El problema de transbordo es el más general de los problemas de redes, dado que cada nodo puede

tener al mismo tiempo oferta y demanda y no existen restricciones sobre los flujos o sobre los tipos de nodos.

## 1.5 EL PROBLEMA DE LA RUTA MÁS CORTA

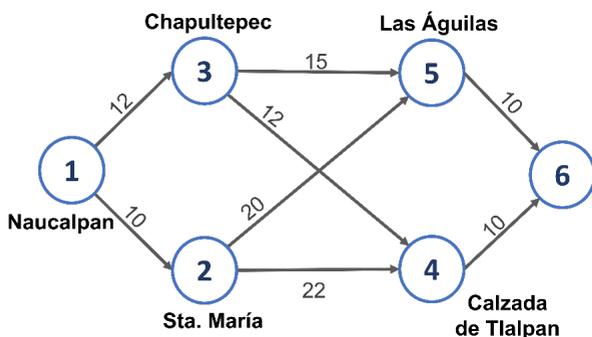
Si se define una red de manera que los coeficientes de cada arco sean no negativos (tal como medidas de distancia), entonces podríamos estar interesados en encontrar la ruta más corta entre dos nodos de la red. A este problema se le denomina problema de la ruta más corta.

Como ejemplo del problema de ruta más corta considere el siguiente:

### Ejemplo 1.3

La compañía de muebles “El mueble moderno” quiere transportar unos comedores de su planta en Naucalpan, Estado de México, a una distribuidora que se encuentra en Tlalpan, en el sur de la Ciudad de México, en el menor tiempo posible.

Las rutas que enlazan estas dos instalaciones forman la red que se muestra en la figura 1.7



**Figura 1.7**  
Rutas del mueble moderno

En donde las “distancias” son los tiempos de viaje del camión dados en minutos y considerando que se traslada a las 6 de la mañana y por lo tanto el tránsito es rápido, a una velocidad media de 50 kph.

Observe que, con el objeto de plantear este problema en forma de problema de costo mínimo, puede elegirse en cualquier nodo solo el camino a través de un arco.

Lo anterior implica que es necesario tener los siguientes flujos en los arcos:

$$X_{ij} = \begin{cases} 1 & \text{si se viaja a través de la ruta entre la instalación } i \\ & \text{y la instalación } j \\ 0 & \text{de otra manera} \end{cases}$$

Si se recorre la ruta  $(i, j)$ , esto significa que no puede utilizarse ninguna otra ruta que parta de la instalación  $i$ .

Por ejemplo, si se viaja por la ruta de Naucalpan a Chapultepec, entonces el flujo entre esos puntos será igual a 1 y  $X_{13}$  será cero.

Es posible satisfacer estas condiciones utilizando en la red un flujo imaginario de una unidad. Este flujo parte del nodo de salida u **origen** y llega a un nodo final o **terminal**. En otras palabras, existe un suministro de una unidad en el origen y una **demanda** de una unidad en el nodo terminal. En nuestro ejemplo, habría un suministro (u oferta) de una unidad en Naucalpan y una demanda de esa misma unidad en Tlalpan.

La última pregunta que permanece pendiente es: ¿Qué costos deben utilizarse en este problema de costo mínimo? Para responder a esta pregunta observe que si  $X_{ij}=1$ , entonces será necesario viajar de los

nodos  $i$  al  $j$ . Si se denotan estas distancias mediante  $d_{ij}$  y se utiliza la ruta entre  $i$  y  $j$ , entonces el costo para esa ruta se convierte en  $d_{ij} X_{ij}$ . Dado que  $X_{ij}$  es cero o uno, el costo para cualquier ruta será  $d_{ij}$  o cero. Por esto, podemos utilizar las distancias,  $d_{ij}$  como los costos para el problema de flujo a costo mínimo.

De acuerdo con la figura 1.7, para salir de Naucalpan se puede ir por dos caminos, por Chapultepec o por Santa María, pero no por ambas. Para los puntos Chapultepec, Santa María, Las Águilas y Calzada de Tlalpan se requiere que el flujo que llega a esos nodos sea igual al flujo que sale, puesto que no existe demanda en ninguno de ellos. La restricción para Tlalpan exige llegar a ese punto ya sea por Las Águilas o por Calzada de Tlalpan, obligando a que la suma de los flujos sea igual a 1.

En un problema de ruta más corta puede haber arcos dirigidos y arcos no dirigidos. En nuestro ejemplo, solo se tienen arcos dirigidos. Para el caso de arcos no dirigidos sería necesario tener una variable de  $i$  a  $j$  y otra de  $j$  a  $i$ . No sería difícil modificar el planteamiento para manejar las variables en ambas direcciones.

Finalmente, cabe mencionar que existen diversas variaciones del problema de la ruta más corta, y que la que se ha presentado aquí es solo una de ellas. Las dos variaciones que existen son:

1. Encontrar la ruta más corta entre algún nodo y cada uno de los otros nodos de la red.
2. Encontrar la ruta más corta entre cualquier par de nodos de la red.

## 1.6 EL PROBLEMA DE FLUJO MÁXIMO

En el problema anterior estábamos interesados en los valores que se generan a través de cierto flujo que pasa por una red. Este valor puede estar dado en términos de dinero, distancia, tiempo o alguna otra medida. Existen problemas en los que el valor del flujo no es tan importante como la **cantidad de flujo** que pasa a través de la red. Los gasoductos, tráfico de automóviles y las líneas de transmisión de electricidad son ejemplos de esta situación. Los problemas en los que interesa determinar el flujo máximo que pasa a través de una red se denominan **problemas de flujo máximo**.

Para estudiar este tipo de problemas, es necesario suponer que existen **restricciones de capacidad** en los arcos. Si no fuera así, el flujo máximo que pasaría a través de la red sería infinito. Considere el siguiente ejemplo:

E

### Ejemplo 1.4

PROCAMPO es un programa de la Secretaría de Agricultura que tiene entre otros un programa anual costalero. Esta se compra de dos fábricas, una en Mérida con capacidad de producción máxima de 10 millones de costales al año, y otra en Saltillo con capacidad de producción máxima de 7 millones de costales al año. Los excedentes en la fábrica de Mérida pueden transferirse a la planta de Saltillo.

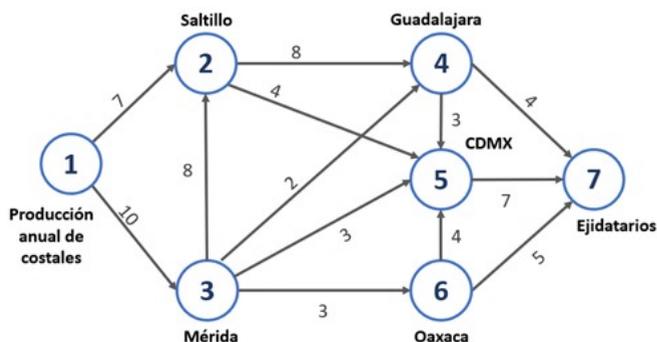
La disponibilidad de transporte entre las dos fábricas permite un máximo de 8 millones de costales por año. Hay tres almacenes: en la Ciudad de México, Guadalajara y Oaxaca. La tabla 1.2 proporciona la capacidad máxima anual de transporte de las fábricas a los almacenes.

**TABLA 1.2** Capacidad máxima anual de transporte

De: Ciudad	A: CIUDAD		
	Cd. Mx.	Guadalajara	Oaxaca
Saltillo	4	8	-
Mérida	3	2	3

Los excedentes de Guadalajara y Oaxaca pueden transferirse a la Ciudad de México. La capacidad máxima anual es de 3 y 4 millones de costales, respectivamente. Una vez en los almacenes, los costales se entregan a los ejidatarios de la región. La capacidad máxima anual de entrega es de 4 millones en la región del almacén de Guadalajara, 7 millones en la región de la Ciudad de México y 5 millones en la región de Oaxaca.

La pregunta es ¿cuál es el flujo máximo anual de costales nuevos que pueden circular en este sistema? El problema se puede representar gráficamente en la red de la figura 1.8



**Figura 1.8**  
Flujo de costales nuevos

Se puede observar en la red que los números en los arcos se refieren a las capacidades máximas de transporte, no se pusieron las capacidades mínimas por ser éstas todas iguales a cero. Así también los nodos 1 y 7 se pueden considerar nodos artificiales que permiten visualizar con mayor claridad la red.

## 1.7 EL ÁRBOL DE EXPANSIÓN MÍNIMA

Antes de especificar en que consiste este problema cabe mencionar lo que en redes se entiende por un árbol. Un **árbol** es un subconjunto de los arcos de la red original que conecta a todos los nodos sin formar ningún circuito.

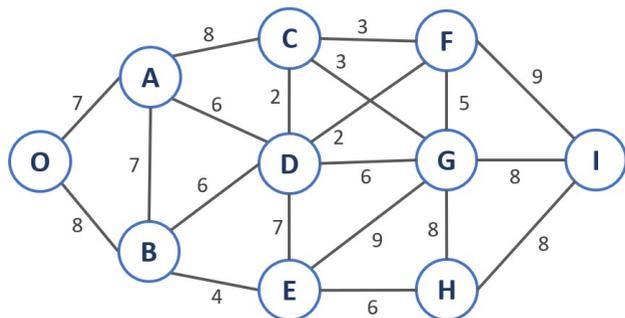
En este problema se conocen los costos o distancias entre diferentes nodos en una red. Sin embargo, los arcos **no** se especifican, y lo que se trata de encontrar es un árbol que comunique a todos los nodos de la red, pero cuyo costo o distancia total sea mínima.

Este tipo de problemas se ubica en las redes de comunicación eléctrica, telefónica, telegráfica, carretera, ferrocarrilera, aérea, marítima, etc., donde los nodos representan, por ejemplo, puntos de consumo eléctrico, teléfonos, telégrafos, terminales de autobuses, trenes, aeropuertos, puertos marítimos, etc., y los arcos podrían ser las líneas de alta tensión eléctrica, líneas telefónicas y telegráficas, carreteras y vías de ferrocarril, rutas aéreas y marítimas, entre otras. A continuación, se presenta un ejemplo de este caso.

E

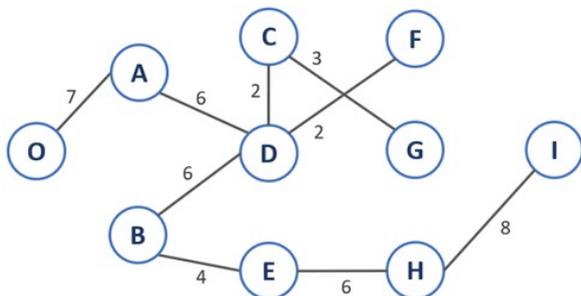
### Ejemplo 1.5

Suponga que en la red que se muestra en la figura 1.9, los nodos son centros de consumo eléctrico, y los números en los arcos son distancias en kilómetros. Se trata de encontrar el árbol que, con una longitud total mínima, comunica a todos los nodos. Como el costo de tendido de cable eléctrico es proporcional a la distancia, se habrá encontrado, con la distancia mínima, también el costo mínimo.



**Figura 1.9**  
Centros de consumo eléctrico

Después de usar algún paquete de cómputo el árbol de expansión mínima que conecta los centros de consumo y que no necesariamente es el único, se muestra en la figura 1.10

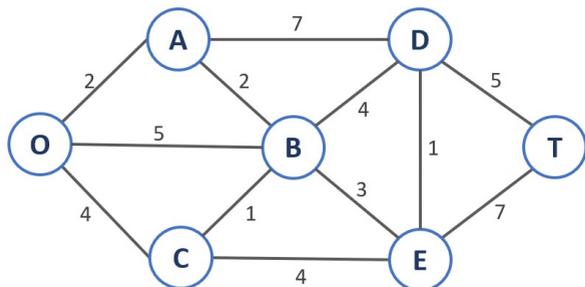


**Figura 1.10**  
Árbol de comunicación eléctrica

El siguiente ejemplo muestra los tres casos de problemas de redes que hemos visto hasta aquí.

## Ejemplo 1.6

En fecha reciente se ha reservado el parque La Marquesa para pasear y acampar. No se permite la entrada de automóviles al parque, pero existe un sistema de caminos angostos para tranvías y jeeps conducidos por los guardabosques.



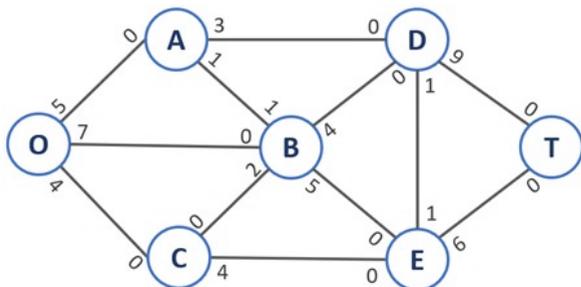
**Figura 1.11**  
Red de caminos  
de la Marquesa

En la figura 1.11 se muestra este sistema de caminos (sin las curvas), en donde O es la localización de la entrada al parque; las otras letras designan la localización de estaciones de guardabosques (y otras instalaciones). Los números dan las distancias de estos caminos sinuosos en kilómetros, los arcos no muestran flechas porque los caminos se pueden recorrer en ambas direcciones.

El parque contiene un paisaje maravilloso en la estación T. Se usa un número pequeño de tranvías para transportar visitantes de la entrada del parque a la estación T, y de regreso, para quienes desean contemplar este paisaje sin tener que caminar.

La dirección del parque está encarando en este momento tres problemas. Uno es determinar cuál ruta de la entrada del parque a la estación T tiene la menor distancia total, para la operación de los tranvías.

Un segundo problema se refiere a la instalación de líneas telefónicas debajo de los caminos para establecer comunicación de este tipo entre todas las estaciones, incluyendo la entrada al parque. Puesto que la instalación es cara y, a la vez, perturba el medio ambiente natural, se instalarán las líneas solo debajo del número suficiente de caminos para proveer cierta conexión entre todo par de estaciones. La cuestión es dónde deben colocarse las líneas para realizar esto con un número total mínimo de kilómetros de línea instalada.



**Figura 1.12**  
Número máximo de viajes en los caminos

El tercer problema es que, durante la temporada pico, más personas desean el viaje en tranvía desde la entrada del parque a la estación T que las que pueden ser acomodadas. Con el fin de evitar la perturbación indebida de la ecología y la vida salvaje de la región, se han racionado estrictamente el número de viajes al día del tranvía que pueden llevarse a cabo sobre cada uno de los caminos, estos límites son diferentes para los diferentes caminos como se muestra en la figura 1.12.

Por lo tanto, durante la temporada pico, podrían seguirse diversas rutas, sin importar la distancia, para incrementar el número de viajes por tranvía que pueden efectuarse diariamente. La cuestión consiste en trazar las rutas de los diversos viajes para maximizar el número de viajes que pueden efectuarse diariamente, sin violar los límites de cada camino.

Para resolver el problema de ruta más corta considere la siguiente tabla 1.3:

**TABLA 1.3** Distancias en la red y conexiones de los nodos (primera parte)

N	Nodos resueltos conectados a nodos no resueltos	Su nodo no resuelto conectado más próximo	Distancia total involucrada	N_ésimo nodo más cercano	Su distancia mínima	Su última conexión
1	O	A	2	A	2	OA
2	O A	C B	4 2+2=4	C B	4 4	OC AB

**TABLA 1.3** Distancias en la red y conexiones de los nodos (segunda parte)

N	Nodos resueltos conectados a nodos no resueltos	Su nodo no resuelto conectado más próximo	Distancia total involucrada	N.º nodo más cercano	Su distancia mínima	Su última conexión
3	A	D	$2+7=9$			
	B	E	$4+3=7$	E	7	BE
	C	E	$4+4=8$			
4	A	D	$2+7=9$			
	B	D	$4+4=8$	D	8	BE
	E	D	$7+1=8$	D	8	ED
5	D	T	$8+5=13$			
	E	T	$7+7=14$	T	13	DT

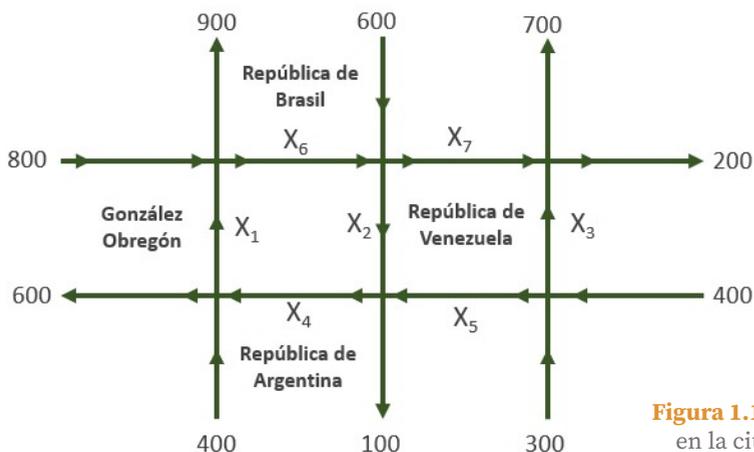
Ahora puede recorrerse hacia atrás la ruta más corta, del destino al origen, a través de la última columna de la tabla, como a  $T \rightarrow D \rightarrow E \rightarrow B \rightarrow A \rightarrow O$ , o bien,  $T \rightarrow D \rightarrow B \rightarrow A \rightarrow O$ . Por lo tanto, se han identificado las dos alternativas para la ruta más corta del origen al destino como  $O \rightarrow A \rightarrow B \rightarrow E \rightarrow D \rightarrow T$  y  $O \rightarrow A \rightarrow B \rightarrow D \rightarrow T$ , con una distancia total de 13 millas sobre cualquiera de las dos.

Como se podrá observar los problemas 1, 2 y 3 son respectivamente problema de ruta más corta, problema de árbol de expansión mínima y problema de flujo máximo, dicho problema se resolverá posteriormente.

## E

**Ejemplo 1.7**

En este ejemplo se muestra la forma en que se puede resolver un problema práctico de tránsito. Para concretar más, empezamos con una red que muestra una parte del centro de la ciudad de México.



**Figura 1.13** Red de calles en la ciudad de México

En la red se indicó el flujo de tránsito que entra o sale a cada calle, en unidades de vehículos por hora (VPH). Ya que el flujo de tránsito varía considerablemente durante el día, supondremos que los números mostrados representan el flujo de tránsito promedio a la hora de máximo flujo que se da aproximadamente entre las 3 y las 5 de la tarde.

Suponga ahora que un grupo político está planeando una manifestación en República de Argentina, entre República de Venezuela y González Obregón a las 4 de la tarde del miércoles. La policía puede, hasta cierto punto, controlar el flujo de tránsito reajustando los semáforos, colocando policías en los cruces clave, o cerrando la calle crítica al tránsito de vehículos. Si se disminuye el tránsito por República de Argentina, aumentará el de las calles adyacentes. La cuestión es minimizar el tránsito por República de Argentina (entre República de Venezuela y González Obregón) sin ocasionar congestionamientos en las otras calles. Para resolver este problema de minimización le agregamos marcas a nuestro mapa, ver figura 1.13.

Aquí se han marcado las seis intersecciones  $A$  hasta  $F$  y se ha denotado el flujo de tránsito entre las intersecciones adyacentes por las variables  $x_1$  hasta  $x_7$ . El problema consiste ahora en minimizar  $x_4$  sujeta a las restricciones del problema.

Para encontrar estas restricciones veamos, por ejemplo, la intersección  $B$ . El tránsito que fluye a la intersección  $B$  es, según el mapa,  $x_2 + x_5$ ; mientras que el tránsito que sale de la intersección  $B$  es  $x_4 + 100$ . Suponiendo que el tránsito no se acumula en la intersección  $B$ , el tránsito de "entrada" debe ser igual al tránsito de "salida". Así se obtiene la ecuación

$$x_2 + x_5 = x_4 + 100$$

o bien

$$x_2 - x_4 + x_5 = 100$$

A partir de este análisis en cada intersección, se obtiene el siguiente sistema de seis ecuaciones en siete incógnitas:

en $A$	$x_1$			$-x_4$			$= -200$
en $B$		$x_2$		$-x_4 + x_5$			$= 100$
en $C$			$x_3$	$+x_5$			$= 700$
en $D$	$x_1$					$-x_6$	$= 100$
en $E$		$x_2$				$-x_6 + x_7$	$= 600$
en $F$			$x_3$			$+x_7$	$= 900$

## 1.8 PLANTEAMIENTO DE MODELOS DE REDES COMO PROBLEMAS DE PROGRAMACIÓN LINEAL

Consideraremos cuatro tipos de problemas de redes: problemas de transbordo, de ruta más corta, de flujo máximo y de transporte. Analizaremos también el planteamiento de problemas de PERT/CPM y problemas de asignación de personal como problemas de redes. El tema más importante de esta sección es que *todos* estos problemas de flujo pueden plantearse de manera similar como problemas de flujo de costo mínimo y pueden resolverse como problemas de programación lineal. Para varios de estos problemas existen algoritmos especiales que pueden ser más eficientes que el método simplex de programación lineal, pero debido a que se hará hincapié en los planteamientos, aquí no se analizan esos algoritmos.

Enseguida se presentan y analizan cada uno de los problemas de redes y después se plantea cada uno de ellos.

### PROBLEMA DE TRANSBORDO

Como segundo ejemplo de un problema de transbordo, considere el siguiente: la Compañía Petrolera Alianza tiene un solo campo petrolero desde donde envía todo el petróleo a través de un oleoducto a uno de dos centros de embarque, en donde se almacena en buques tanque para su envío a refinerías de Estados Unidos.

La oferta diaria en el campo es de 2,000 barriles. Deben considerarse los costos del oleoducto, los costos de embarque y las cantidades de petróleo que pueden enviarse a través de los oleoductos. Los costos del oleoducto y las capacidades diarias de este se muestran en la tabla 1.4.

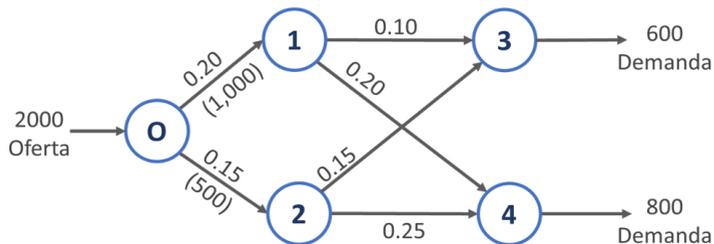
**TABLA 1.4** Compañía Petrolera Alianza. Costos y capacidades de los ductos

Instalación de envío	Costo por barril	Capacidad del oleoducto (en barriles)
1	\$0.20	1 000
2	\$0.15	500

En la tabla 1.5 se presentan los costos de embarque de cada estación de embarque a cada refinería y las demandas diarias de las refinerías. En la figura 1.14 se plantea este problema en forma de red.

**TABLA 1.5** Compañía Petrolera Alianza. Costos de transporte y demandas

Costo de transporte por barril		Refinería	Demanda diaria
Del centro 1	Del centro 2		
0.10	0.15	1	600
0.20	0.25	2	800



**Figura 1.14.** Red de la Compañía Petrolera Alianza

En el planteamiento de red, el nodo 0 es el campo petrolero, los nodos 1 y 2 son los nodos de embarque y los nodos 3 y 4 representan las refinerías de Nuevo León e Hidalgo, respectivamente. Las capacidades de los oleoductos se muestran en los arcos entre paréntesis y los costos se ilustran encima de los arcos.

Este problema puede plantearse en forma de programación lineal sí

$$x_{ij} = \text{barriles enviados del nodo } i \text{ al nodo } j$$

Entonces, el problema es

$$\text{Min } z = 0.2x_{01} + 0.15x_{02} + 0.10x_{13} + 0.2x_{14} + 0.15x_{23} + 0.25x_{24}$$

Sujeto a:

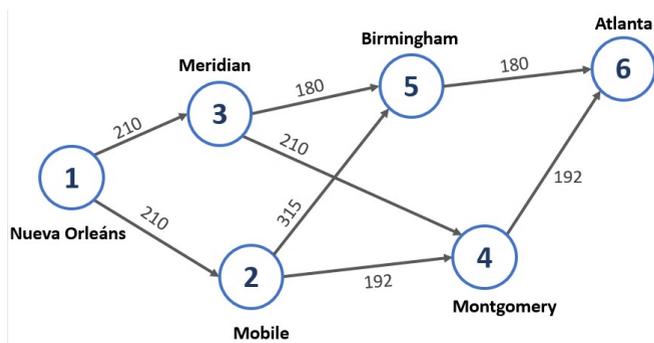
	$x_{01}$	$+x_{02}$					$\leq 2000$
	$x_{01}$		$-x_{13}$	$-x_{14}$			$= 0$
		$x_{02}$			$-x_{23}$	$-x_{24}$	$= 0$
			$x_{13}$		$+x_{23}$		$= 600$
				$x_{14}$		$+x_{24}$	$= 800$
	$x_{01}$						$\leq 1000$
		$x_{02}$					$\leq 500$
							todas las $x_{ij} \geq 0$

Este problema difiere un poco del de General Motors ya que existen capacidades en algunos de los arcos. Estas capacidades resultan ser restricciones de cota superior. Las restricciones de la oferta aparecen como una restricción de menor o igual, en vez de ser una restricción de igualdad, puesto que la demanda total es inferior a la oferta disponible.

### Problema de la ruta más corta

Si se define una red de manera que los coeficientes de cada arco sean no negativos (tal como medidas de distancia), entonces podríamos estar interesados en encontrar la ruta más corta entre dos nodos de la red. A este problema se le denomina *problema de la ruta más corta*.

Como ejemplo del problema de ruta más corta, consideremos el problema de viajar en automóvil desde Nueva Orleans a Atlanta en el tiempo más corto, dentro de lo legal. Las carreteras que enlazan estas dos ciudades forman la red que se muestra en la figura 1.15 en donde las "distancias" son los tiempos de viaje en automóvil dados en minutos.



**Figura 1.15.** Rutas del manejo de Nueva Orleans a Atlanta.

Observe que, con el objeto de plantear este problema en forma de problema de costo mínimo, puede elegirse en cualquier nodo solo el camino a través de un arco.

Lo anterior implica que es necesario tener los siguientes flujos en los arcos:

$$x_{ij} = \begin{cases} 1 & \text{si se viaja a través de la carretera entre la ciudad } i \\ & \text{y la ciudad } j \\ 0 & \text{de otra manera} \end{cases}$$

Si se recorre una ruta  $(i, j)$ , esto significa que no puede utilizarse ninguna otra que parta de la ciudad  $i$ . Por ejemplo, si se viaja por la carretera de Nueva Orleans a Mobile, entonces el flujo entre esas ciudades será igual a 1 y  $x_{13}$  será cero.

Es posible satisfacer estas condiciones utilizando en la red un flujo imaginario de una unidad. Este flujo parte del nodo de salida u *origen* y llega al nodo final o *terminal*. En otras palabras, existe un *suministro* de una unidad en el origen y una *demanda* de una unidad en el nodo terminal. En nuestro ejemplo, habría un suministro (u oferta) de una unidad en Nueva Orleans y una demanda por esa misma unidad en Atlanta..

La última pregunta que permanece pendiente es: ¿Qué costos deben utilizarse en este problema de flujo? Para responder esta pregunta observe que si  $x_{ij} = 1$ , entonces será necesario viajar de los nodos  $i$  al  $j$ . Si se denotan estas distancias mediante  $d_{ij}$  y si se utiliza la ruta entre  $i$  y  $j$ , entonces el costo para esa ruta se convierte en  $d_{ij}x_{ij}$ . Dado que  $x_{ij}$  es cero o uno, el costo para cualquier ruta será  $d_{ij}$  o cero. Por esto, podemos utilizar las distancias  $d_{ij}$ , como los costos para el problema de flujo de costo mínimo. Ahora, podemos plantear este problema de la siguiente manera:

$$\text{Min } z = 210x_{12} + 210x_{13} + 192x_{24} + 315x_{25} + 210x_{34} + 180x_{35} + 192x_{46} + 180x_{56}$$

Sujeto a:

New Orleans	$x_{12}$	$+x_{13}$							= 1
Mobile	$x_{12}$		$-x_{24}$	$-x_{25}$					= 0
Meridian		$x_{13}$			$-x_{34}$	$-x_{35}$			= 0
Montgomery			$x_{24}$		$+x_{34}$		$-x_{46}$		= 0
Birmingham				$x_{25}$		$+x_{35}$		$-x_{56}$	= 0
Atlanta	$x_{01}$						$x_{46}$	$+x_{56}$	= 1

$$x_{ij} \geq 0 \text{ para toda } i \text{ y toda } j$$

Para propósitos de análisis, hemos anotado con qué ciudad se relaciona cada restricción. La restricción de Nueva Orleans establece que puede utilizarse la carretera que va a Mobile o la que va a Meridian, pero no ambas. Sabemos que las soluciones de programación lineal

para problemas de redes son enteras, por lo cual estamos seguros de que  $x_{12} = 1$  o 0 y que  $x_{13} = 1$  o 0, al mismo tiempo que la restricción impone que  $x_{12} = 1$  ó  $x_{13} = 1$ , pero no ambas.

Las restricciones de Mobile, Meridian, Montgomery y Birmingham requieren todas que el flujo que llega a esos nodos (ciudades) sea igual al flujo que sale, puesto que aún no existe demanda en ninguno de ellos. La restricción para Atlanta exige llegar a esta ciudad ya sea de Birmingham o de Montgomery, obligando a que la suma de los flujos sea igual a 1.

En un problema de ruta más corta puede haber arcos dirigidos y arcos no dirigidos. En nuestro ejemplo solo se tienen arcos dirigidos. Para el caso de arcos no dirigidos, sería necesario tener una variable de  $i$  a  $j$  y otra de  $j$  a  $i$ . No sería difícil modificar el planteamiento para manejar las variables en ambas direcciones.

Puesto que ya hemos formulado el problema de la ruta más corta como problema de flujo de costo mínimo, podemos entonces utilizar un programa de computación especial de programación lineal, para resolver el problema.

El lector debe percatarse de que existen diversas variaciones del problema de la ruta más corta, y que la que se ha presentado aquí es sólo una de ellas. Las dos variaciones que existen son: (1) encontrar la ruta más corta entre algún nodo y cada uno de los otros nodos de la red; (2) encontrar la ruta más corta entre cualquier par de nodos de la red. Al igual que con la versión que se presenta aquí, es posible formular como problemas de flujo de costo mínimo estos otros problemas de ruta más corta. Existen también algoritmos de aplicación especial para las otras versiones.

1

2

3

4

5

6

7

8

## Problema de Pert/CPM

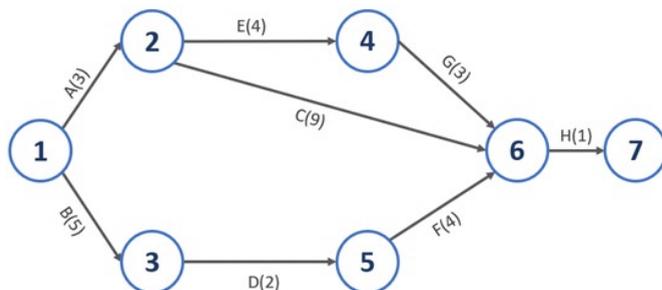
El objetivo de estos modelos es encontrar la secuencia de actividades o tareas que requieren del *mayor* tiempo para su terminación. A esta secuencia de trabajo se le denomina *ruta crítica*. Como ejemplo de un análisis de red de actividades, considérese la siguiente situación.

La Facultad de Administración de la UAM está planeando una conferencia para ejecutivos de negocios. En la tabla 1.6, se listan las actividades que deben llevarse a cabo antes de la conferencia y también se presentan los tiempos estimados para terminar cada una de las labores.

**TABLA 1.6** Lista de actividades con tiempo estimado

Actividad	Descripción	Tiempo estimado semanas	Actividades precedentes
A	Elaborar el programa	3	-
B	Recopilar lista de asistentes	5	-
C	Contactar a oradores para que asistan	9	A
D	Elaborar folleto para el programa	2	B
E	Hacer arreglos físicos	4	A
F	Enviar los folletos	4	D
G	Preparar programa de las conferencias	3	E
H	Detalles de último minuto	1	C, G, F

Este problema de planeación también se muestra en un formato de redes en la figura 1.16 y se ilustran en forma directa en los arcos los símbolos que representan a las actividades y el tiempo estimado para terminar la actividad; también se enumeraron los nodos.



**Figura 1.16.** Red de planeación de conferencias

Dado que estamos intentando descubrir cuál es la ruta más larga en esta red, tenemos la situación opuesta al problema de la ruta más corta. No obstante, este problema puede planearse en forma similar. En este caso, los tiempos estimados para cada arco se utilizan como "distancias", que van a maximizarse en vez de minimizarse. De nuevo se incluye un suministro imaginario de una unidad de flujo en el primer nodo y una demanda imaginaria de una unidad en el último. El planteamiento de programación lineal de este problema es:

$$\text{Min } z = 3x_{12} + 5x_{13} + 4x_{24} + 9x_{26} + 2x_{35} + 3x_{46} + 4x_{56} + 1x_{67}$$

Sujeto a:

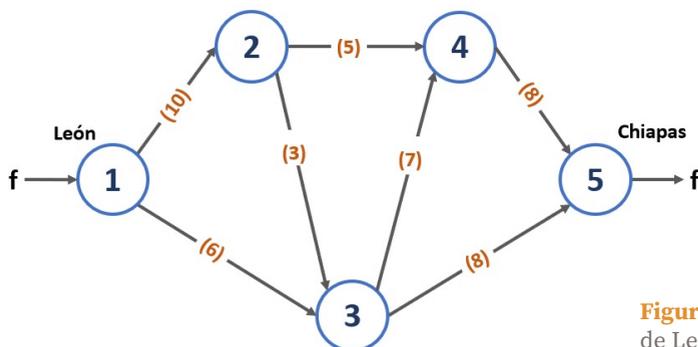
	$x_{12}$	$+x_{13}$							$=1$
	$x_{12}$		$-x_{24}$	$-x_{26}$					$=0$
		$x_{13}$			$-x_{35}$				$=0$
			$x_{24}$			$-x_{46}$			$=0$
					$x_{35}$		$-x_{56}$		$=0$
				$x_{26}$		$+x_{46}$	$+x_{56}$	$-x_{67}$	$=0$
								$x_{67}$	$=1$

$$x_{ij} \geq 0 \text{ para toda } i \text{ y toda } j$$

## Problema de flujo máximo

En los problemas anteriores estábamos interesados en los valores que se generan a través de cierto flujo que pasa por una red. Este valor puede estar dado en términos de dinero, distancia, tiempo o alguna otra medida. Existen problemas en los que el valor del flujo no es tan importante como la cantidad del flujo que pasa a través de la red. Los gasoductos y las líneas de transmisión de electricidad son ejemplos de esta situación. Los problemas en los que interesa determinar el flujo máximo que pasa a través de una red se denominan *problemas de flujo máximo*.

Para estudiar este tipo de problemas, es necesario suponer que existen restricciones de capacidad en los arcos. Si no fuera así, el flujo máximo que pasaría a través de la red sería infinito. Como ejemplo de problema de flujo máximo considérese el problema de enviar gas natural desde un campo de gas que se encuentra en León hasta Chiapas, a través de una red de gasoductos. Esta red se muestra en la figura 1.17. Los valores que se encuentran encerrados en semicírculos en cada arco representan las restricciones de capacidad en millones de pies cúbicos de gas por hora.



**Figura 1.17.** Red de León a Chiapas

En esta figura se muestra también una cantidad desconocida de flujo,  $f$ , que entra en el gasoducto en el nodo 1 (el campo de gas) y que sale del gasoducto en el nodo 5 (la terminal de Chiapas). Utilizando este flujo,  $f$ , puede plantearse este problema de la siguiente manera:

Maximizar:  $f$

Sujeto a:

	$x_{12}$	$+x_{13}$						$=f$
	$x_{12}$		$-x_{23}$	$-x_{24}$				$=0$
		$x_{13}$	$+x_{23}$		$-x_{34}$	$-x_{35}$		$=0$
				$x_{24}$	$+x_{34}$		$-x_{45}$	$=0$
						$x_{35}$	$+x_{45}$	$=f$
	$x_{12} \leq 10,$	$x_{13} \leq 6,$	$x_{23} \leq 3,$	$x_{24} \leq 5,$	$x_{34} \leq 7,$	$x_{35} \leq 8,$	$x_{45} \leq 8$	

$$x_{ij} \geq 0 \text{ para toda } i \text{ y toda } j$$

Este planteamiento no se ajusta a nuestra formulación estándar de programación lineal de flujo de costo mínimo, puesto que el flujo que se desconoce,  $f$ , aparece tanto como variable de la función objetivo, como en forma de valor de lado derecho de las restricciones. Si se plantea de esta manera no es posible utilizar el método de flujo de costo mínimo para resolverlo.

Para evitar esta dificultad, en primer lugar, se elimina el flujo  $f$  y se introduce un arco artificial o ficticio que conecta los nodos 5 y 1. El objetivo se convierte entonces en maximizar el flujo que pasa por este arco ficticio. Maximizar el flujo que regresa del nodo 5 al nodo 1 por un arco ficticio que no tiene capacidad dará la cantidad de flujo que va del nodo 1 al nodo 5 a lo largo de la red de capacidades. En la figura 1.18 se muestra la red de gasoductos, incluyendo el arco de regreso.

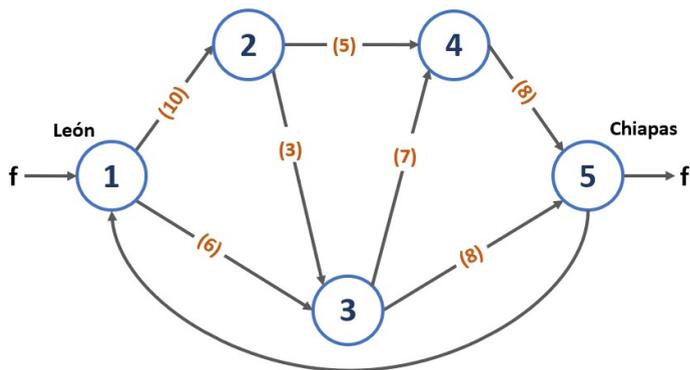


Figura 1.18. Red modificada de los gasoductos

Ahora, utilizando esta nueva red de gasoductos, se tiene un planteamiento modificado, en donde el objetivo es maximizar  $x_{51}$ :

Maximizar:  $x_{51}$

Sujeto a:

	$x_{51}$	$+x_{12}$	$-x_{13}$					$=0$
		$x_{12}$		$-x_{23}$	$-x_{24}$			$=0$
			$x_{13}$	$+x_{23}$		$-x_{34}$	$-x_{35}$	$=0$
					$x_{24}$	$+x_{34}$		$-x_{45} = 0$
	$-x_{51}$						$+x_{35}$	$+x_{45} = 0$
		$x_{12} \leq 10,$	$x_{13} \leq 6,$	$x_{23} \leq 3,$	$x_{24} \leq 5,$	$x_{34} \leq 7,$	$x_{35} \leq 8,$	$x_{45} \leq 8$

$$x_{ij} \geq 0 \text{ para toda } i \text{ y toda } j$$

Ahora queda planteado el problema de flujo máximo en forma estándar de programación lineal de redes, excepto que no existen ofertas ni demandas. Los problemas de este tipo se denominan *redes circulares*.

## Problema de transporte

El denominado *problema de transporte* es un caso especial del problema de transbordo, en el que todos los nodos son o fuentes (nodos de oferta) o destinos (nodos de demanda). En un problema de transporte no existen nodos de transbordo. Dado que es posible dividir el problema de transporte en dos conjuntos diferentes de nodos, es un problema de red *bipartita*. Para continuar analizando los problemas de transporte utilizaremos de nuevo un ejemplo.

Considere el caso de la Cervecería Modelo. Esta empresa elabora una cerveza que se distribuye a nivel nacional a partir de dos fábricas de cerveza, una en cada una de las dos costas de México (en Jalisco y Veracruz). La cerveza se envía a cuatro mayoristas que se encargan de la distribución subsecuente (en Chihuahua, Guerrero, Nuevo León y Yucatán), por lo que la Modelo se ocupa solo de la distribución a los mayoristas. Los costos de distribución, por conjuntos de 100 cajas que se envían a cada mayorista, se presentan en la tabla 1.7 junto con la oferta mensual de cada fábrica y la demanda mensual de cada mayorista.

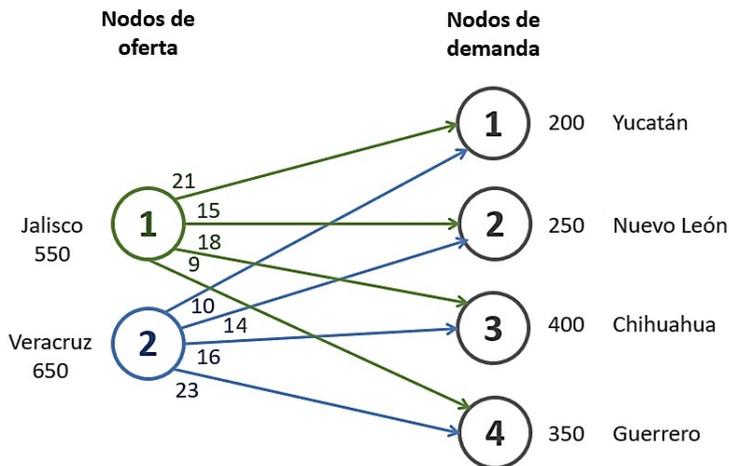
**TABLA 1.7** Costos de distribución para la Modelo.

Fábrica de Cerveza	Yucatán	N. León	Chih	Gro.	Oferta (Cientos de cajas)
Jalisco	\$21	\$15	\$18	\$9	550
Veracruz	\$10	\$14	\$16	\$23	650
Demanda (cientos de cajas)	200	250	400	350	

Si se representa este problema en forma de red, aparecerá según se muestra en la figura 1.19. Dado que el flujo de un nodo de oferta solo va a un nodo de demanda, se codificará el sistema común de numeración de los nodos para numerar los nodos de oferta en forma independiente de los nodos de demanda.

Observe que esto dará como resultado que haya nodos que se denotan como  $x_{11}$ , pero dado que los nodos de oferta están conectados solo a los nodos de demanda, no existe ambigüedad con respecto a qué arco identifica a  $x_{11}$ .

**Figura 1.19.** Red de transporte para la Modelo



El planteamiento de programación lineal del problema de transporte es muy similar al de los problemas de transbordo que se analizaron al inicio del capítulo. En este caso habrá 8 variables, una para cada arco, y 6 restricciones, una para cada nodo.

$$\text{Min } z = 21x_{11} + 15x_{12} + 18x_{13} + 9x_{14} + 10x_{21} + 14x_{22} + 16x_{23} + 23x_{24}$$

Sujeto a:

	$x_{11}$	$+x_{12}$	$+x_{13}$	$+x_{14}$					= 550
					$+x_{21}$	$+x_{22}$	$+x_{23}$	$+x_{24}$	= 650
	$x_{11}$				$+x_{21}$				= 200
		$x_{12}$				$+x_{22}$			= 250
			$x_{13}$				$+x_{23}$		= 400
				$x_{14}$				$+x_{24}$	= 350

$$x_{ij} \geq 0 \text{ para toda } i \text{ y toda } j$$

Es necesario señalar varios detalles acerca de este planteamiento. En primer lugar, las primeras dos restricciones imponen que la cantidad que se envía sea igual a la cantidad disponible. Se utilizan aquí restricciones de igualdad debido a que la oferta total es igual a la demanda total, por lo que debe transportarse la totalidad de la oferta. A un problema de este tipo se le denomina *problema en equilibrio de transporte*.

En segundo lugar, las siguientes 4 restricciones exigen que la cantidad que llega a cada nodo de demanda sea igual a la de demanda de ese nodo. Aquí se utilizan restricciones de igualdad por la misma razón que antes. Debe observarse que, para un problema en equilibrio de transporte, una de las restricciones no es necesaria. Si la oferta total que se transporta es de 1,200 y los primeros 3 nodos de demanda absorben 850 unidades, entonces el último nodo de demanda debe absorber 350 unidades, y esto hace que la última restricción resulte redundante y pueda eliminarse. Si se hace esto entonces habrá, para este problema que tiene dos nodos de oferta o suministro y cuatro de demanda,  $2 + 4 - 1 = 5$  restricciones.

En general, un problema que tiene  $m$  nodos de oferta y  $n$  nodos de demanda tendrá  $m + n - 1$  restricciones. Este es un resultado importante puesto que siempre habrá el mismo número de variables básicas que restricciones.

En término de planteamiento de redes original, el hecho de que existan  $(m + n - 1)$  variables básicas implica que habrá  $(m + n - 1)$  arcos en el árbol abierto que represente este problema. En otras palabras, en cualquier solución no más de  $m + n - 1$  arcos (rutas) de un problema de transporte serán positivos.

El resultado es:

De Jalisco a Nuevo León	200 (cientos de cajas)
De Jalisco a Chihuahua	0
De Jalisco a Guerrero	350 (cientos de cajas)
De Veracruz a Yucatán	200 (cientos de cajas)
De Veracruz a Nuevo León	50 (cientos de cajas)
De Veracruz a Chihuahua	400 (cientos de cajas)
De Veracruz a Guerrero	0

No es necesario utilizar un paquete de programación lineal de redes para resolver un problema de transporte tan pequeño como ejemplo. Para ello se ha adaptado una versión del método simplex que se denomina **método del cruce del arroyo** (o de la piedra de paso: “*stepping stone*”) y se analiza en detalle en otro capítulo.

### Problema de asignación

El ejemplo final de problemas de redes que consideramos aquí es el *problema de asignación de personal* (o en forma más simple, el problema de asignación). Este problema se refiere a la asignación de agentes a trabajos en forma tal que se minimicen los costos de esa asignación. En calidad de ejemplo, considérese la situación que debe manejar el entrenador Ricardo Martínez del Equipo de Natación de la Acuática “El Pez Feliz”.

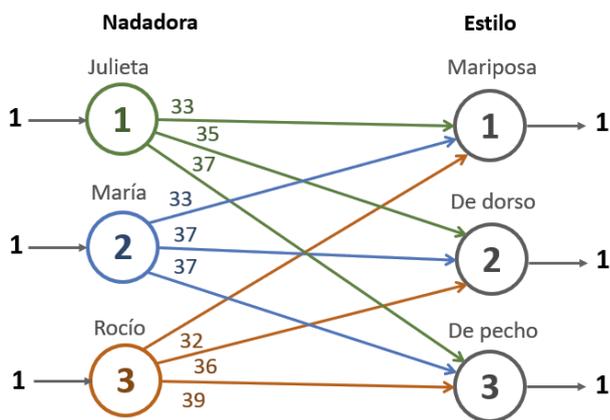
El entrenador está intentando organizar el mejor equipo de relevo de mujeres para los 200 metros. Tiene 4 muchachas en el equipo: Luisa, Julieta, María y Rocío. Luisa solo nada el estilo libre, por lo que no hay problema respecto de esa parte del equipo. Cada una de las otras tres

chicas puede nadar en cualquiera de los otros tres estilos: mariposa, dorso y pecho. Entonces la cuestión es cuál de ellas debe nadar en qué estilo. Los tiempos de cada una de las nadadoras en cada uno de los estilos se muestran en la tabla 1.8.

**TABLA 1.8** Tiempos por estilo en segundos del equipo de “El Pez Feliz”

Nadadora	Mariposa	Dorso	Pecho
Julieta	33	35	37
María	33	37	37
Rocío	33	36	39

Este problema puede plantearse en forma de red considerando tres nodos fuente, cada uno con un suministro de una unidad, y tres nodos destino, cada uno con una demanda de una unidad. Los nodos fuente representan los agentes disponibles (nadadoras) y los nodos destino, los trabajos (estilos). La red para este ejemplo se muestra en la figura 1.20.



**Figura 1.20.** Red de asignación para el equipo de natación de “El Pez Feliz”

Es fácil de observar que el problema de asignación no es otra cosa que un caso especial de red de transporte en el que todas las ofertas o suministros y las demandas son iguales a 1 y en la que el número de nodos

de oferta es igual al número de nodos de demanda. Puesto que esto es cierto, el procedimiento de programación lineal es muy similar:

$$\text{Min } z = 33x_{11} + 35x_{12} + 37x_{13} + 33x_{21} + 37x_{22} + 37x_{23} + 32x_{31} + 36x_{32} + 39x_{33}$$

Sujeto a:

$x_{11}$	$+x_{12}$	$+x_{13}$							$=1$
			$x_{21}$	$+x_{22}$	$+x_{23}$				$=1$
						$x_{31}$	$+x_{32}$	$+x_{33}$	$=1$
$x_{11}$			$+x_{21}$			$+x_{31}$			$=1$
	$x_{12}$			$+x_{22}$			$+x_{32}$		$=1$

$$x_{ij} \geq 0 \text{ para toda } i \text{ y toda } j$$

Se ha omitido la última restricción de este planteamiento puesto que sería redundante, al igual que en el planteamiento de transporte. Una característica especial de este problema es que  $x_{ij} = 0$  ó  $1$ .

En otras palabras:

$$x_{ij} = \begin{cases} 1 & \text{si a la nadadora } i \text{ se le asigna el trabajo (estilo) } j \\ 0 & \text{de otra manera} \end{cases}$$

Estamos seguros de obtener este resultado porque en cada nodo de oferta existe solo una unidad que pueda enviarse a lo largo de varios arcos.

## 1.9 NOTAS HISTÓRICAS

El estudio de los modelos de redes de flujo antecede al desarrollo de la programación lineal. Los primeros estudios en esta área los realizaron Kantoróvich (1939), Hitchcock (1941) y Koopmans (1947) considerando el problema de transporte como un caso especial del problema de flujo a costo mínimo.



Leonid Vitálievich Kantoróvich



Tjalling Charles Koopmans

Esos estudios proporcionaron una nueva luz en la estructura del problema y dieron nuevos métodos y algoritmos. El interés por las redes de flujo creció con el advenimiento del método simplex desarrollado por Dantzig en 1947, quien incluso especializó este algoritmo para el problema de transporte (ver Dantzig, 1951).

Durante los años cincuenta, las investigaciones comenzaron a mostrar un creciente interés en el problema de flujo a costo mínimo y sus especializaciones: el problema de la ruta más corta, el problema de flujo máximo y el problema de asignación; esto se debió principalmente a la importante aplicación de estos modelos en la vida real. Algunos autores desarrollaron algoritmos para resolver estos problemas. Dantzig, Ford y Fulkerson fueron pioneros en este aspecto. Mientras Dantzig se enfocó en los métodos basados en el simplex, Ford y Fulkerson desarrollaron algoritmos combinatorios primales duales. Los libros de Dantzig (1962) y de Ford



George Bernard Dantzig

y Fulkerson (1962) están presentes a través de la discusión de estas primeras contribuciones.

En los años siguientes, los problemas de flujo en red y sus generalizaciones emergieron como tópicos de gran interés en cientos de artículos y numerosos textos, así como en libros de referencia.

### 1.10 EJERCICIOS PROPUESTOS

1. Hay ocho cuadrados iguales ordenados en tres renglones, cada uno con dos cuadrados en el primer renglón, cuatro en el segundo y dos en el tercero. Los cuadrados de cada fila están arreglados en forma simétrica respecto al eje vertical. Se desea poner números distintos en los cuadrados, entre los límites de 1, 2, ..., 8, de modo que no haya dos cuadrados adyacentes vertical, horizontal o diagonalmente que tengan números consecutivos. Use la representación en red como vehículo hacia la solución de una forma sistemática.
2. Se deben transportar en un bote tres reclusos escoltados por tres guardias, de San Francisco a la penitenciaría de la isla de Alcatraz, para que purguen sus sentencias. El bote no puede transportar más de dos personas en cualquier dirección. Los reclusos les ganan con seguridad a los guardias si su número es mayor en cualquier momento. Formule un modelo de red para diseñar los viajes del bote de modo que se asegure la transferencia segura de los reclusos. Suponga que no se fugan si tienen oportunidad.
3. Una persona desea viajar un cierto día de una ciudad A hacia una ciudad B. la persona pretende llegar a la ciudad B a más tardar a las cuatro de la tarde y planea viajar a costo mínimo. Los únicos

vuelos que ofrecen las líneas de aviación para ir de A hacia B son vuelos con escalas cuyas salidas y llegadas, así como los costos se muestran a continuación:

No. de vuelo	Origen	Destino	Salida	Llegada	Costo
436	A	C	7:00	11:00	$C_1$
	C	D	12:10	13:00	$C_2$
	D	B	13:20	15:00	$C_3$
524	A	D	9:30	14:00	$C_4$
	D	C	14:15	15:00	$C_5$
	C	B	15:10	16:00	$C_6$
301	D	X	14:00	15:00	$C_7$
	X	B	15:00	16:00	$C_8$
409	A	N	7:00	11:00	$C_9$
	N	D	11:10	14:00	$C_{10}$
	D	B	14:10	15:00	$C_{11}$

En la planeación debe tomarse en cuenta que si se cambia de vuelo se requiere de una hora para llevar a cabo la interconexión, por ejemplo, si la persona toma el vuelo 524 y llega a la ciudad D a las 14:15 horas, no puede tomar el vuelo 409 de la ciudad D a B. Represente la red y formule la red de flujo a costo mínimo que representa la problemática.

4. *Planeación de producción.* DirectCo vende un artículo cuya demanda en los 4 meses venideros será 100, 140, 210 y 180 unidades, respectivamente. La empresa puede almacenar sólo la cantidad justa para abastecer la demanda de cada mes, o puede almacenar más y cumplir con la demanda de dos o más meses consecutivos. En el segundo caso se carga un costo de retención de \$ 1.20 por unidad en exceso por mes. DirectCo estima que los precios unitarios de compra durante los 4 meses siguientes serán de \$ 15, \$ 12, \$ 10

y \$14, respectivamente. Se incurre en un costo de preparación de \$200 cada vez que se coloca un pedido. La empresa desea desarrollar un plan de compras que minimice los costos totales de pedido, compra y retención del artículo en el almacén. Formule la red que representa la problemática y el problema como modelo de ruta más corta a través de la programación lineal.

5. Dibuje la red asociada al siguiente planteamiento de programación lineal.

$$\text{Min } z = 3x_{12} + 5x_{14} + 2x_{23} + 4x_{24} + 2x_{34}$$

Sujeto a:

$x_{12}$	$+x_{14}$				$= 100$
$x_{12}$		$-x_{23}$	$-x_{24}$		$= 50$
		$x_{23}$		$-x_{34}$	$= 0$
	$x_{14}$		$+x_{24}$	$+x_{34}$	$= 50$
					$x_{ij} \geq 0$

# Conceptos básicos de redes

## 2.1 INTRODUCCIÓN

En este capítulo se presentan conceptos fundamentales para poder desarrollar diferentes algoritmos de redes, así como la representación matricial de una red que nos permite encontrar árboles de expansión y que además muestra la relación entre el teorema fundamental del álgebra lineal y las estructuras de red. Asimismo, la dualidad entre potenciales y tensión y su relación con la programación lineal, que nos introducirá a otro concepto dual: el de trayectorias y cortes que se verá en el capítulo 3.

## 2.2 REPRESENTACIÓN MATRICIAL DE UNA RED

Para responder a la pregunta de cómo se puede representar numéricamente una red se introduce aquí el concepto de matriz de incidencia como sigue. Sea  $G$  una red, la matriz de incidencias nodos-arcos  $E$  es una matriz de  $m \times n$  donde  $m$  es el número de nodos y  $n$  es el número de arcos cuyos elementos  $e(i, j)$  son:

$$e(i, j) = \begin{cases} +1 & \text{si el arco } j \text{ sale del nodo } i \\ -1 & \text{si el arco } j \text{ entra al nodo } i \\ 0 & \text{en otro caso} \end{cases}$$

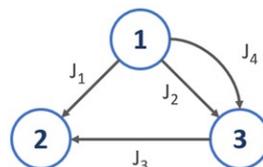
Una forma de recordar los signos de esta definición es que “una flecha siempre va desde donde está hasta donde no está”.

## Ejemplo 2.1

En la figura 2.1 se presenta la red asociada a la siguiente matriz nodos-arcos, donde los nodos están dados en los renglones y los arcos en las columnas.

$$E = \begin{array}{c} \begin{array}{cccc} & J_1 & J_2 & J_3 & J_4 \\ 1 & \begin{bmatrix} 1 & 1 & 0 & 1 \end{bmatrix} \\ 2 & \begin{bmatrix} -1 & 0 & -1 & 0 \end{bmatrix} \\ 3 & \begin{bmatrix} 0 & -1 & 1 & -1 \end{bmatrix} \end{array} \end{array}$$

**Figura 2.1**  
Red asociada  
a la matriz E



Si existen rizos la matriz nodos-arcos no sirve, pues un rizo entra y sale del mismo nodo.

$J_1, J_2, J_3$  son linealmente dependientes ya que  $J_1 + J_3 - J_2 = 0$ .

¿Qué significado tiene esto en la red?

Si se cambia de sentido un arco significa que se multiplica por  $-1$  la columna, entonces en este caso que los vectores sean linealmente dependientes significa que se forma un circuito.

Los árboles son aquellas estructuras que no tienen circuitos, por lo tanto, son linealmente independientes.

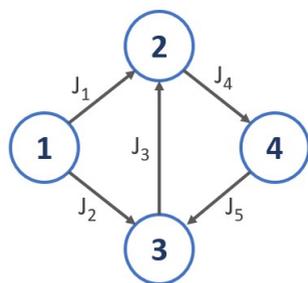
Además de la matriz de incidencia (o función de incidencia), existe una matriz de adyacencia  $Q$  de orden  $m \times m$  definida en  $N \times N$  de la siguiente manera:

$$q(i_1, j_2) = \begin{cases} 1 & \text{si existe un arco que parte de } i_1 \text{ y llegue a } i_2 \\ 0 & \text{en otro caso} \end{cases}$$

Por supuesto se tiene que  $G$  es una digráfica, esto es  $G \subset N \times N$

### Ejemplo 2.2

Encuentre la matriz de adyacencia nodos-nodos de la siguiente red:



**Figura 2.2** Red con matriz de adyacencia

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

Una matriz de  $m \times m$  puede ser la matriz adyacente de una red con  $m$  nodos si consiste solo de 0 y 1 y si no contiene rizados debe tener ceros en la diagonal principal y se define como se muestra en la figura 2.2.

El grado exterior de un nodo  $i$  es el número de arcos que salen de  $i$  por ejemplo:

$$\begin{aligned} g_e(1) &= 2 & g_e(3) &= 1 \\ g_e(2) &= 1 & g_e(4) &= 1 \end{aligned}$$

El grado interior de un nodo  $i$  es el número de arcos que entran al nodo  $i$ .

$$g_i(1) = 0$$

$$g_i(2) = 2$$

$$g_i(3) = 2$$

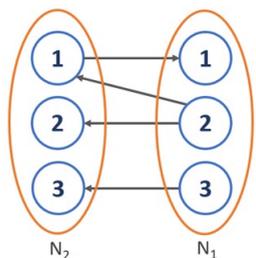
$$g_i(4) = 1$$

La suma de los grados de todos los arcos es par

$$\sum g_e + \sum g_i = 5 + 5 = 10$$

## 2.3 REDES ESPECIALES

Una red  $G$  es **bipartita** si el conjunto de nodos  $N$  puede partitionarse en  $N_1, N_2$  con  $N_1 \cap N_2 = \Phi$  de manera que todo arco  $j \sim (i_1, i_2)$  es tal que  $i_1 \in N_1, i_2 \in N_2$  o  $i_2 \in N_1, i_1 \in N_2$



**Figura 2.3**  
Red Bipartita

**Nota:**

1. Las matrices nodos - nodos para gráficas no dirigidas son simétricas.
2. La matriz nodos - nodos de una red bipartita es de la forma:

$$q = \begin{matrix} & I_1 & I_2 \\ I_1 & \begin{bmatrix} 0 & \# \\ \# & 0 \end{bmatrix} & \end{matrix} \text{ que es simétrica.}$$

## Lema de handshaking

En una gráfica, la suma de los grados de todos los nodos es igual al doble del número de arcos.

### Demostración

Ya que cada arco tiene dos extremos, debe contribuir exactamente dos veces a la suma de los grados.

Este lema apareció por primera vez, aunque en forma diferente en un artículo de Leonhard Euler (1707-1783), titulado *Solutio Problematis ad Geometrian Situs Pertinentis* (la solución de un problema referido a la geometría de posición) que data de 1736 y es ampliamente conocido como el primer artículo en la teoría de gráficas, donde también da la solución de los puentes de Königsberg.

Para gráficas que no son dirigidas simplemente se habla del grado de un vértice como el número de arcos que “llegan a él” (pueden entrar o salir)

### Redes simples

Es una gráfica que no contiene rizados ni arcos en paralelo con un nodo sumidero y un nodo fuente.

Un nodo fuelle es un nodo de donde únicamente salen arcos.

Un nodo sumidero es un nodo a donde únicamente entran arcos.

## Redes circulatorias

Una red es circulatoria si no existen nodos fuente o sumideros.

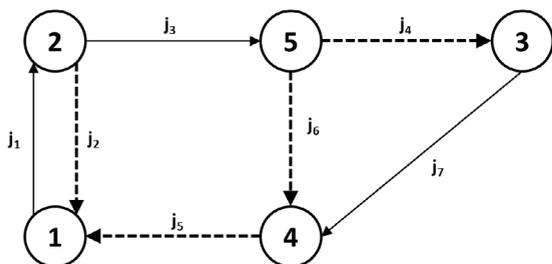
### Conectividad

Una cadena de longitud  $q$  (cardinalidad  $q$ ) es una sucesión de  $q$  arcos

$$L = \{j_1, \dots, j_q\}$$

tal que cada arco  $j_r$  de la sucesión ( $2 \leq r \leq q-1$ ) tiene un punto final en común con el arco  $j_{r-1}$  ( $j_{r-1} \neq j_r$ ) y un segundo punto final común con el arco  $j_{r+1}$  ( $j_{r+1} \neq j_r$ ).

El punto final  $i$  de  $j_1$  que no es adyacente a  $j_2$  y el punto final  $i'$  de  $j_q$  que no es adyacente a  $j_{q-1}$  son llamados los puntos finales de la cadena  $L$ . Decimos que la cadena  $L$  une los nodos  $i, i'$ . Así en la siguiente red:



**Figura 2.4** Ejemplo de cadena

$L = \{j_2, j_5, j_6, j_4\}$  es una cadena del nodo 2 al nodo 3.

Una cadena es elemental si no se pasa dos veces por un mismo nodo.

Un ciclo es una cadena cuyos puntos finales coinciden. Una trayectoria de longitud  $q$  (cardinalidad  $q$ ) es una sucesión de  $q$  arcos  $P = \{j_1, j_2, \dots, j_q\}$  con  $j_1 = (i_0, i_1), j_2 = (i_1, i_2), \dots, j_q = (i_{q-1}, i_q)$ . Es decir, una trayectoria es una cadena cuyos arcos están todos dirigidos en el mismo sentido.

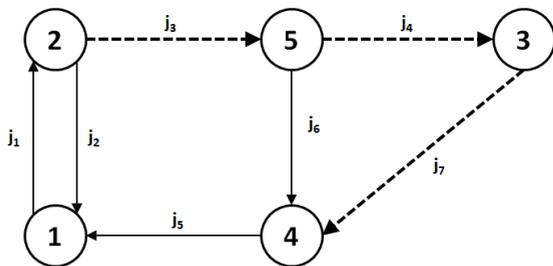


Figura 2.5 Ejemplo de trayectoria

$P = \{j_3, j_4, j_7\}$  es una trayectoria del nodo 2 al nodo 3.

El nodo  $i_0$  es el punto inicial de la trayectoria  $P$  y el nodo  $i_q$  es el punto terminal de la trayectoria  $P$ .

Así para la gráfica anterior  $P = \{j_1, j_3, j_4, j_7\}$  es una trayectoria del nodo 1 al nodo 4, y se puede describir como la sucesión de nodos  $\{1, 2, 5, 3, 4\}$ .

Un circuito es una trayectoria cuyos nodos finales coinciden. Una gráfica se llama conectada si para cualquier par de nodos  $(i_1, i_2)$  existe una cadena que une  $i_1$  con  $i_2$ . La relación

$$i_1 R i_2 = \begin{cases} i_1 = i_2 \\ \text{o existe una cadena que una } i_1 \text{ con } i_2 \end{cases}$$

Es una relación de equivalencia (reflexiva, simétrica y transitiva).

## 2.4 FLUJO Y DIVERGENCIA EN UNA RED

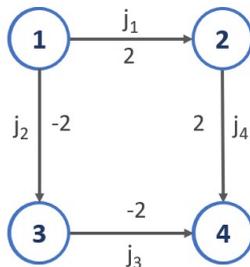
Un flujo en una red es una función real definida en  $A$  (donde  $A$  es el conjunto de arcos de una red  $G$ ).

$$x:A \rightarrow R$$

El valor  $x(j)$  es el flujo en el arco  $j$ , y se interpreta como la cantidad de material que fluye en el arco  $j$ .

### Ejemplo 2.3

Sea la red siguiente:



**Figura 2.6**  
Ejemplo de flujo  
en una red

El tipo de material es el mismo para todos los arcos. Para entender en forma más clara los conceptos es útil entender  $j$  como un canal y  $x(j)$  como el número de litros de agua por segundo que pasa por cada punto de  $j$  un flujo estacionario del nodo inicial  $i$  al nodo final  $i'$ . La cantidad que entra en  $i$  es la misma que sale por  $i'$ , pero puede ser positivo, negativo o cero dependiendo de la dirección física del flujo.

Para un orden fijo de arcos  $A = \{j_1, \dots, j_n\}$  un flujo  $x$  se puede representar como un vector  $(x_1, \dots, x_n)$  así para la figura anterior el flujo se puede representar como

$$x = (2, -2, -2, 2)$$

Así  $x$  es un “vector” indexado por el conjunto  $A$  (es decir un elemento del espacio  $R^A$ )

Existe otra notación para flujos que está vinculada con la representación de una red por su matriz de adyacencia. Como más tarde se requerirá que la red sea una digráfica (gráfica dirigida) cada arco se puede identificar por un par de nodos  $(i, i')$ . La idea es simplemente escribir  $x(i, i')$  en lugar de  $x(j)$  tomando  $x(i, i') = 0$  en los casos en que no exista arco entre  $i, i'$ .

Los flujos en  $G$  se representan como funciones en  $N \times N$  que pueden representarse matricialmente como:

$$x = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} 0 & 2 & -2 & 0 \\ 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & -2 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{matrix}$$

## Diurgencia

Las matrices de incidencia vuelven a entrar en escena cuando tratamos de analizar qué sucede a un flujo  $x$  en un nodo  $i$ , particularmente las entradas y salidas o pérdidas. El nodo  $i$  está representado en la matriz de incidencia  $E$  por el renglón  $i$ .

Las cantidades de material que salen de  $i$  están asociadas con los arcos  $j$  tales que  $x(j) > 0$  y  $e(i, j) = 1$ , o tales que  $x(j) < 0$  y  $e(i, j) = -1$ .

¿Qué significa esto?

Ejemplo 2.4

Sea la red de la figura 2.7.

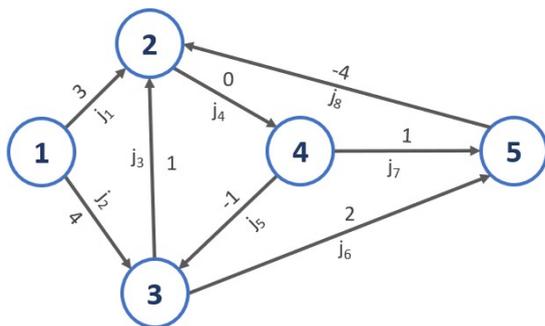


Figura 2.7 Red con flujo en los arcos

Para el nodo 3 se tiene en la matriz de incidencia:

$$E = \begin{matrix} & j_1 & j_2 & j_3 & j_4 & j_5 & j_6 & j_7 & j_8 \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & -1 & 1 & 0 & 0 & 0 & -1 \\ 0 & -1 & 1 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & -1 & 1 \end{bmatrix} \end{matrix}$$

El arco  $j_3$  sale del nodo 3 y tiene asociado un 1 es decir  $e(3,3)=1$  (recuerde que cada entrada de la matriz está representada por  $e(i, j)$ ) si se tiene un flujo  $x(j=3)=1$  (cantidad de material que sale del nodo 3) significa que el flujo "pasa" por el arco  $j_3$  en el sentido que lleva el arco y por lo tanto  $x(j)e(i, j) > 0$ . Si  $x(j) < 0$  por ejemplo  $x(j) = -4$  significa que el flujo recorre en sentido contrario el arco y por lo tanto  $e(i, j) = -1$  como es el caso de  $x(8) = -4$  y  $e(2, 8) = -1$  lo que implica que  $x(j)e(i, j) > 0$ .

Así la cantidad total que sale de  $i$  es la suma de todos los términos de la forma  $e(i, j)x(j)$  que son positivos.

En forma semejante las cantidades que llegan a  $i$  corresponden a los casos donde  $x(j) > 0$  y  $e(i, j) = -1$  por ejemplo:

En el nodo 5:  $x(6) = 2$  es decir la cantidad de flujo que llega al nodo 5 por el arco  $j_6$  es de 2 y  $e(5, 6) = -1$  en la matriz de incidencia por lo tanto  $x(6)e(5, 6) < 0$

O  $x(j) < 0$  y  $e(i, j) = 1$  como en el nodo 4,  $x(5) = -1$  es decir por el arco  $j_5$  pasa un flujo que va al nodo 4 con  $-1$  y  $e(4, 5) = 1$ .

La suma de todos los términos de la forma  $e(i, j)x(j)$  que son negativos es el negativo de la cantidad de flujo que llega a  $i$ .

De esta forma la suma de  $e(i, j)x(j)$  sobre todos los arcos  $j$  nos da el total de salidas de  $i$  menos el total de llegadas a  $i$ .

Esta cantidad es la divergencia del flujo en el nodo  $i$ , y se denota  $y(i)$ , así

$$y(i) = \sum_{j \in A} e(i, j)x(j) = \text{divergencia de } x \text{ en } i$$

Por ejemplo, de la figura anterior:

En el nodo 3 llegan 4 unidades por  $j_2$

salen	2	$j_6$
sale	1	$j_5$
sale	1	$j_3$

Este nodo tiene la particularidad de que:

# total de llegadas = # total de salidas.

Así  $y(3) = 0$  en el nodo 5 se tiene  $y(5) = -7$  esto significa que llegan 7 unidades más de las que salen y en general se dice que:

- Un nodo es fuente para el flujo  $x$  si  $y(i) > 0$
- Un nodo es sumidero para el flujo  $x$  si  $y(i) < 0$
- Si el flujo se conserva  $y(i) = 0$

Se llama  $y$  a la función de divergencia (vector) asociada con  $x$ , se escribe:

$$y = Ex = \text{div } x$$

$$y = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & -1 & 1 & 0 & 0 & 0 & -1 \\ 0 & -1 & 1 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 4 \\ 1 \\ 0 \\ -1 \\ 2 \\ 1 \\ -4 \end{bmatrix} = \begin{bmatrix} 7 \\ 0 \\ 0 \\ 0 \\ -7 \end{bmatrix}$$

En este caso para el flujo de la figura anterior usamos la matriz de incidencia y se calcula

así  $i_1$  es una fuente,  $i_5$  es sumidero, y el flujo se conserva en  $i_2, i_3$  e  $i_4$

El hecho de que las 7 unidades creadas en el nodo fuente  $i$  sean las 7 que se destruyen en el sumidero no es un accidente. La intuición física sugiere, y el álgebra lo confirma que la cantidad de flujo que se crea en la fuente es igual a la cantidad que se destruye en el sumidero.

Esto se expresa como el principio de la divergencia total:

$$\sum_{i \in N} y(i) = 0 \quad \text{para } y = \text{div } x$$

Esto se verifica de la siguiente forma:

$$\sum_{i \in N} y(i) = \sum_{i \in N} \sum_{j \in A} e(i, j) x(j) = \sum_{j \in A} \sum_{i \in N} e(i, j) x(j)$$

Pero

$$\sum_{i \in N} e(i, j) = 0 \quad \text{para toda } j \in A$$

Ya que cada columna de la matriz de incidencia contiene exactamente un 1 y un -1 y la suma da 0.

## Operaciones vectoriales

Dos flujos  $x$  y  $x^1$  se pueden sumar o superponer para producir un flujo resultante:

$$x'' : x''(j) = x(j) + x'(j) \quad \forall j \in A.$$

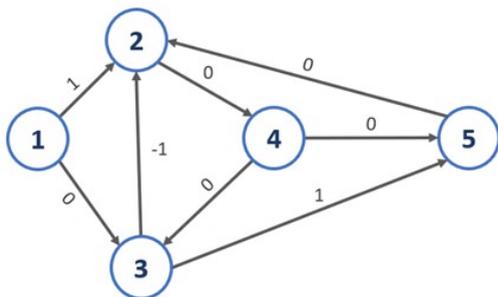
Asimismo, un flujo se puede multiplicar por un escalar  $x' = \alpha x$  lo que significa que

$$x'(j) = \alpha x(j) \quad \forall j \in A.$$

¿Que está sucediendo físicamente?

Considerando la red del ejemplo 2.4 se tiene:

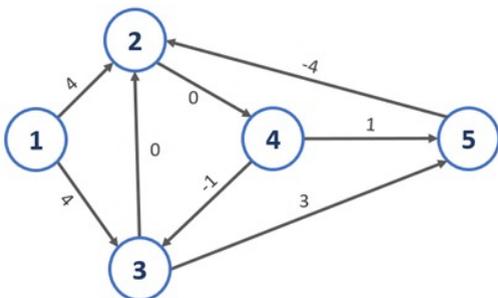
El flujo  $x'$  está representado en esta red, como una unidad de flujo pasando desde el nodo 1 hasta el nodo 5 pasando por los nodos 2 y 3.



**Figura 2.8**  
Representación del flujo  $x'$

Haciendo la suma se tiene:

$x'' = x + x'$  y se tienen 8 unidades en el nodo fuente y llegan 8 unidades al nodo sumidero. Note que el arco  $j_3$  es usado por  $x$  y  $x'$  pero no por  $x''$ .



**Figura 2.9**  
Flujo resultante de  $x + x'$

El flujo  $2x'$  podría representar dos unidades moviéndose en el mismo sentido que  $x'$  de  $i_1 \rightarrow i_2 \rightarrow i_3 \rightarrow i_5$ , mientras  $-x'$  podría representar una unidad moviéndose en forma inversa. Trivialmente se tienen las reglas

$$\begin{aligned} \operatorname{div}(x + x') &= \operatorname{div} x + \operatorname{div} x' \\ \operatorname{div}(\lambda x) &= \lambda \operatorname{div} x \end{aligned}$$

## 2.5 CIRCULACIONES

Un papel especial lo constituyen los flujos  $x$  en una red  $G$  tales que:

$$\operatorname{div} x = 0$$

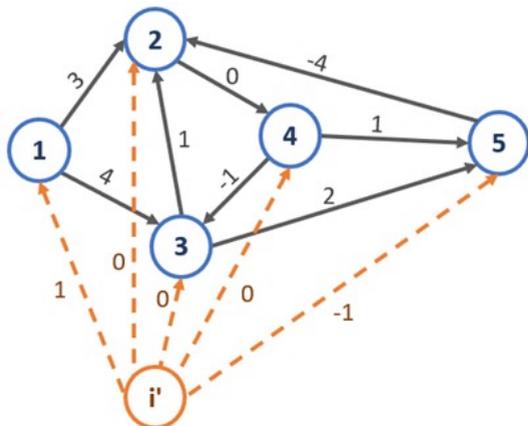
Esto es que  $x$  se conserva en cada nodo. Tales flujos se conocen como circulaciones. La suma y el producto por un escalar de una circulación es nuevamente una circulación. Así el conjunto de todas las circulaciones forma un subespacio de  $R^A$ : el espacio de las circulaciones  $C$ , claramente  $C$  es el espacio nulo de la matriz de incidencia  $E$ .

Una de las razones de porqué las circulaciones son importantes es que las discusiones teóricas a menudo pueden simplificarse en términos de ellas, esto debido a que todo flujo en una red  $G$  se puede identificar como una circulación de una red más grande. En el caso del flujo  $x$  del ejemplo 2.4 la idea se ilustra en la figura 2.10.

Formamos de una red  $G$  una nueva red  $\underline{G}$  aumentando un nuevo nodo  $i^1$  (el nodo de distribución) y un arco  $j_i \sim (i, i^1)$  (un arco de distribución) para cada uno de los nodos viejos  $i$ . Para esta nueva red los conjuntos de nodos y arcos se denotan  $\underline{N}$  y  $\underline{A}$ . Para cada flujo  $x$  en  $\underline{G}$  le corresponde un flujo  $x$  en  $G$  definido por:

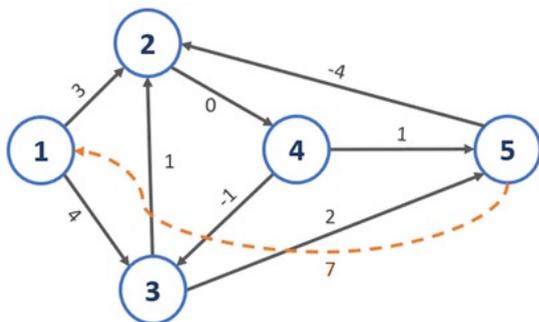
$$x(j) = x(j) \quad \text{para todos los arcos viejos}$$

$$\underline{x}(j_i) = y(i) \quad \text{para todos los arcos de distribución}$$



**Figura 2.10**  
Formación de una red circulatoria a través de agregar un nodo

Así  $\underline{x}$  es una circulación en  $\underline{G}$ . Inversamente, cada circulación en  $\underline{G}$  corresponde de esta forma a un flujo en  $G$ . También se tiene el caso en que se tienen identificados un nodo fuente y un nodo sumidero por lo cual sólo se agrega un arco como se ilustra en la figura siguiente:



**Figura 2.11**  
Formación de una red circulatoria a través de agregar un arco

El nuevo arco  $\underline{j}$  se conoce como un arco de suministro.

## 2.6 POTENCIALES Y TENSIONES

Un potencial  $u$  en una red  $G$  es una función real definida en el conjunto de los nodos  $N$ .

$$u: N \rightarrow R$$

El valor  $u(i)$  es llamado el potencial en el nodo  $i$ . Con un arco  $j \sim (i, i')$  se asocia la diferencia de potenciales

$$v(j) = u(i') - u(i) = \text{tensión a través de } j$$

El signo de la diferencia depende de la orientación del arco. Así se define la función tensión  $v$  en  $A$ , y se denomina el diferencial del potencial  $u$ .

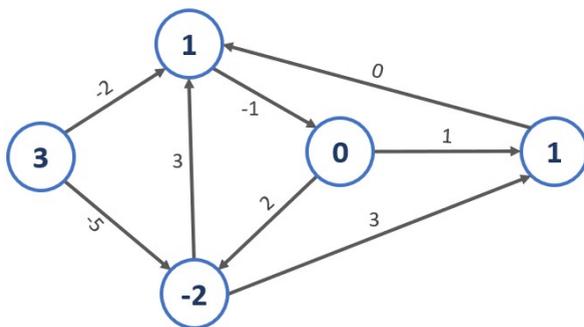
La tensión  $v(j)$  se puede escribir como

$$v(j) = -\sum_{i \in N} u(i) e(i, j)$$

o

$$v = -uE = \Delta u$$

Un ejemplo de potenciales y tensiones se muestra en la figura 2.11 donde los números en los nodos son los potenciales y los números en los arcos son las tensiones correspondientes.



**Figura 2.12**  
Potenciales y tensiones en una red

En general, llamamos a  $v$  un diferencial en  $G$  si  $v = \Delta u$  para algún potencial  $u$ . El conjunto de todos los diferenciales se preserva bajo la suma y la multiplicación por un escalar, y así como el espacio de las circulaciones forma un subespacio de  $R^A$ , y se llama el espacio de los

diferenciales y se denota por  $D$ . De esta forma  $D$  es el espacio de los renglones en la matriz de incidencia  $E$ , el rango de la transformación lineal  $u \rightarrow -uE$  de  $R^N \rightarrow R^A$ .

Es natural usar la notación siguiente de producto punto para elementos de  $R^N$  y  $R^A$ .

$$u \cdot y = \sum_{i \in N} u(i) y(i) \qquad v \cdot x = \sum_{j \in A} v(j) x(j)$$

Y se obtiene de esta manera la fórmula de conversión:

$$v \cdot x = -u \cdot y \qquad \text{si} \qquad y = \text{div } x, v = \Delta u$$

La validez de esta fórmula parte del hecho de que ambos lados se reducen a la expresión:

$$v(j) = -\sum u(i) e(i, j)$$

Entonces

$$v \cdot x = -\sum_{i \in N, j \in A} u(i) e(i, j) x(j) = -uEx = -u \cdot y$$

Una consecuencia inmediata de la fórmula de conversión es el hecho de que

$$v \cdot x = 0 \qquad \text{para todos } v \in D, x \in C, \text{ ya que:}$$

1.  $v \cdot x = -u \cdot y$  como y está asociada a un flujo  $x$  que es circulación, entonces  $y(i) = 0$  para todo  $i$ , lo que implica que  $v \cdot x = 0$

2. Si  $x \in C$ ,  $v \in D$  y  $C$  es el espacio nulo de  $E$  y  $D$  es el espacio renglón de  $E$  por la segunda parte del Teorema Fundamental de Algebra Lineal que dice:

Dada una matriz de  $m \times n$  se tiene

$$N(A) = (R(A^T))^{\perp}$$

$$R(A^T) = N(A)^{\perp}$$

Además

$$N(A^T) = (R(A))^{\perp}$$

$$R(A) = (N(A^T))^{\perp}$$

Y  $\dim(\text{espacio fila}) + \dim(\text{espacio nulo}) = \text{número de columnas}$ .

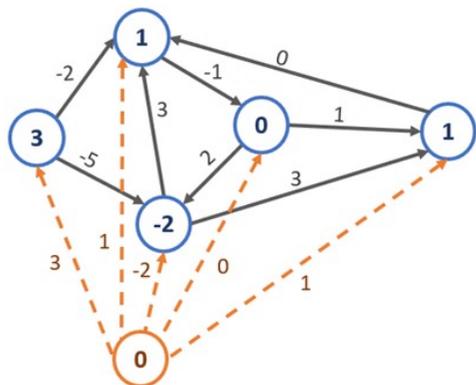
Entonces  $C$  y  $D$  son complementos ortogonales, lo que significa que el producto entre 2 elementos cualesquiera de ellos es igual a cero, además:

$$\dim C + \dim D = |A|$$

$$D = C^{\perp} = \{v \in R^A \mid vx = 0 \quad \forall x \in C\}$$

$$C = D^{\perp} = \{x \in R^A \mid vx = 0 \quad \forall v \in D\}$$

Así como un flujo en  $G$  se puede ver como parte de una circulación de la red aumentada  $\underline{G}$ , se puede también ver cada potencial en  $G$  como parte de un diferencial en  $\underline{G}$ : como se muestra en la figura siguiente:



**Figura 2.13**  
Diferencial en una red

Entonces dado cualquier potencial  $u$  en  $G$  con una tensión asociada  $v$  se define el potencial  $\underline{u}$  en  $\underline{G}$  como:

$$\underline{u}(i) = \begin{cases} u(i) & \forall i \text{ tal que } i \text{ es "nodo viejo"} \\ u(i) = 0 & \text{para el nodo de distribución } \underline{i} \end{cases}$$

La tensión  $v = \Delta u$  en  $\underline{G}$  satisface

$$\begin{aligned} v(j) &= v(j) && \forall j \text{ donde } j \text{ son "arcos viejos"} \\ v(j_i) &= u(i) && \text{para los arcos de distribución } j_i \end{aligned}$$

En términos de dualidad los variables definidas sobre los arcos se convierten en variables definidas sobre los nodos.

La tabla siguiente define un par de sistemas lineales duales.

<i>Primal</i>	<i>Dual</i>
Flujos - variables definidas sobre los arcos	Potencial - variables definidas sobre los nodos
divergencias - variables definidas sobre los nodos	tensión - variables definidas sobre los arcos

Los potenciales y las tensiones se pueden sumar o multiplicar por escalares, dos mismos potenciales pueden tener la misma tensión. En particular si:

$$u'(i) = u(i) + k \text{ donde } k \text{ es una constante, } \forall i \in N$$

E

Ejercicio

Considere las redes de la figura 2.14 (a y b). En la figura a se muestran los flujos y se pide calcular las divergencias. En la figura b se muestran los potenciales y se pide calcular las tensiones en los arcos.

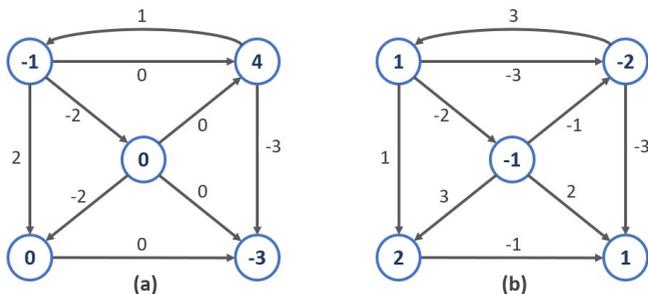


Figura 2.14 a) flujos en la red y b) potenciales en la red

2.7 FLUJOS ÓPTIMOS Y POTENCIALES

Para cada arco  $j \in A$  existe un intervalo  $c(j) \subset R$  y una función  $f_j: c(j) \rightarrow R$  de tal forma que para cada nodo  $i \in N$  existe un intervalo  $c(i) \rightarrow R$  y una función  $f_i: c(i) \rightarrow R$ . El problema de flujo óptimo es a grandes rasgos:

$$\min \sum_{j \in A} f_j(x(j)) + \sum_{i \in N} f_i(y(i)) \text{ sobre todos los flujos}$$

$$x \in R^A \text{ satisface } x(j) \in c(j) \forall j \in A$$

$$y(i) \in c(i) \forall i \in N \text{ donde } y = \text{div } x$$

En forma análoga se tiene el problema de potencial óptimo

$$\min \sum g_i(u(i)) + \sum g_j(u(j)) \quad \forall u \in R^N$$

Que satisfacen

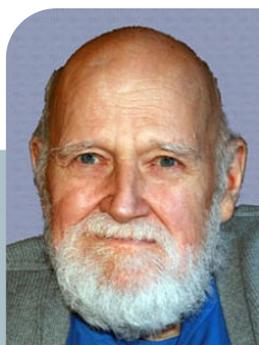
$$u(i) \in D(i) \quad \forall i \in N \quad \text{y} \quad v(j) \in D(j) \quad \forall j \in A$$

## 2.8 NOTAS HISTÓRICAS

El lema de Handshaking apareció por primera vez, aunque en forma diferente en un artículo de Leonhard Euler (1707-1783) titulado *Solutio Problematis ad Geometriam Situs Pertinentis* (la solución de un problema referido a la geometría de posición). Este artículo data de 1736 y es ampliamente conocido como el primer artículo en la teoría de gráficas, donde también da la solución de los puentes de Königsberg.



Leonhard Euler



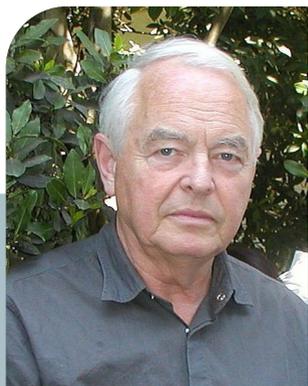
Lester Ford Jr.

Aunque las redes eléctricas se han estudiado por mucho tiempo, la utilidad de los flujos y potenciales de las redes en la modelación de problemas en economía e investigación de operaciones no fue reconocida hasta los años cincuenta.

El libro de L.R. Ford y D.R. Fulkerson (1962) ha jugado un papel realmente significativo al estimular el crecimiento y aplicaciones de la teoría

de redes en esas nuevas áreas. La notación y terminología en ese libro son naturales para la gente familiarizada con la programación lineal y han sido ampliamente aceptadas en la Investigación de Operaciones.

Los términos flujo y divergencia no fueron usados en teoría de redes antes de Rockafellar y pueden no estar completamente ligados a la ingeniería eléctrica debido a otro uso conectado con magnetismo. Sin embargo son simples, naturales y cubren una necesidad definida. El "nodo de distribución" en la red aumentada corresponde a la "tierra" en teoría eléctrica.



Ralph Tyrrell Rockafellar

Ralph Tyrrell Rockafellar (nacido el 10 de febrero de 1935) es un matemático estadounidense y uno de los principales eruditos en teoría de la optimización y campos relacionados de análisis y combinatoria.

Es autor de cuatro libros importantes, incluido el texto histórico *Convex Analysis* (1970), que ha sido citado más de 27000 veces según Google Scholar y sigue siendo la referencia estándar sobre el tema, y *Variational Analysis* (1998, con Roger JB Wets) por el que los autores recibieron el premio Frederick W. Lanchester del Instituto de Investigación de Operaciones y Ciencias de la Gestión (INFORMS). Para redes tiene este libro: Rockafellar, R. T. (1984). *Network Flows and Monotropic Optimization*. Wiley.



Delbert Ray Fulkerson

1

2

3

4

5

6

7

8



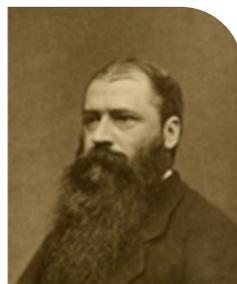
Gustav Kirchhoff

Los flujos  $x$  que pertenecen al espacio de circulaciones  $C$  se dice que también satisfacen la Ley de Corriente de Kirchhoff, mientras que las tensiones  $v$  en el espacio de diferenciales  $D$  satisfacen la Ley de Voltaje de Kirchhoff. El uso de esta terminología es en honor al trabajo pionero de G Kirchhoff en 1847.

Estrecho colaborador del químico Robert Bunsen, aplicó métodos de análisis espectrográfico (basados en el análisis de la radiación emitida por un cuerpo excitado energéticamente) para determinar la composición del Sol.

Una discusión de esas condiciones en la terminología de topología combinatoria se encuentra en Slepian (1968).

Slepian, Paul nació el 26 de marzo de 1923 en Boston, Massachusetts, Estados Unidos. Hijo de Philip e Ida (Goldstein) Slepian.



Paul Slepian

Una discusión de las condiciones de  $C$  circulaciones, y  $D$  tensiones se encuentra en la terminología de topología combinatoria de Slepian (1968).

El hecho de que  $C$  y  $D$  son ortogonales es también conocido como el Teorema de Tellegen.

## 2.9 EJERCICIOS PROPUESTOS

1. Dibuje una red cuya matriz de incidencia sea como sigue:

$$y = \begin{bmatrix} -1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & -1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \end{bmatrix}$$

2. Determine las matrices de incidencia y de adyacencia de la red que se muestra en la figura 1.

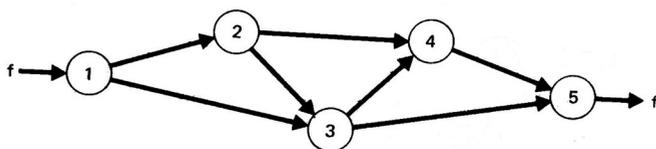


Figura 1

3. La figura 2 indica un cierto flujo  $x$ . Determine la divergencia  $y = \text{div } x$ , así como los nodos fuentes y los nodos sumideros con respecto a este flujo.

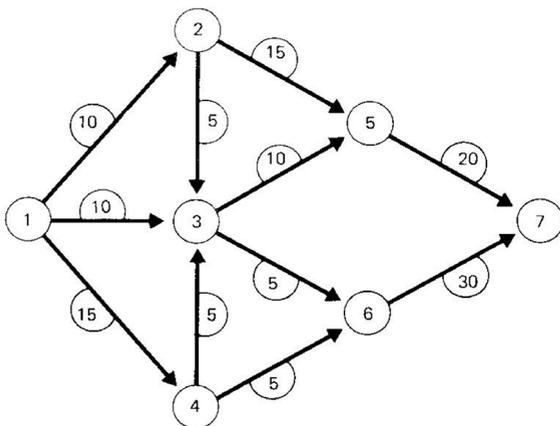


Figura 2

4. Sea  $x$  un flujo en una red  $G$  con la propiedad de que  $x$  se conserva en todos los nodos excepto en un par de nodos  $s$  y  $s'$ . Demuestre que  $y = \text{div } x$  satisface  $y(s) = y(s')$ , y de una interpretación física de esta relación.
5. Los valores que se muestran en la figura 3 definen una cierta  $y \in \mathbb{R}^N$ . ¿Existe un flujo  $x$  tal que  $\text{div } x = y$ ?

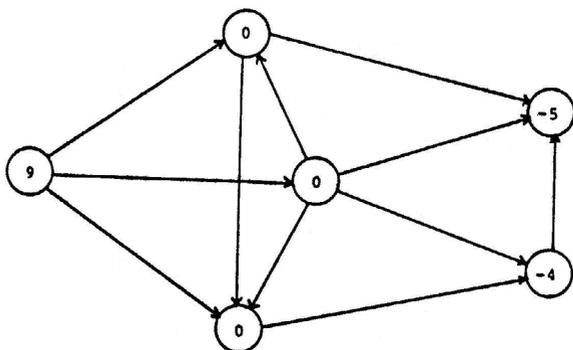


Figura 3

### 3.1 INTRODUCCIÓN

El objetivo de este capítulo es describir dos conceptos duales asociados con nodos y arcos: la trayectoria y el corte; así como el resultado central que los relaciona: el teorema de la red pintada y su equivalencia, el lema de Minty.

#### D Definición 3.1

En una red  $G$  se dice que  $P$  es una trayectoria, si consiste en una sucesión finita de la forma:

$$i_0, j_1, i_1, j_2, \dots, j_r, i_r.$$

Donde  $r > 0$  y cada  $i_k$  es un nodo mientras que  $j_k$  es un arco tal que

$$j_k \sim (i_{k-1}, i_k) \text{ o } j_k \sim (i_k, i_{k-1})$$

Los nodos  $i_0$  e  $i_r$  reciben el nombre de nodos inicial y final de la trayectoria, respectivamente. Si  $i_0 = i_r$  se dice que  $P$  es un circuito.

Un arco  $j_k$  en una trayectoria  $P$  se dice que se recorre positivamente (negativamente) si  $j_k \sim (i_k - 1, i_k)$  (si  $j_k \sim (i_k, i_{k-1})$ ). Si todos los arcos de  $P$  se recorren positivamente, se dice que  $P$  es una trayectoria o circuito positivo. De manera semejante  $P$  es una trayectoria o circuito negativo si todos sus arcos se recorren negativamente.

Si  $G$  es una digráfica, la trayectoria  $P$  se puede denotar como:

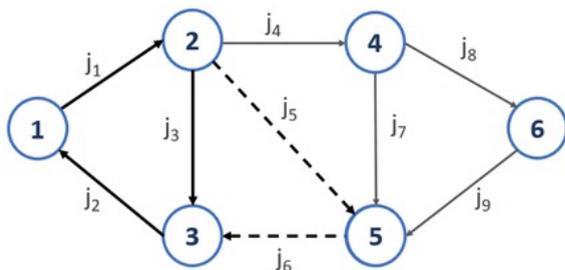
$$P: i_0 \rightarrow i_1 \rightarrow i_2 \rightarrow i_3$$

Pues a cada arco le corresponde un par de nodos en forma única.

### Ejemplo 3.1

En la figura 3.1 se muestran un circuito y una trayectoria como se describen a continuación, el circuito está indicado con línea gruesa y la trayectoria con línea punteada.

- a) Un circuito es:  $1, J_1, 2, J_3, 3, J_2, 1$  o bien  $P: 1 \rightarrow 2 \rightarrow 3 \rightarrow 1$
- b) Una trayectoria es:  $2, J_4, 4, J_8, 6, J_9, 5, J_5, 3, J_6, 2$  o bien  $P: 2 \rightarrow 4 \rightarrow 6 \rightarrow 5 \rightarrow 3 \rightarrow 2$



**Figura 3.1** Circuito y trayectoria

Una trayectoria  $P$  puede recorrer más de una vez un arco, si esto sucede se dice que  $P$  tiene multiplicidades o que es una trayectoria con multiplicidades.

Una trayectoria elemental o simple es aquella en donde cada arco y nodo se recorren una sola vez.

Sea  $P$  una trayectoria sin multiplicidades. Denotaremos por:  $P^+$  el conjunto de arcos que se recorren positivamente y por  $P^-$  el conjunto de arcos que se recorren negativamente.

Se define la función de incidencia de la trayectoria elemental  $P$  como:

$$e_p(j) = e(j,p) = \begin{cases} +1 & \text{si } j \in P^+ \\ -1 & \text{si } j \in P^- \\ 0 & \text{si } j \notin P \end{cases}$$

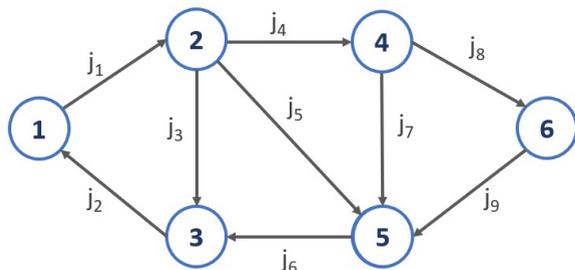
Note que esta función además de indicar la orientación en el arco puede interpretarse como un flujo “unitario” en la red para el ejemplo 3.1

- a)  $e_p = [1, 1, 1, 0, 0, 0, 0, 0, 0]$   
 b)  $e_p = [0, 0, 0, 0, 1, 1, 0, 0, 0]$

## E

## Ejemplo 3.2

Dada la siguiente red en la figura 3.2



**Figura 3.2** Red con trayectorias y circuitos positivos

- a) ¿Existe una trayectoria positiva del nodo 5 al 2?  
 b) ¿Existe un circuito positivo que contenga al arco  $J_6$ ?

## Conectividad

Una red  $G$  se dice **conexa** si para cada par de nodos distintos  $s$  y  $s^1$  existe una trayectoria  $P:s \rightarrow s^1$ , esto es, una trayectoria  $P$  que tiene como nodo inicial  $s$  y como nodo final  $s^1$ .

Si  $G$  no es conexa podemos particionarla en  $k$  redes  $G_i(N_i, A_i)$   $i=1, 2, \dots, k$  donde los subconjuntos de nodos  $N_i$  forman una partición de  $N$  y son tales que dos nodos  $s$  y  $s^1$  pertenecen al mismo conjunto  $N_i$  si y solo si existe una trayectoria  $P:s \rightarrow s^1$  o bien  $s=s^1$ .

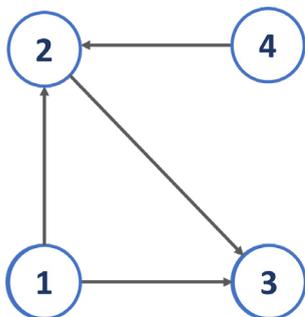
Los subconjuntos de arcos  $A_i$ , se definen como los arcos cuyos extremos pertenecen a  $N_i$  y también forman una partición de  $A$ .

Una red fuertemente conexa es aquella para la cual existe una trayectoria positiva  $p:s \rightarrow s^1$  para cada par de nodos distintos  $s$  y  $s^1$ .

### E

#### Ejemplo 3.3

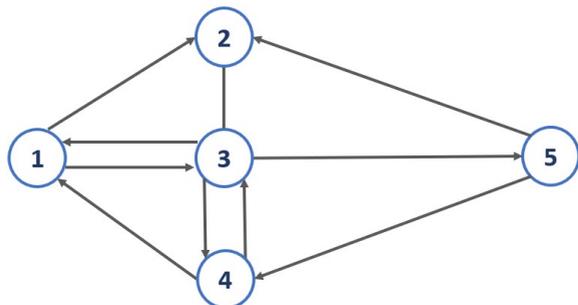
En la figura 3.3 se muestra una red conexa



**Figura 3.3**

Red conexa

Una red fuertemente conexa se muestra en la figura 3.4



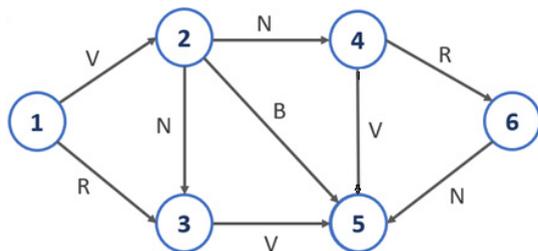
**Figura 3.4** Red fuertemente conexa

### 3.2 EL PROBLEMA DE DETERMINAR UNA TRAYECTORIA

Un aspecto básico de la definición de conexidad de una red es la existencia de trayectorias de un nodo a otro con el propósito de analizar este problema, que se repetirá continuamente, diremos que los arcos pueden recorrerse de acuerdo con las siguientes reglas:

- a) Arcos que se recorren en cualquier sentido (verdes)
- b) Arcos que únicamente se recorren en el sentido del arco (blancos)
- c) Arcos que únicamente se recorren en sentido inverso (negros)
- d) Arcos que no pueden recorrerse (rojos)

Para el ejemplo 3.2 dado el siguiente coloreado



**Figura 3.5** Red coloreada

Encuentre una trayectoria del nodo 5 al nodo 2 y otra del nodo 1 al nodo 6.

## Problema de la trayectoria pintada

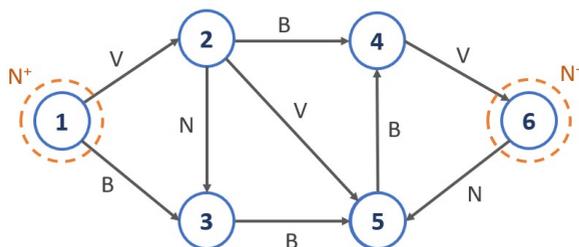
Sea  $G$  una red y  $N^+$  y  $N^-$  conjuntos ajenos de nodos. Suponga que hemos efectuado un pintado de la red (colores: verde, blanco, negro y rojo).

Determine una trayectoria  $P: N^+ \rightarrow N^-$ ; esto es, una trayectoria que parte de  $N^+$  y termina en  $N^-$  tal que cada arco de  $P^+$  es verde o blanco y cada arco de  $P^-$  es verde o negro.

Una trayectoria  $P$  con las propiedades anteriores se dice compatible con la coloración o bien el problema anterior equivale a encontrar una trayectoria compatible de  $N^+$  a  $N^-$ .

### Ejemplo 3.4

En la siguiente red encuentre una trayectoria compatible con la coloración de  $N^+$  a  $N^-$



**Figura 3.6**  
Trayectoria compatible con la coloración

Se propone la siguiente trayectoria  $P: 1 \rightarrow 2 \rightarrow 4 \rightarrow 6$ .

Es importante hacer notar que en el pintado cada arco de la red tiene exactamente uno de los colores que se han dado, aunque puede haber un color no usado. Por ejemplo, para probar conexidad todos los arcos se pintan de verde, mientras que para probar que una red es fuertemente conexas se pintarían de blanco o negro como se verá más adelante.

## Cortes

Un concepto dual a una trayectoria es el concepto de corte o cortadura que podemos formularlo como sigue:

Sean  $S$  y  $S'$  dos subconjuntos de nodos de una red  $G$  y se definen los conjuntos de arcos:

$$[S, S']^+ = \{j \in A \mid j \sim (i, i') \ i \in S, i' \in S'\}$$

$$[S, S']^- = \{j \in A \mid j \sim (i', i) \ i \in S, i' \in S'\}$$

El caso más usual es que  $S' = N \setminus S$ , es decir que  $S'$  es el complemento de  $S$  en  $N$ .

En este caso se dice que un conjunto de la forma  $[S, N \setminus S]$  es un corte  $Q = [S, N \setminus S]$  tal que:

$$Q^+ = [S, N \setminus S]^+ \quad y \quad Q^- = [S, N \setminus S]^-$$

O también

$$Q^+ = \{j \in A \mid j \sim (i, i'), i \in S, i' \in S'\}$$

$$Q^- = \{j \in A \mid j \sim (i', i), i \in S, i' \in S'\}$$

La palabra corte para  $Q$  proviene de la idea de que cualquier trayectoria  $P$  con nodo inicial en  $S$  y nodo terminal en  $N \setminus S$  debe, al menos en una etapa, atravesar uno de los arcos en  $Q$ , al borrar arcos en  $Q$  se cortaría tal trayectoria. El hecho de que  $P$  debe usar un arco de  $Q$  se establece formalmente de la siguiente manera:

Sea  $i$  el primero de los nodos en  $P$  que no está en  $S$ , tal nodo existe ya que  $P$  va de  $S$  a  $N \setminus S$ . El arco de  $P$  inmediatamente anterior  $i$ , llamado  $j$ , que une al nodo de  $S$  con un nodo de  $N \setminus S$ , y pertenece a  $Q$ , entonces  $j \in P^+ \cap Q^+$  o  $j \in P^- \cap Q^-$

Por ejemplo, en la red de la figura 3.6.

$$Q = \{S, N \setminus S\} \quad \text{Si} \quad S = \{1, 2, 3\}$$

$$Q^+ = \{(2,4), (2,5), (3,5)\} \quad Q^- = \emptyset$$

E

Ejemplo 3.5

Se considera la red de la figura 3.7 con la siguiente coloración:

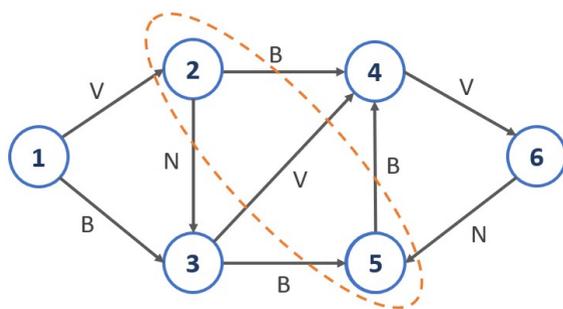


Figura 3.7  
Red con corte

Sea el corte  $Q = [S, N \setminus S]$  si  $S = \{2,5\}$

$$Q^+ = \{(2,4), (2,3), (5,4)\}$$

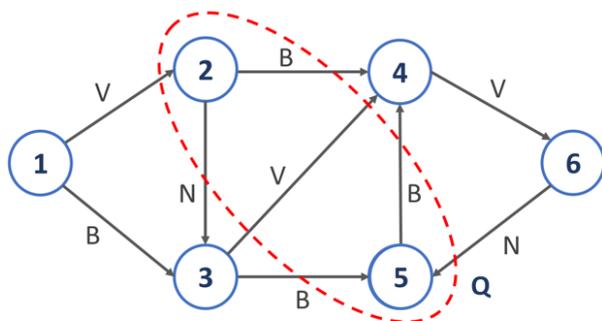
$$Q^- = \{(1,2), (3,5), (6,5)\}$$

E

Ejemplo 3.6

En la red del ejemplo 3.5 especifique el corte  $Q$  considerando que los elementos de  $Q^+$  sean blancos o verdes y los de  $Q^-$  sean negros o verdes.

Sea  $S = \{2, 5\}$  con  $Q = [S, N \setminus S]$



**Figura 3.8**  
Red con corte pintado

La función de incidencia del corte  $Q$  se define como:

$$e_Q = \begin{cases} 1 & \text{si } j \in Q^+ \\ -1 & \text{si } j \in Q^- \\ 0 & \text{en otro caso} \end{cases} \quad e_s E = e_Q$$

Es interesante observar que  $e_Q$ , como una función en el conjunto de los arcos, se puede considerar como un diferencial; sin embargo, esta es la tensión correspondiente al negativo del potencial

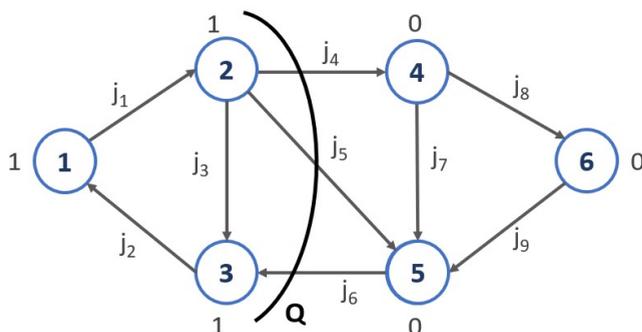
$$e_s(i) = \begin{cases} 1 & \text{si } i \in S \\ 0 & \text{si } i \notin S \end{cases}$$

Donde  $S$  es tal que  $Q = [S, N \setminus S]$ .

En general,  $e_Q = \Delta u$  con

$$u(i) = \begin{cases} 1 & \text{si } i \in N \setminus S \\ 0 & \text{si } i \in S \end{cases}$$

Sea la red de la figura 3.9, con el corte que se describe y la notación para el corte como se muestra a continuación.



**Figura 3.9**  
Red con corte

Sea  $Q = [S, N \setminus S]$  con  $S = \{1, 2, 3\}$

$$Q^+ = \{(2, 4), (2, 5)\}$$

$$Q^- = \{(5, 3)\}$$

$$e_q = [0, 0, 0, 1, 1, -1, 0, 0, 0]$$

Un corte se dice elemental si al quitar estos arcos de la red, el número de componentes de la red aumenta en uno.

### 3.3 ALGORITMO DE ENRUTAMIENTO

Se usa para determinar una solución al problema de las trayectorias compatibles con una coloración dada o bien un corte.

La idea del algoritmo es combinar un conjunto de nodos  $S \subset N^+$  y una función  $\Theta: S \setminus N^+ \rightarrow A$  que recibe el nombre de enrutamiento (o  $\Theta$ -en-

rutamiento) de  $S$  con base en  $N^+$  tal que etiqueta cada nodo  $i \in S \setminus N^+$ , con un arco  $j \in A$ , dicha etiqueta servirá para representar las trayectorias construidas.

- a) Para cada  $i \in S \setminus N^+$ ,  $\Theta(i)$  es un arco que une  $i$  con algún nodo de  $S$
- b) Se genera la sucesión  $i_0, \Theta(i_0), i_1, \Theta(i_1), \dots, i_k, \Theta(i_k)$ , donde  $i_k$  es el nodo final del arco  $\Theta(i_{k-1})$  y eventualmente se llega a  $N^-$

## Algoritmo de enrutamiento

**Propósito:** Determinar una trayectoria de  $N^+$  a  $N^-$  compatible con una coloración dada.

### Descripción

**Paso 1** Sean  $S = N^+$  y  $S' = N^-$  el enrutamiento vacío.

**Paso 2** Determine el corte  $Q = [S, N \setminus S]$

**2.1** Si existe  $j \in Q^+$  verde o blanco o si  $j \in Q^-$  es verde o negro ir a 3

**2.2** Si no existe  $j$  terminar. No hay solución al problema

**Paso 3** Sea  $\Theta(i) = j$  con  $i \notin S$  hacer  $S := S \cup \{i\}$  (enrutamiento compatible con la coloración)

**3.1** Si  $i \in N^-$  terminar.  $\Theta$  contiene una trayectoria compatible  $P: N^+ \rightarrow N^-$

**3.2** Si  $i \notin N^-$  se tiene  $S \cap S' = \emptyset$  ir a 2

### Ejemplo 3.8

Dada la red con  $N^+ = \{s\}$  y  $N^- = \{s'\}$  encontrar una trayectoria compatible con el pintado

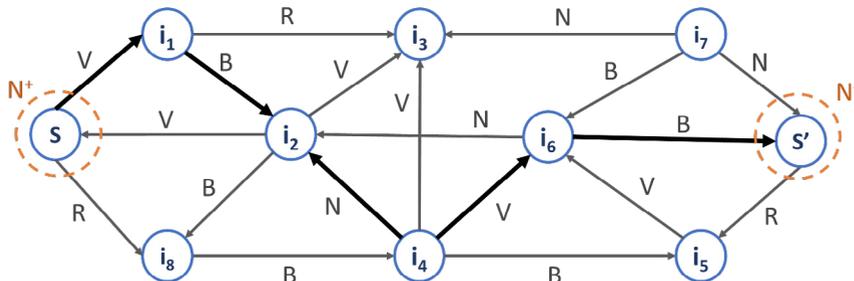


Figura 3.10 Una red con trayectoria compatible

El nodo  $i_k$  se aumenta a  $S$  en la  $k$ -ésima iteración

$$P: s \rightarrow i_1 \rightarrow i_2 \leftarrow i_4 \leftarrow i_6 \leftarrow s'$$

### Ejemplo 3.9

Determine si existe una trayectoria  $P: N^+ \rightarrow N^-$  en la siguiente red con  $N^+ = \{1\}$ ,  $N^- = \{6\}$

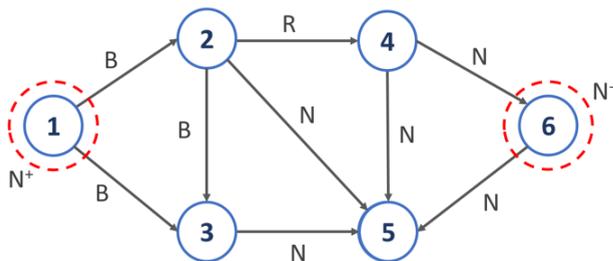


Figura 3.11 Red con pintado

Definimos un corte que solo contenga al nodo 1, nos fijamos en sus arcos, si el arco que va al nodo 2 es blanco extendemos  $s$  a ese nodo,

lo mismo para el nodo 3, el conjunto de estados alcanzables es 2, 3. Se define otro corte que abarca a los nodos 1, 2, 3 y se vuelve a empezar.

## E

## Ejemplo 3.10

Dada la siguiente red en la figura 3.12, encuentre una trayectoria compatible con la coloración o un corte.

**Paso 1** Comenzamos con  $S_0 = N^+ = \{i_1\}$  y  $\Theta$  vacío.

**Paso 2**  $Q_1 = \{j_1, j_2, j_3\}$ ;  $j_1, j_2, j_3$  son compatibles con el pintado.

**Paso 3** Sea

$$\Theta(i_2) = j_1$$

$$\Theta(i_3) = j_2$$

$$\Theta(i_4) = j_3$$

$$S_1 = S_0 \cup \{i_2, i_3, i_4\} = \{i_1, i_2, i_3, i_4\}$$

$$S_1 \cap N^- = \emptyset, \text{ ir al paso 2.}$$

**Paso 2**

Iteración 1

$Q_2 = \{j_4, j_5, j_7, j_9, j_{10}, j_{11}\}$ ;  $j_4, j_5$  son compatibles con el pintado.

**Paso 3** Sea  $\Theta(i_7) = j_4$

$$\Theta(i_5) = j_5$$

$$S_2 = S_1 \cup \{i_5, i_7\} = \{i_1, i_2, i_3, i_4, i_5, i_7\}.$$

$$S_2 \cap N^- = \emptyset, \text{ ir al paso 2.}$$

**Paso 2**

Iteración 2

$$Q_3 = \{j_9, j_{10}, j_{11}, j_{13}, j_{14}, j_{17}, j_{18}\}.$$

1

2

3

4

5

6

7

8

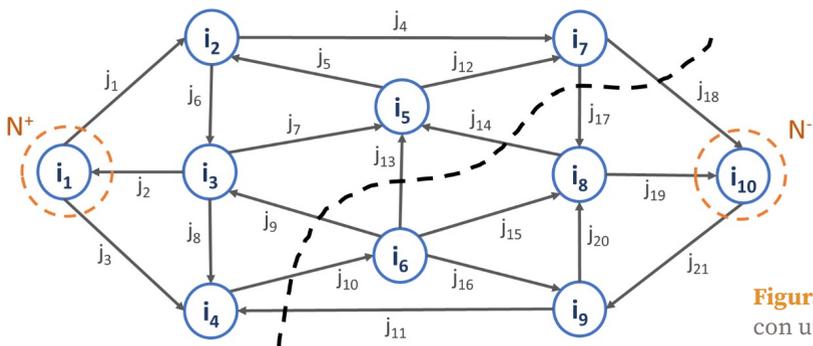


Figura 3.12 Red con un corte compatible

No existen arcos compatibles con la coloración. Por lo cual no existe solución al problema.

**NOTA:** El pintado es:

<u>Color</u>	<u>Arco</u>
Verde	$j_1, j_3, j_4, j_{12}, j_{15}, j_{21}$
Blanco	$j_9, j_{11}, j_{13}, j_{14}$
Negro	$j_2, j_5, j_6, j_7, j_8, j_{16}, j_{18}, j_{20}$
Rojo	$j_{10}, j_{17}, j_{19}$

### Dualidad de trayectorias y cortes

El problema complementario al de encontrar una trayectoria compatible con una coloración dada es el corte compatible. Se dice que el corte  $Q$  separa  $N^+$  de  $N^-$  si es de la forma  $[S, N \setminus S]$ , para algún  $S \subset N$ , tal que  $N^+ \subset S$  y  $N^- \cap S = \emptyset$ . Se denotará  $Q$  con  $N^+ \downarrow N^-$ .

**Problema del corte coloreado (o pintado).** Sean  $N^+$  y  $N^- \subset N$  tales que  $N^+ \cap N^- = \emptyset$ . Sea una coloración en la red  $G$  con los colores verde, blanco, negro y rojo.

El problema es determinar un corte  $Q: N^+ \downarrow N^-$  tal que todo arco de  $Q^+$  sea rojo o negro mientras que todo arco de  $Q^-$  sea rojo o blanco.

Un corte que cumple las restricciones de color se dice compatible con la coloración y si además separa  $N^+$  de  $N^-$  constituye la solución al problema del corte coloreado. Observe la dualidad entre las restricciones de color para trayectorias y cortes.

**Teorema (de la red coloreada).** Sean  $N^+ \subset N$  y  $N^- \subset N$ , tales que  $N^+ \cap N^- = \emptyset$ . Entonces, para toda coloración de la red  $G$  con los colores verde, blanco, negro y rojo, una y solo una de las siguientes afirmaciones es válida.

- a) El problema de la trayectoria coloreada tiene solución  $P$ .
- b) El problema del corte coloreado tiene solución  $Q$ .

## D Demostración

Sea  $G$  una red coloreada y  $N^+ \subset N$  y  $N^+ \cap N^- = \emptyset$ . Aplique el algoritmo de enrutamiento, entonces el algoritmo tiene 2 terminaciones posibles y excluyentes, si termina en el paso 2 se ha construido un corte  $Q$  compatible con la coloración, la existencia de este corte garantiza la no existencia de una trayectoria compatible; o bien, en el paso 3, donde se obtiene una trayectoria coloreada  $P$  la que a su vez excluye la posibilidad de encontrar un corte  $Q$  compatible con la coloración

Existe otro resultado fuertemente relacionado con el teorema de la red coloreada. Este resultado es el lema de Minty que utiliza el concepto de corte elemental paralelo al de trayectoria elemental.

## Lema de Minty

Considere una red  $G$  con pintado de arcos verde, blanco, negro o rojo. Dado cualquier arco  $j'$  blanco o negro uno y solo uno de los siguientes postulados es cierto:

- c) Existe un corte  $Q$  (elemental) compatible con la coloración que usa  $j'$ .
- d) Existe un circuito (elemental) compatible con la coloración que usa  $j'$ .

### D Demostración

Sea  $\mathbf{j} = (i_1, i_2)$  un arco blanco o negro, si  $\mathbf{j}$  es blanco considere  $N^+ = \{i_2\}$  y  $N^- = \{i_1\}$ . Si  $\mathbf{j}$  es negro considere  $N^+ = \{i_1\}$  y  $N^- = \{i_2\}$ .

Aplicamos el algoritmo del enrutamiento para encontrar una trayectoria de  $N^+$  a  $N^-$

Observe que el arco  $\mathbf{j}$  siempre pertenece al corte  $Q$  del paso 2 del algoritmo, pero debido a su coloración y la forma de selección de  $N^+$  y  $N^-$  entonces  $\mathbf{j}$  no es compatible con la coloración.

Ahora el algoritmo solo tiene dos terminaciones excluyentes en el paso 2: que es el determinar un corte que contendría a  $\mathbf{j}$  o bien en el paso 3 al encontrar una trayectoria de  $N^+$  a  $N^-$  que no contiene a  $\mathbf{j}$  y como  $\mathbf{j}$  es compatible con una trayectoria de  $N^-$  a  $N^+$  por su forma de selección se forma un circuito.

### P Proposición

El teorema de la red coloreada y el lema de Minty son equivalentes.

D

**Demostración**

Use Minty para probar el teorema.

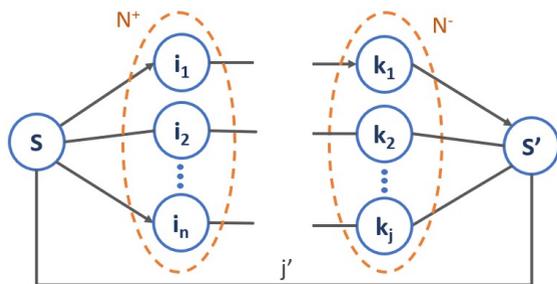
Ya vimos que se demuestra el lema de Minty usando el teorema de la red coloreada, para demostrar que son equivalentes basta demostrar el teorema de la red coloreada usando el lema de Minty.

Entonces procedemos a demostrar el teorema de la red coloreada.

D

**Demostración**

Sea  $G$  una red coloreada con los cuatro colores de costumbre y sean  $N^+$ ,  $N^-$  dos subconjuntos ajenos del conjunto  $N$  de nodos de la red. Sea  $G'$  una red construida agregando a  $G$  el arco  $j' \sim (s', s)$  y los arcos  $(s, i)$  para todo  $i \in N^+$  y  $(k, s')$  para toda  $k \in N^-$  como se muestra en la figura siguiente:



**Figura 3.13**  
Demostración del lema de Minty

Se colorean todos los nuevos arcos de blanco. La aplicación del lema de Minty para el arco  $j'$  lleva dos posibilidades:

- i) Existe un circuito elemental  $P$  compatible con la coloración que contiene al arco  $j'$ . Puesto que  $j'$  es blanco, este circuito debe ser de la forma  $P: s', j', s, j_1, i, j_2, k, j_3, s'$  en donde  $j_1 \sim (s, i)$  con  $i \in N^+$ ,  $j_2 \sim (i, k)$  y  $j_3 \sim (k, s')$  con  $k \in N^-$  y  $P'$  es una trayectoria de  $i$  a  $k$  compa-

tible con la coloración. Con esto se establece una correspondencia biunívoca de los circuitos  $P'$  con estas características y las trayectorias  $P': i \rightarrow k$  con  $i \in N^+$ ,  $k \in N^-$  de donde se deriva la primera condición del teorema de la trayectoria coloreada.

- ii) Existe un corte elemental  $Q$  compatible con la coloración que contiene el arco  $j'$ . En este caso  $Q$  no puede contener ningún arco nuevo distinto de  $j'$  puesto que, siendo estos blancos, no satisfarían las restricciones de color (nótese que  $j \in Q^-$ ).

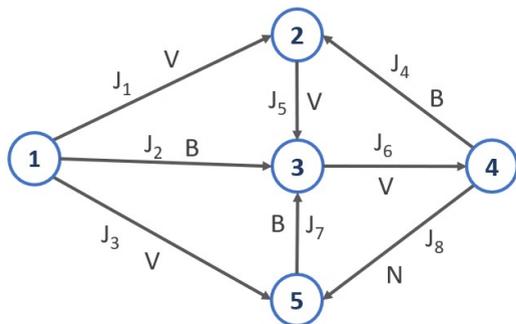
De aquí que los demás arcos de  $Q$  son arcos de la red original  $G$  y por tanto constituyen un corte compatible con la coloración en ella. De nuevo, esta correspondencia de cortes es biunívoca por lo que se deriva la segunda alternativa del teorema de la red coloreada.

Conviene señalar que el lema de Minty en términos algebraicos es equivalente al lema de Farkas; que no es otra cosa que un resultado de separación de dos conjuntos convexos (poliedros).

E

Ejemplo 3.11

Determine si el arco  $j \sim (4, 5)$  pertenece a un circuito elemental compatible o a un corte elemental compatible con la coloración definida en la red de la figura siguiente:



**Figura 3.14**  
Red donde se aplica lema de Minty

S

Solución

Para verificar si este arco pertenece a un circuito o corte elementales compatibles con la coloración se utilizará el algoritmo de Minty. Es decir, se resuelve el problema de la trayectoria coloreada de  $N^+ = \{i_4\}$  a  $N^- = \{i_5\}$

Utilizando el algoritmo de enrutamiento se obtiene la trayectoria compatible de  $i_4$  a  $i_5$

$$P: 4 \rightarrow 2 \leftarrow 1 \rightarrow 5$$

Esta trayectoria  $P$  junto con el arco  $j_8$  forman un circuito elemental.

E

Ejemplo 3.12

Determine si el arco  $j_8$  pertenece a un circuito elemental compatible o a un corte elemental compatible con la coloración dada en la red de la figura siguiente:

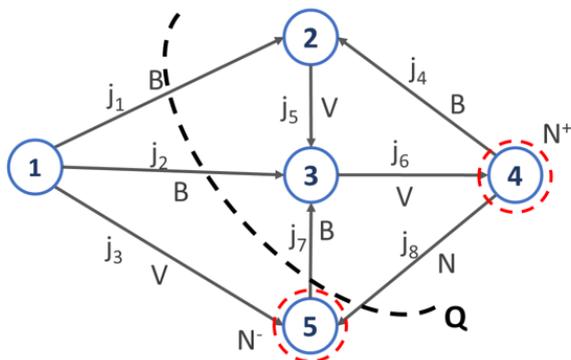


Figura 3.15 Red donde se aplica lema de Minty

S

Solución

Aplicando el algoritmo de enrutamiento se determina  $Q = [S, N \setminus S]$  donde  $S = \{i_4, i_2, i_3\}$  que contiene a  $j_8$  y es compatible con la coloración.

### 3.4 APLICACIONES A CONEXIDAD

El algoritmo de la red pintada proporciona un medio eficiente para probar cuándo una red es conexa y si no lo es nos sirve para determinar sus componentes.

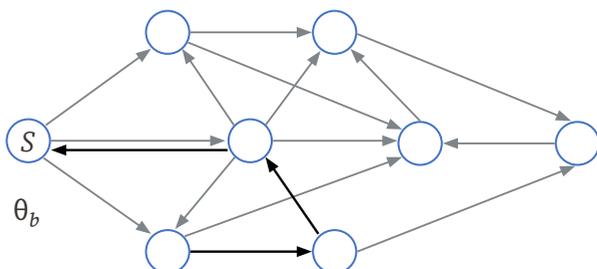
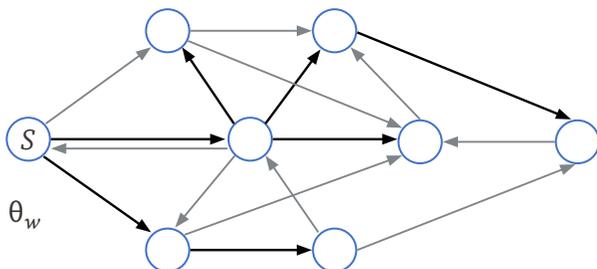
Seleccione un nodo arbitrario  $s$  y aplique el algoritmo con  $N^+ = \{s\}$  y  $N^- = \emptyset$  y todos los arcos pintados de verde. En la terminación habrá un enrutamiento máximo con base  $s$  (con notación  $\{s\}$ ) correspondiente a cierto conjunto  $S$  que contiene a  $s$ . De la naturaleza del pintado, es claro que  $S$  consiste en todos los nodos alcanzables por trayectorias que empiezan en  $s$  y describe un sistema particular de trayectorias que completan la tarea. Si  $S = N$ ,  $G$  está conectada. Si no, entonces los nodos en  $S$ , junto con los arcos incidentes a ellos forman la componente de  $G$  conteniendo a  $s$ .

Para determinar otra componente seleccione cualquier nodo que no está en  $S$  y repita el procedimiento. Después de una sucesión finita de tales cálculos todas las componentes se identificarán.

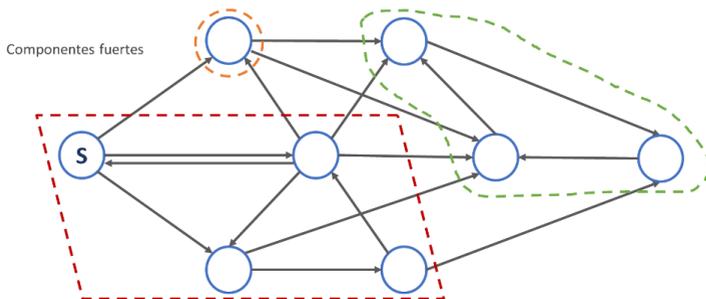
La prueba para una red fuertemente conexa necesita el doble del esfuerzo. Aplicando el algoritmo como se hizo antes, y comenzando desde un nodo arbitrariamente seleccionado  $s$ , pero con los arcos pintados de blanco. Esto nos da un enrutamiento máximo compatible con la coloración  $\Theta_w$  asociado con un conjunto de nodos  $S_w$ . Repetimos la aplicación con los arcos todos blancos, obteniendo  $\Theta_b$  y  $S_b$ .

Obviamente, los nodos en  $S_w$  (fuera de  $s$ ) son aquellos que pueden alcanzarse por una trayectoria positiva desde  $s$ , mientras aquellos en  $S_b$  son los que pueden llegar a  $s$  a través de una trayectoria positiva. Entonces el conjunto  $S = S_w \cap S_b$  está compuesto por todos los nodos fuertemente

conectados a  $s$ . Si  $S=N$ ,  $G$  es fuertemente conectada. Si no,  $S$  suministra una componente fuerte que contiene a  $s$  (como podemos ver en la figura siguiente).



**Figura 3.16**  
Prueba de  
conexidad en  
una red



En el caso de una red en una ciudad se pueden pintar las calles de doble sentido de verde, todas las de un sentido de blanco, y todas las que estén cerradas por reparaciones de rojo. ¿Será posible ir de un nodo a cualquier otro nodo?

### 3.5 NOTAS HISTÓRICAS

La teoría de gráficas y redes ha sufrido de una falta de estandarización en cuanto a su terminología, diferentes autores usan diferentes palabras para el mismo concepto, o una misma palabra en diferente sentido. En parte, esto se debe a que hay muchas áreas de aplicación.

En el caso de las “trayectorias” como las que se exponen aquí es difícil encontrar dos textos que estén de acuerdo, lo mismo sucede con cortes. La terminología usada aquí es acorde con el teorema de la red coloreada y el lema de Minty (1960), aunque Minty solo usó tres colores, él no admitió la categoría de usar negro, solo se refiere al sentido inverso de los arcos blancos. Sin embargo, esto no provoca ningún problema al usar arcos negros. Enrutamiento se usa por otros autores como arcos con raíz o arborescencias, que veremos más adelante.

196

G. J. Minty

disjoint sets of branches connecting  $W$  to the other components; these sets are easily seen to be co-cycles in which the notion of ‘similarly directed’ is the same as in  $U(W)$ . If, on the other hand, deletion of  $U(W)$  splits  $W$  into several components  $W^1, \dots, W^m$ , then  $U(W)$  is the union of the disjoint sets  $U(W^1), \dots, U(W^m)$ , in which the notion of ‘similarly directed’ is the same as in  $U(W)$ . By the preceding remark, each of these co-boundaries can be decomposed into co-cycles, and in general we have:

**LEMMA 3-1.** *For any set  $W$  of nodes of a network: If  $U(W)$  is non-empty, it can be partitioned into disjoint co-cycles, in each of which the notion of ‘similarly directed’ is the same as in  $U(W)$ . A co-cycle can be characterized as a minimal non-empty co-boundary.*

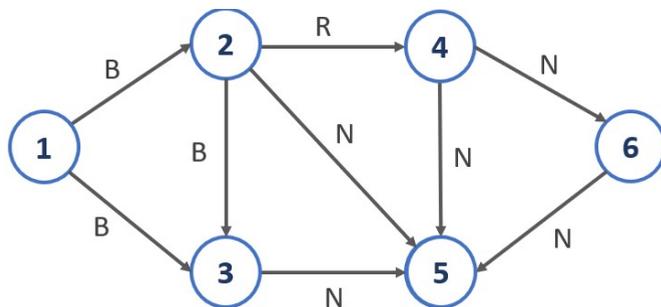
A *skeleton* is a maximal set of branches containing no cycle, and a *co-skeleton* is a maximal set of branches containing no co-cycle. Clearly a skeleton contains at least one branch of each co-cycle, and a co-skeleton contains at least one branch of each cycle, so that their complements contain no co-cycle and no cycle, respectively.

**THEOREM 3-1.** *Let  $N$  be a network whose branches are partitioned into three sets (or coloured with three colours)  $A$ ,  $B$ , and  $C$ , and let one branch of the set  $B$  be distinguished and called  $b$ . Then there exists one (but not both) of the following: (i) a cycle, containing  $b$  but no element of  $C$ , in which all elements of  $B$  are similarly directed, or (ii) a co-cycle, containing  $b$  but no element of  $A$ , in which all elements of  $B$  are similarly directed.*

### 3.6 EJERCICIOS PROPUESTOS

1. (Opción múltiple). En las siguientes preguntas seleccione una opción y justifique su respuesta. Sea  $G(N, A)$  una red,  $N^+$  y  $N^-$  dos subconjuntos de nodos no vacíos y disjuntos. Dada una coloración con los colores verde, blanco, negro y rojo con  $j' \sim (i, i')$  un arco blanco con  $i, i'$  fuera de  $N^+$  y  $N^-$  entonces
  - a) Siempre existe una trayectoria de  $N^+$  a  $N^-$  compatible con la coloración que usa  $j'$
  - b) Siempre existe un corte  $Q$  compatible con la coloración que usa  $j'$
  - c)  $j'$  siempre forma parte de un circuito compatible con la coloración
  - d) Ninguna de los anteriores es correcto
  
2. (Opción múltiple). Sea  $G = (N, A)$  una digráfica, el rango de la matriz de incidencia o número de renglones o columnas linealmente independientes es
  - a)  $= |A|$
  - b) Siempre igual a  $|N| - 1$
  - c)  $< |N|$
  - d)  $= |N|$
  - e) Ninguna de las anteriores
  
3. (Opción múltiple). Considere la condición  $e_p v = 0$  donde  $P$  es un circuito y  $v$  es un vector
  - a) Si la condición se cumple, entonces  $v = 0$
  - b) La condición se cumple para todo  $P$  si y solo si  $P$  es una circulación

- c) Si la condición se cumple,  $v$  es un diferencial si y solo si  $P$  es una circulación
  - d) La condición implica que  $v$  es un diferencial
  - e) Ninguna de las anteriores.
4. (Algoritmo de enrutamiento) Aplique el algoritmo de enrutamiento a la red siguiente y encuentre una trayectoria compatible con la coloración o un corte.



# El problema de flujo máximo-corte mínimo

## 4.1 INTRODUCCIÓN

En muchos problemas de redes se involucran flujos que tienen ciertas restricciones en al menos uno de los arcos, también se pueden tener restricciones en las divergencias permitidas en los nodos. El propósito de este capítulo es introducir los conceptos necesarios para resolver estos problemas, analizar el problema clásico de flujo máximo y su problema dual, el corte mínimo, así como sus correspondientes métodos de solución. Estos métodos de solución son el coloreado de la red y el de etiquetas. De este modelo se desprende el problema de distribución factible, en donde se supone de antemano una oferta y una demanda, las cuales se tienen que cumplir para lo que se usan los algoritmos de distribución factible y de rectificación de flujo.

## 4.2 PRINCIPIO FUNDAMENTAL DE DIVERGENCIA

El problema de flujo máximo fue resuelto en 1986 por R. L. Ford y R.D. Fulkerson, que de hecho fueron los iniciadores de la era de flujo en redes independientes de redes eléctricas.

## Intervalos de capacidad

El flujo en un arco  $j$  de una red varía en un intervalo cerrado  $c(j)$ , denominado intervalo de capacidad de  $j$ . Dicho intervalo se denota por:

$$c(j) = [c^-(j), c^+(j)]$$

Donde  $c^-(j)$  es la mínima capacidad y  $c^+(j)$  es la máxima.

La única restricción es que  $c(j)$  sea un intervalo no-vacío. En particular  $c^+(j)$  puede ser  $+\infty$  y  $c^-(j)$  puede ser  $-\infty$ .

Un flujo se dice factible si  $x(j) \in c(j) \forall j \in A$ .

Algunos ejemplos de intervalos de capacidad son:

1.  $c(j) = [-c, c]$  con  $0 \leq c < +\infty$ ; el flujo  $x$  puede usar el arco  $j$  en ambas direcciones, pero el flujo debe satisfacer  $|x(j)| \leq c$ .
2.  $c(j) = [0, c]$  con  $0 \leq c < +\infty$  como en el ejemplo anterior, pero el arco solo puede usarse en dirección positiva.
3.  $c(j) = [0, \infty)$  el arco solo puede usarse en dirección positiva, pero no hay una cota superior en el flujo.
4.  $c(j) = (-\infty, +\infty)$  aquí no hay ninguna restricción de flujo en el arco  $j$ .
5.  $c(j) = [c, c]$  con  $-\infty < c < +\infty$ ; aquí hay un requerimiento exacto,  $x(j) = c$

Una consecuencia directa de que los flujos en los arcos estén acotados es que se restringe el flujo que puede pasar a través de un corte. Específicamente, el flujo  $x$  a través de un corte  $Q$  se define como:

$$e_Q x = \sum_{j \in Q^+} x(j) - \sum_{j \in Q^-} x(j) \quad (4.1)$$

Donde  $e_Q$  es la función de incidencia para  $Q$ . Este se puede interpretar como la cantidad neta de material que fluye a través de  $Q$  en la dirección de la orientación de  $Q$ . Para reforzar esta interpretación escriba  $Q = [S, N \setminus S]$  para el conjunto de nodos  $S$ , se define la divergencia de  $x$  desde  $S$  como:

$$y(S) = \sum_{i \in S} y(i) \quad (4.2)$$

Esta cantidad representa la cantidad neta de material originada en  $S$  (esto es, la cantidad total de la fuente menos la cantidad total del sumidero). Estas cantidades (4.1) y (4.2) se relacionan a través del principio de divergencia.

### Principio fundamental de divergencia

Sea  $x$  un flujo en la red  $G$  y sea el corte  $Q = [S, N \setminus S]$ . Entonces  $y(S) = e_Q x$

$$[\text{Divergencia de } x \text{ desde } S] = [\text{flujo de } x \text{ a través del corte } Q]$$

### Demostración

Usando las definiciones de  $y = \text{div } x$  y de  $e_Q$  se tiene:

$$\begin{aligned} y(S) &= \sum_{i \in S} y(i) = \sum_{i \in S} \sum_{j \in A} e(i, j) x(j) = \sum_{j \in A} \sum_{i \in S} e(i, j) x(j) = \\ &= \sum_{j \in Q^+} \sum_{i \in S} e(i, j) x(j) + \sum_{j \in Q^-} \sum_{i \in S} e(i, j) x(j) = \\ &= \sum_{j \in Q^+} x(j) - \sum_{j \in Q^-} x(j) = e_Q x \end{aligned}$$

Nota: La regla de divergencia total donde  $y(N) = 0$  es un caso particular del principio de divergencia ya que si  $S = N$  entonces  $Q = \phi$  y de aquí se concluye el resultado.

En un corte  $Q$  se observa:

$$c^-(j) \leq x(j) \leq c^+(j) \quad j \in Q^+$$

$$-c^+(j) \leq -x(j) \leq -c^-(j) \quad j \in Q^-$$

y sumando se tiene:

$$\sum_{j \in Q^+} c^-(j) \leq \sum_{j \in Q^+} x(j) \leq \sum_{j \in Q^+} c^+(j)$$

$$-\sum_{j \in Q^-} c^+(j) \leq -\sum_{j \in Q^-} x(j) \leq -\sum_{j \in Q^-} c^-(j)$$

$$\sum_{j \in Q^+} c^-(j) - \sum_{j \in Q^-} c^+(j) \leq \sum_{j \in Q^+} x(j) - \sum_{j \in Q^-} x(j) \leq \sum_{j \in Q^+} c^+(j) - \sum_{j \in Q^-} c^-(j)$$

$$C^-(Q) \leq \text{flujo a través del corte } Q \leq C^+(Q)$$

Estas son las capacidades inferior y superior del corte, el intervalo de capacidad asociado con  $Q$  es  $c(Q) = [c^-(Q), c^+(Q)]$

### Ejemplo 4.1

Dada la red de la figura 4.1 se observa el corte y se calcula la divergencia en los nodos de  $S$  así como el flujo a través del corte.

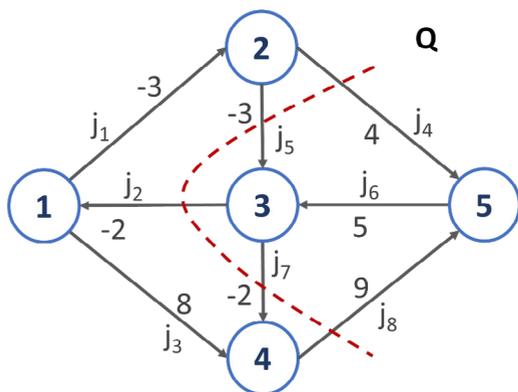


Figura 4.1 Red con corte Q

Con  $y(1) = 7, y(2) = 4, y(3) = -6, y(4) = 3, y(5) = -8$

Sea  $S = \{1, 2, 4\}$ , entonces  $Q = \{j_4, j_5, j_2, j_8, j_7\}$

Se puede observar que la divergencia de  $x$  desde  $S$ :  $y(S) = 7 + 4 + 3 = 14$   
 Y el flujo  $x$  a través de  $Q$

$$Q^+ = \{J_4, J_5, J_8\} = 4 - 3 + 9 = 10$$

$$Q^- = \{J_2, J_7\} = -2 - 2 = -4$$

$$e_Q x = Q^+ - Q^- = 10 - (-4) = 14, \text{ son iguales, por (4.1).}$$

E

Ejemplo 4.2

Dada la red de la figura 4.2 con divergencias en los nodos marcadas con [.] y el flujo en los arcos con el corte que se describe, verifique el principio fundamental de divergencia.

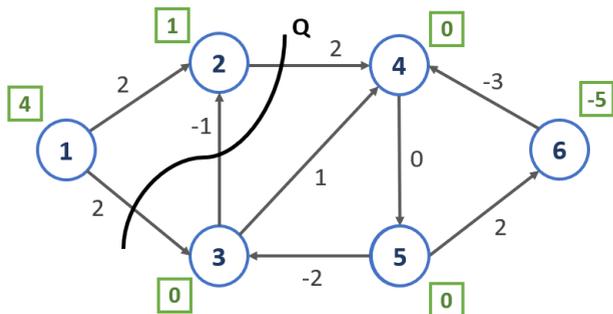


Figura 4.2 Red con corte, flujos y divergencias

$$S = \{1, 2\}$$

$$Q = \{(1, 3), (2, 4), (3, 2)\}$$

Nodo $i$	1	2	3	4	5	6
$y(i)$	4	1	0	0	0	-5

$$Q^+ = \{(1,3), (2,4)\}$$

$$Q^- = \{(3,2)\}$$

$$Q^+ = \text{flujo} = \{2 + 2\}$$

$$Q^- = \text{flujo} = \{-1\}$$

$$Q^+ - Q^- = \{(2 + 2) - (-1)\} = 5$$

$$y(S) = 5 = e_Q x$$

E

Ejercicio

Verifique el principio fundamental de divergencia en la siguiente red:

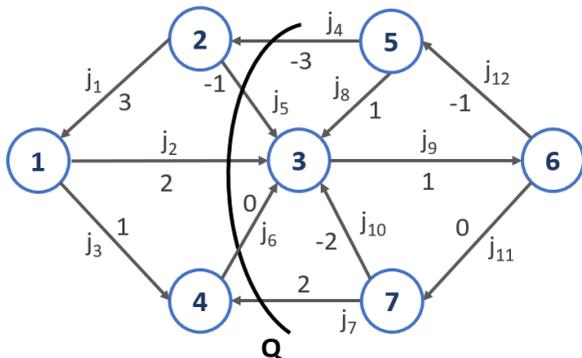


Figura 4.3 Red con corte

Nodo $i$	1	2	3	4	5	6	7
$y(i)$	0	5	1	-3	-1	-2	0

$$S = \{1, 2, 4\} \quad Q = \{J_4, J_5, J_2, J_6, J_7\} \quad y(S) = 2 = e_Q \cdot x$$

$$Q^+ = \{J_2, J_5, J_6\}$$

$$Q^- = \{J_4, J_7\}$$

### 4.3 Problema de flujo máximo en una red

Sea  $G$  una red con intervalos de capacidad, y sean  $N^+$  y  $N^-$  conjuntos de nodos disjuntos de  $G$ . Sea también cualquier flujo  $x$  que se conserva en todos los nodos que no pertenecen a  $N^+$  ó  $N^-$ , en otras palabras, tal que tienen  $y(i) = 0 \quad \forall i \in (N^+ \cup N^-)$  y donde  $y = \text{div } x$ . Por la regla de divergencia total se tiene que:

$$0 = \sum_{i \in N} y(i) = \sum_{i \in N^+} y(i) + \sum_{i \in N^-} y(i)$$

lo que implica

$$y(N^+) = -y(N^-)$$

Esta última cantidad recibe el nombre de flujo de  $x$  de  $N^+$  a  $N^-$

## Problema de flujo máximo

El problema de flujo máximo queda entonces definido de la siguiente manera:

El problema consiste en maximizar el flujo de  $N^+$  a  $N^-$  sobre todos los flujos  $x$  tales que  $y(i) = 0$  para todas las  $i \notin N^+ \cup N^-$  factibles con respecto a las capacidades.

Se supondrá la existencia de al menos un flujo que satisfaga todas las restricciones de capacidad y conservación.

La mínima cota superior del conjunto de flujos factibles de  $N^+$  a  $N^-$  se denomina el supremo en el problema de flujo máximo. El supremo puede ser infinito, un flujo cuyo valor es igual al supremo se dice solución al problema de flujo máximo.

Una formulación del problema en términos de programación lineal equivale a

$$\max z = \sum_{i \in N^+} \sum_{j \in A} e(i, j) x(j)$$

sujeto a

$$\sum_{j \in A} e(i, j) x(j) = 0 \quad i \notin N^+ \cup N^- \quad (\text{restricciones de conservación})$$

$$c^-(j) \leq x(j) \leq c^+(j) \quad j \in A \quad (\text{restricciones de capacidad})$$

Para resolver este problema se introduce el concepto dual de corte mínimo que será utilizado como herramienta de optimización. Como

se había mencionado anteriormente, todas las trayectorias de  $N^+$  a  $N^-$  utilizan algún arco de cualquier corte  $Q: N^+ \downarrow N^-$ , de este modo los cortes constituyen “cuellos de botella” para el valor del flujo de  $N^+$  a  $N^-$ . Es decir:

### P Proposición 4.1

Sea  $x$  un flujo que satisfice las restricciones del problema y sea  $Q: N^+ \downarrow N^-$  un corte que separa  $N^+$  de  $N^-$  entonces

$$\text{flujo de } x \text{ de } N^+ \text{ a } N^- \leq C^+(Q)$$

### D Demostración

Puesto que  $Q$  es un corte que separa  $N^+$  de  $N^-$  entonces es de la forma  $[S, N \setminus S]$  con  $N^+ \subset S$  y  $S \cap N^- = \emptyset$ , Además como  $x$  se conserva en todos los nodos excepto los de  $N^+ \cup N^-$ , se tiene que  $y(i) = 0$  para  $i \in S \setminus N^+$ , por tanto  $y(N^+) = y(S)$ , pues  $y(i) = 0$  para toda  $i \in S \setminus N^+$ . Asimismo, por el principio de divergencia  $y(S) = e_Q x \leq C^+(Q)$  lo que demuestran la proposición.

### Problema de corte mínimo

Minimizar  $C^+(Q)$  sobre todos los cortes  $Q: N^+ \downarrow N^-$

Una consecuencia inmediata, de la proposición anterior está dada por:

Supremo en problemas de flujo máximo	$\leq$	Mínimo en problemas de corte mínimo
--------------------------------------	--------	-------------------------------------

La relación más importante entre los dos problemas es el hecho de que la igualdad se cumple en los óptimos. Para probar este resultado es necesario definir los conceptos de trayectoria aumentante de flujo y trayectoria de capacidad ilimitada.

### Definición 4.1

Una trayectoria  $P: N^+ \rightarrow N^-$  es aumentante para el flujo  $x$

Si  $x(j) < c^+(j) \quad \forall j \in P^+$

y  $x(j) > c^-(j) \quad \forall j \in P^-$

Claramente, el flujo  $x$  puede mejorarse a través de una trayectoria aumentante. En efecto, puede garantizarse la existencia de un número  $\alpha > 0$  tal que:

$$x'(j) = x(j) + \alpha e_p(j) = \begin{cases} x(j) + \alpha & j \in P^+ \\ x(j) - \alpha & j \in P^- \\ x(j) & j \notin P^+ \cup P^- \end{cases}$$

Es importante notar que el flujo  $x'$  construido de esta manera cumple con todas las restricciones del problema para todos los valores de  $\alpha$  que cumplan

$$\alpha \leq c^+(j) - x(j) \quad \forall j \in P^+$$

$$\alpha \leq x(j) - c^-(j) \quad \forall j \in P^-$$

Ya que con esto  $c^-(j) \leq x(j) \leq c^+(j)$ , para toda  $j \in A$ , además

$$\operatorname{div} x' = \operatorname{div} x + \alpha \operatorname{div} e_p(i)$$

Pero

$$\begin{aligned} \operatorname{div} e_p(i) &= 1 && \text{si } i \in N^+ \\ \operatorname{div} e_p(i) &= -1 && \text{si } i \in N^- \\ \operatorname{div} e_p(i) &= 0 && \text{si } i \notin N^+ \cup N^- \\ \alpha &> 0 \end{aligned}$$

Esto implica que  $x'$  se conserva en todos los nodos fuera de  $N^+ \cup N^-$  y también que:

$$[\text{flujo } x' \text{ de } N^+ \text{ a } N^-] = [\text{flujo } x \text{ de } N^+ \text{ a } N^-] + \alpha$$

De aquí que  $x'$  sea mejor que  $x$ .

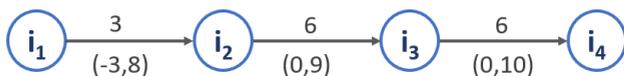
Al valor máximo del número  $\alpha$  se le llama capacidad incremental de la trayectoria  $P$ . Se dice que una trayectoria  $P: N^+ \rightarrow N^-$  tiene capacidad ilimitada si su capacidad incremental no es finita; es decir, si:

$$\begin{aligned} c^+(j) &= +\infty && \forall j \in P^+ \\ c^-(j) &= -\infty && \forall j \in P^- \end{aligned}$$

### Ejemplo 4.3

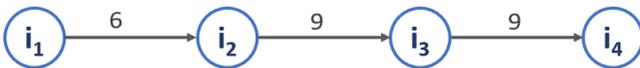
En la figura 4.4 se muestran trayectorias aumentantes, con capacidades incrementales 3, 5 y 1 respectivamente:

a) Flujo inicial

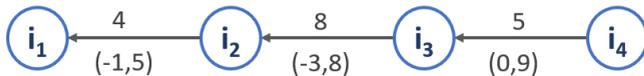


**Figura 4.4 a**  
Red con flujo inicial en arcos

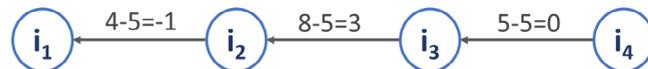
Flujo actualizado:  $\alpha = \min\{8-3, 9-6, 10-6\} = 3$



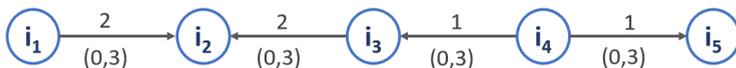
b) Flujo inicial



Flujo actualizado:  $\alpha = \min\{4-(-1), 8-(-3), 5-0\} = 5$



c) Flujo inicial



Flujo actualizado:  $\alpha = \min\{3-2, 2-0, 1-0, 3-1\} = 1$



La siguiente es una trayectoria no aumentante



Ya que se tiene  $\alpha = 0$  en el segundo arco.

**Figura 4.4b**  
Red con flujo actualizado aumentando 3 unidades

**Figura 4.4c**  
Red con flujo inicial en arcos

**Figura 4.4d**  
Red con flujo actualizado igual a 5

**Figura 4.4e**  
Red con flujo inicial

**Figura 4.4f**  
Red con flujo actualizado igual a 1

**Figura 4.4g**  
Red con trayectoria no aumentante

## Teorema de flujo máximo-corte mínimo (Ford y Fulkerson)

Suponga que existe al menos un flujo  $x$  que satisface todas las restricciones del problema, entonces:

Supremo en problemas de flujo máximo	=	Mínimo en problemas de corte mínimo
--------------------------------------	---	-------------------------------------

### Demostración

La demostración de este teorema es constructiva por lo cual constituye un algoritmo para resolver ambos problemas. En vista del resultado que afirma que el supremo en el problema de flujo máximo es menor o igual que el mínimo en el problema del corte mínimo, basta exhibir un flujo  $x$  y un corte  $Q$  tales que:

$$[\text{flujo de } x \text{ de } N^+ \text{ a } N^-] = C^+(Q)$$

Para ello, considere cualquier flujo  $x$  en la red  $G$  tal que se conserva en todos los nodos fuera de  $N^+ \cup N^-$  y construya un corte aplicando

- i)  $S = N^+$
- ii) Si  $i \in N^+$ 
  - $j \sim (i, i') \in A$  y  $x(j) < c^+(j)$  o
  - $j \sim (i', i) \in A$  y  $x(j) > c^-(j)$

Actualice  $S = N^+ \cup \{i\}$

Repita (ii) hasta que no sea posible agregar nodos a  $S$ .

Es importante señalar que los nodos de  $S$  son aquellos a los cuales aún puede enviarse flujo. Pueden presentarse entonces dos casos:  $S \cap N^- \neq \emptyset$  o bien  $S \cap N^- = \emptyset$ .

### Caso 1

Si  $S \cap N^- \neq \emptyset$  entonces aún puede enviarse flujo de  $N^+$  a  $N^-$ , es decir, dada la construcción de  $S$ , existe una trayectoria  $P: N^+ \rightarrow N^-$  tal que  $x(j) < c^+(j)$  para toda  $j \in P^+$  y  $x(j) > c^-(j)$  para toda  $j \in P^-$ . De aquí se concluye que  $P$  es una trayectoria aumentante para el flujo  $x$ . Sea  $\alpha$  la capacidad incremental de  $P$  y se define  $x$  como sigue:

$$x(j) = \begin{cases} x(j) + \alpha & \text{si } j \in P^+ \\ x(j) - \alpha & \text{si } j \in P^- \\ x(j) & \text{en otro caso} \end{cases}$$

Construya de nuevo el conjunto  $S$  con el procedimiento (i), y (ii) utilizando este flujo mejorado.

### Caso 2

Si  $S \cap N^- = \emptyset$  entonces  $Q = [S, N \setminus S]$  es un corte de  $Q$  por construcción se tiene que  $x(j) = c^+(j)$  para toda  $j \in Q^+$  y  $x(j) = c^-(j)$  para toda  $j \in Q^-$ ; entonces puesto que  $S \cap N^- = \emptyset$

$$\begin{aligned} [\text{flujo de } x \text{ de } N^+ \text{ a } N^-] &= y(N^+) = y(S) = e_Q x = \\ &= \sum_{j \in Q^+} x(j) - \sum_{j \in Q^-} x(j) = \sum_{j \in Q^+} c^+(j) - \sum_{j \in Q^-} c^-(j) = \\ &= C^+(Q) \end{aligned}$$

En este caso  $x$  es flujo máximo y  $Q$  es corte mínimo.

## Algoritmo de Ford y Fulkerson

**Propósito:** Determinar el flujo máximo de  $N^+$  a  $N^-$  en una red  $G$  en la cual no existen trayectorias de capacidad ilimitada.

### Descripción

**PASO 1** Determinar  $x$ , un flujo que satisfaga todas las restricciones del problema.

**PASO 2** Colorear los arcos de  $G$  de acuerdo con:

Verde	si	$c^-(j) < x(j) < c^+(j)$
Blanco	si	$c^-(j) = x(j) < c^+(j)$
Negro	si	$c^-(j) < x(j) = c^+(j)$
Rojo	si	$c^-(j) = x(j) = c^+(j)$

**PASO 3** Utilizar el algoritmo de enrutamiento para determinar una trayectoria compatible con la coloración (es decir, una trayectoria aumentante para  $x$ ).

**3.1** Si se determina la trayectoria  $P: N^+ \rightarrow N^-$  compatible con la coloración, calcular:

$$\alpha = \min \left\{ \begin{array}{ll} c^+(j) - x(j) & j \in P^+ \\ x(j) - c^-(j) & j \in P^- \end{array} \right\}$$

hacer  $x = x + \alpha e_p$  y regresar a 2

**3.2** Si se determina un corte  $Q: N^+ \downarrow N^-$  compatible con la coloración terminar con el flujo máximo  $x$  y el corte mínimo  $Q$  ya que este último satisface

$$c^+(j) = x(j) \quad j \in Q^+$$

$$c^-(j) = x(j) \quad j \in Q^-$$

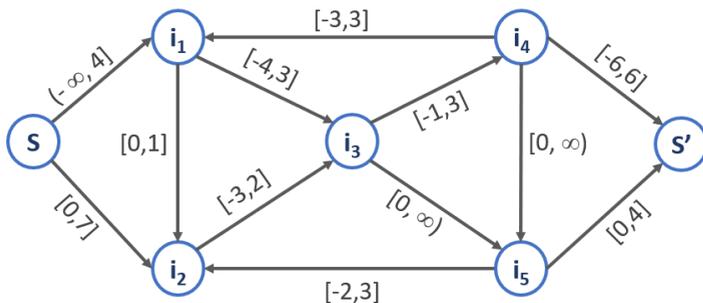
y por lo tanto el flujo  $x$  a través de

$$Q = \sum_{j \in Q^+} x(j) - \sum_{j \in Q^-} x(j) = C^+(Q)$$

**Nota:** El valor en la igualdad del teorema puede ser  $+\infty$  cuando el corte tiene capacidad ilimitada.

### Ejemplo 4.4

En la siguiente red se busca el flujo máximo que se puede enviar del nodo  $s$  al nodo  $s'$ , se aplica el algoritmo de flujo máximo, comenzando con un flujo cero  $x(j) = 0$  para toda  $j \in A$ . Esto es posible porque todos los intervalos de capacidad contienen al cero.



**Figura 4.5**  
Red con capacidades en arcos

En siguiente figura se muestran los flujos en cada arco que corresponden a las iteraciones 1 a 5.

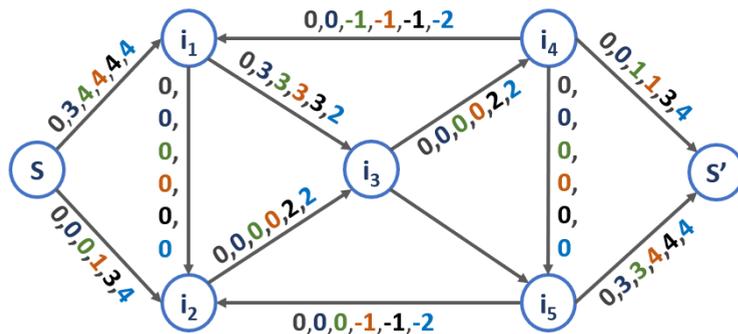


Figura 4.6 Red con los flujos de las trayectorias para cada iteración

Las correspondientes trayectorias aumentantes son:

- $P_1: s \rightarrow i_1 \rightarrow i_3 \rightarrow i_5 \rightarrow s' \quad \alpha_1 = 3$
- $P_2: s \rightarrow i_1 \rightarrow i_4 \rightarrow s' \quad \alpha_2 = 1$
- $P_3: s \rightarrow i_2 \leftarrow i_5 \rightarrow s' \quad \alpha_3 = 1$
- $P_4: s \rightarrow i_2 \rightarrow i_3 \rightarrow i_4 \rightarrow s' \quad \alpha_4 = 2$
- $P_5: s \rightarrow i_2 \leftarrow i_5 \leftarrow i_3 \leftarrow i_1 \leftarrow i_4 \rightarrow s' \quad \alpha_5 = 1$

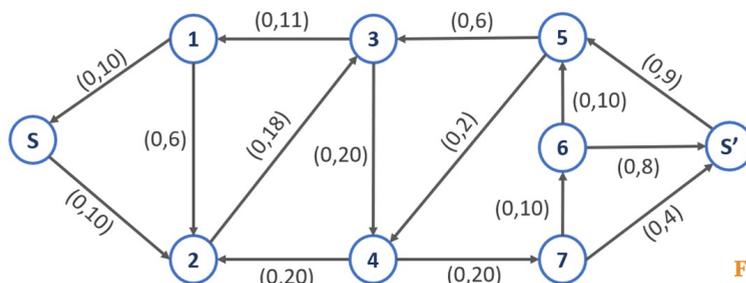
Las iteraciones a detalle se pueden consultar en el Anexo A de este libro.

La iteración cinco no nos proporciona otra trayectoria de flujo aumentante, solo el corte  $Q = [S, N \setminus S]$  correspondiente a  $S = \{s, i_2\}$ . Así, el flujo  $x$  en esta etapa resuelve el problema de flujo máximo mientras  $Q$  resuelve el problema de corte mínimo. El flujo de  $s$  a  $s'$  es 8, y este es el mismo valor de  $c^+(Q)$ .

Note que la cancelación del flujo toma lugar en la iteración 5 en los arcos  $(i_1, i_3)$  e  $(i_3, i_5)$  donde las trayectorias  $P_1$  y  $P_5$  corren contrarias una con otra. Tal cancelación corresponde al enrutamiento de un flujo anterior, y el algoritmo no funcionaría si esto no fuera permitido.

## Ejercicio

Considere la siguiente red y determine el flujo máximo y corte mínimo de  $S$  a  $S'$ .



**Figura 4.7** Red con capacidades en los arcos

### Justificación de convergencia del algoritmo

Suponga que las capacidades de los arcos  $c^+(j)$ ,  $c^-(j)$  y los valores de flujo inicial  $x(j)$  son conmensurables, es decir, son múltiplos de cierta cantidad  $q$ . En particular capacidades y flujos enteros son conmensurables con  $q = 1$ . De esta condición de conmensurabilidad se concluye que los números  $\alpha$  y  $x^1(j)$ , calculados durante el algoritmo, son también múltiplos de la cantidad  $q$ . De aquí que, en cada iteración, el flujo de  $N^+$  a  $N^-$  se incrementa al menos en la cantidad positiva  $q$  y por lo tanto se realiza un número finito de iteraciones si no existen trayectorias de capacidad ilimitada. Sin esta condición puede no converger el algoritmo o que converja dando una solución errada.

### Discriminación de arcos

La condición de conmensurabilidad puede eliminarse si el algoritmo de Ford y Fulkerson se utiliza con una pequeña modificación llamada criterio de discriminación de arcos. Este cambio consiste en utilizar, siempre que sea posible, arcos verdes durante la rutina de enrutamiento; cuando

se utilizan trayectorias de arco verdes, el flujo “mejorado” alcanzará la cota superior o la inferior para al menos un arco, el correspondiente al valor de  $\alpha$ . De este modo, para la siguiente iteración, al menos un arco se vuelve blanco o negro mientras que los que tenían estos colores no se alteran. Después de un número finito de iteraciones no existen trayectorias aumentantes con todos los arcos verdes, por lo que para continuar con el proceso de aumento de flujo (si esto es posible) deberá recurrirse a los arcos blancos o negros. En tal momento hay un conjunto  $S$  correspondiente a un corte  $Q$  que no contiene arcos verdes; es decir, todo arco  $j$  de  $Q$  satisface  $x(j) = c^+(j)$  o  $x(j) = c^-(j)$  o ambas. De aquí que

$$\sum_{j \in Q^+} x(j) - \sum_{j \in Q^-} x(j) = \text{flujo a través del corte } Q = \text{flujo } x \text{ de } N^+ \text{ a } N^-$$

#### 4.4 EL PROBLEMA DE FACTIBILIDAD DE FLUJO

El algoritmo de flujo máximo requiere inicialmente un flujo factible en la red, es decir, un flujo que satisfaga las restricciones de capacidad y conservación en arcos y nodos respectivamente. Se considera un caso especial de un modelo más general llamado el problema de distribución factible, donde además se tiene que las divergencias en los nodos deben coincidir con cantidades preestablecidas de oferta y demanda en los nodos.

##### Problema de distribución factible

Sean los intervalos de capacidad  $C(j) = [c^-(j), c^+(j)]$  para todo arco  $j$  y sean los valores de oferta  $b(i)$  para todo nodo  $i$ . Se desea determinar un flujo  $x$  tal que:

$$\begin{aligned} c^-(j) \leq x(j) \leq c^+(j) & \quad \forall j \in A \\ y(i) = b(i) & \quad \forall i \in N, \text{ con } (y = \text{div } x) \end{aligned}$$

La función  $b$  recibe el nombre de función de oferta, entendiéndose como demanda una oferta negativa. En el caso particular en que  $b = 0$  el problema recibe el nombre de problema de circulación factible.

Debido al principio de divergencia total, una condición necesaria para que exista solución al problema de distribución factible es  $b(N) = 0$ ; es decir, que la oferta total sea igual a la demanda total.

El problema puede ser más general si en vez de considerar un cierto valor para la divergencia, se permite que esta última pueda tomar valores en un determinado intervalo llamado de oferta y se formula como:

### Problema de flujo factible general

Determinar un flujo  $x$  tal que

$$\begin{aligned} x(j) \in C(j) & \quad \forall j \in A \\ y(i) \in C(i) & \quad \forall i \in N \text{ con } (y = \text{div } x). \end{aligned}$$

Donde  $C(j)$  es el intervalo de capacidad para el arco  $j$  y  $C(i)$  es el intervalo de oferta para el nodo  $i$ .

Puede verificarse fácilmente que el problema de distribución factible, circulación y flujo factibles son equivalentes.

## Teorema de distribución factible (Gale y Hoffman)

El problema de distribución factible tiene solución si y solo si  $b(N) = 0$  y  $b(S) \leq C^+(Q) \quad \forall Q = [S, N \setminus S]$  con  $S \subset N$ .

### Demostración

#### Condición necesaria

La condición  $b(N) = 0$  debe cumplirse como se estableció en la formulación del problema. Por otro lado, según el principio de divergencia, si  $S \subset N$  y  $Q = [S, N \setminus S]$

$$b(S) = [\text{divergencia de } x \text{ desde } S] = \text{flujo de } x \text{ a través de } Q.$$

Si  $\text{div } x = b$  como  $x$  debe ser un flujo factible con respecto a las capacidades de los arcos.

$$b(S) \in C(Q) = [c^-(Q), c^+(Q)]$$

En particular  $b(S) \leq c^+(Q)$ .

#### Condición suficiente

Se prueba constructivamente con el algoritmo de distribución factible.

### Algoritmo de distribución factible

**Propósito:** Resolver el problema de distribución factible, considerando un flujo factible con respecto a las capacidades de los arcos y buscando factibilidad en nodos.

Descripción

**PASO 1** Determinar un flujo factible  $x$  con respecto a las capacidades de los arcos.

**PASO 2** Sean  $N^+ = \{i \in N \mid b(i) > y(i)\}$ ,  $N^- = \{i \in N \mid b(i) < y(i)\}$

Si  $N^+ = N^- = \emptyset$  entonces el flujo  $x$  es la solución deseada, terminar.

Si  $N^+ \neq \emptyset$  y  $N^- \neq \emptyset$  colorear los arcos de la red de la siguiente manera:

Verde	si	$c^-(j) < x(j) < c^+(j)$
Blanco	si	$c^-(j) = x(j) < c^+(j)$
Negro	si	$c^-(j) < x(j) = c^+(j)$
Rojo	si	$c^-(j) = x(j) = c^+(j)$

ir al Paso 3.

**PASO 3** Aplicar el algoritmo de enrutamiento a la red.

**3.1** Si se determina una trayectoria  $P: N^+ \rightarrow N^-$  compatible con la coloración, calcular:

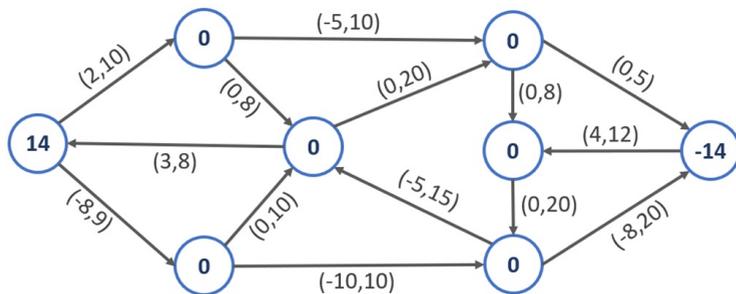
$$\alpha = \min \begin{cases} c^+(j) - x(j) & j \in P^+ \\ x(j) - c^-(j) & j \in P^- \\ b(i) - y(i) & i \text{ nodo inicial de } P \text{ (en } N^+) \\ y(i) - b(i) & i \text{ nodo terminal de } P \text{ (en } N^-) \end{cases}$$

Hacer  $x' = x + \alpha e_p$  y regresar al Paso 2.

**3.2** Si se determina un corte  $Q = [S, N \setminus S]$  compatible con la coloración, terminar, no existe solución al problema.

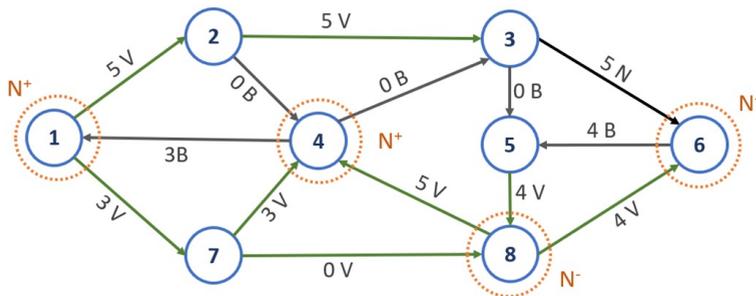
### Ejemplo 4.5

Usando el algoritmo de distribución factible, determine un flujo factible en la red siguiente:



**Figura 4.8**  
Red con divergencias en nodos

**Paso 1** Se inicia con el flujo  $x$  factible con respecto a los arcos. En la figura se muestra en los arcos el flujo y el color del arco, así como los conjuntos  $N^+$  y  $N^-$  de nodos cuya divergencia es mayor o menor que la demanda.



**Figura 4.9**  
Red con flujo factible en arcos

**Paso 2**

$i$	1	2	3	4	5	6	7	8
$y(i)$	5	0	0	-5	0	-5	0	+5
$b(i)$	14	0	0	0	0	-14	0	0

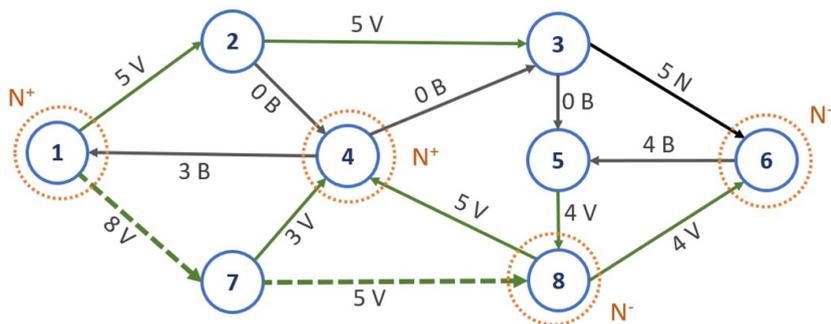
Entonces  $N^+ = \{1, 4\}$  y  $N^- = \{6, 8\}$

**Paso 3** Se obtiene la trayectoria compatible con la coloración:

$$P: 1 \rightarrow 7 \rightarrow 8$$

En la figura 4.10 se muestra la trayectoria con arcos más gruesos y el flujo actualizado en los arcos, así como la coloración.

$$\alpha = \min \{9 - 3, 10, 14 - 5, 5\} = 5$$



**Figura 4.10** Red con trayectoria compatible con la coloración

Iteración 1

**Paso 2**

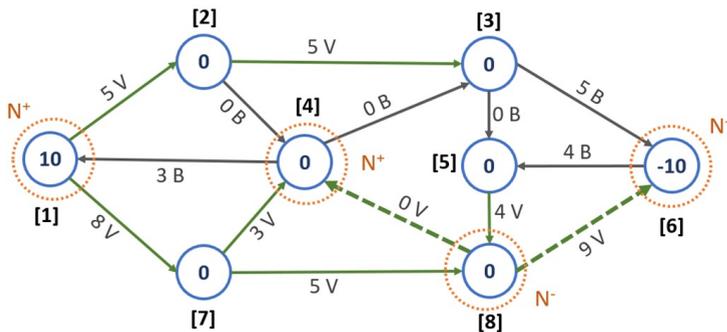
i	1	2	3	4	5	6	7	8
y(i)	10	0	0	-5	0	-5	0	0
b(i)	14	0	0	0	0	-14	0	0

De donde  $N^+ = \{1, 4\}$  y  $N^- = \{6\}$

**Paso 3**

Con la coloración correspondiente y  $P: 4 \leftarrow 8 \rightarrow 6$

y  $\alpha = \min \{10, 16, 5, 9\} = 5$ , con el nuevo flujo que se muestra en la figura 4.11.



**Figura 4.11**  
Red actualizada con trayectoria compatible

Iteración 2

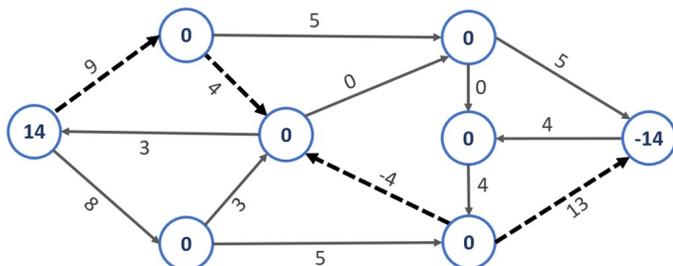
**Paso 2** Las divergencias y las ofertas son

i	1	2	3	4	5	6	7	8
y(i)	10	0	0	0	0	-10	0	0
b(i)	14	0	0	0	0	-14	0	0

**Paso 3** De donde  $N^+ = \{1\}$  y  $N^- = \{6\}$ , y la trayectoria compatible con la coloración

$$P: 1 \rightarrow 2 \rightarrow 4 \leftarrow 8 \rightarrow 6$$

$$\begin{aligned} \text{Con } \alpha &= \min \{10 - 5, 8 - 0, 0 - (-5), 20 - 9, 14 - 10, -10 - (-14)\} = \\ &= \min \{5, 8, 5, 11, 4, 4\} = 4 \end{aligned}$$



**Figura 4.12**  
Red actualizada con trayectoria compatible

Iteración 3

Pasos 2 y 3

i	1	2	3	4	5	6	7	8
y(i)	14	0	0	0	0	-10	0	0

$$N^+ = N^- = \phi \text{ con } y(i) = b(i) \quad \forall i$$

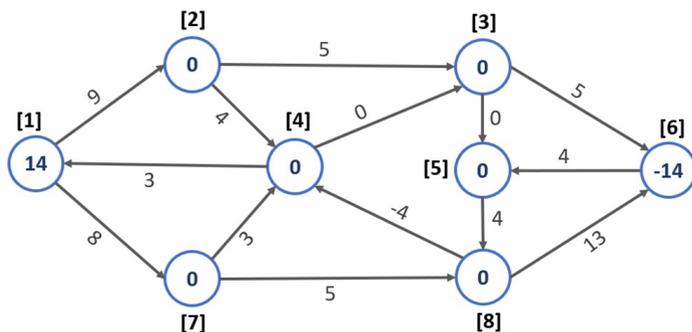


Figura 4.13 Red con flujo compatible con arcos y nodos

Algoritmo de rectificación de flujo

**Propósito:** Resolver el problema de distribución factible, considerando un flujo factible con respecto a los nodos y buscando factibilidad en arcos.

Descripción

**Paso 1** Determine un flujo  $x$  que es factible con respecto a las restricciones de divergencias, es decir  $\text{div } x = b$

**Paso 2** Defina los conjuntos de arcos:

$$A^+ = \{j \in A | x(j) > c^+(j)\} \quad A^- = \{j \in A | x(j) < c^-(j)\}$$

**2.1** Si  $A^+ = A^- = \phi$ , entonces  $x$  es la solución deseada, terminar.

**2.2** Si existe  $j \in A^+$  o  $j \in A^-$ , colorear los arcos de la red, de la siguiente manera:

verde	si	$c^-(j) < x(j) < c^+(j)$
blanco	si	$x(j) \leq c^-(j); x(j) < c^+(j)$
negro	si	$x(j) \geq c^+(j); x(j) > c^-(j)$
rojo	si	$c^-(j) = x(j) = c^+(j)$

ir al paso 3

**Paso 3** Aplique el lema de Minty:

**3.1** Si se determina un circuito elemental  $P$  compatible con la coloración, que contenga a  $j$  calcular:

$$\alpha = \min \begin{cases} c^+(j) - x(j) & j \in P^+ \\ x(j) - c^-(j) & j \in P^- \\ \bar{\alpha} \end{cases}$$

donde

$$\bar{\alpha} = \min \begin{cases} c^+(j) - x(j) & j \in A^+ \\ x(j) - c^+(j) & j \in A^- \end{cases}$$

Hacer  $x' = x + \alpha e_p$  e ir a 2

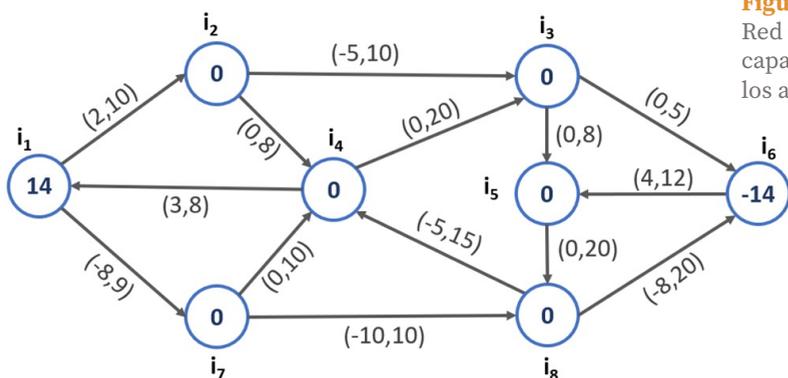
**3.2** Si se determina un corte  $Q = [S, N \setminus S]$  compatible con la coloración que contenga a  $j$  terminar, ya que en este caso no existe solución al problema.

Una alternativa natural para resolver el problema de distribución factible consiste en determinar inicialmente un flujo factible con res-

pecto a las restricciones de oferta en los nodos y posiblemente no factible con respecto a las restricciones de capacidad sobre los arcos. Al igual que en el algoritmo de distribución factible se construye en cada iteración, un flujo que “rompa menos” las restricciones de capacidad en el sentido de que el flujo definido sobre el arco se acerque más a las cotas de capacidad en cada iteración.

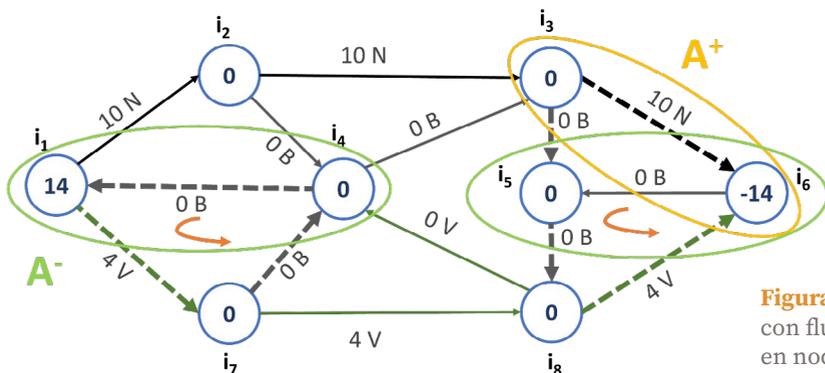
### Ejemplo 4.6

Usando el algoritmo de rectificación de flujo, determine un flujo factible en la red de la figura siguiente:



**Figura 4.14**  
Red con capacidades en los arcos

**Paso 1** Se determina un flujo factible con respecto a los nodos como se muestra en la figura 4.15.



**Figura 4.15** Red con flujo factible en nodos y pintado de arcos

**Paso 2** Se definen los conjuntos de arcos  $A^+$  y  $A^-$  y se colorean los arcos como se muestra en la figura 4.15.

$$A^+ = \{(i_3, i_6)\} \quad A^- = \{(i_4, i_1), (i_6, i_5)\}$$

**Paso 3** Se aplica el lema de Minty y se encuentran los siguientes circuitos:

$$P_1: i_4 \rightarrow i_1 \rightarrow i_7 \rightarrow i_4 \quad \text{Contiene al arco } (i_4, i_1) \text{ que está en } A^-$$

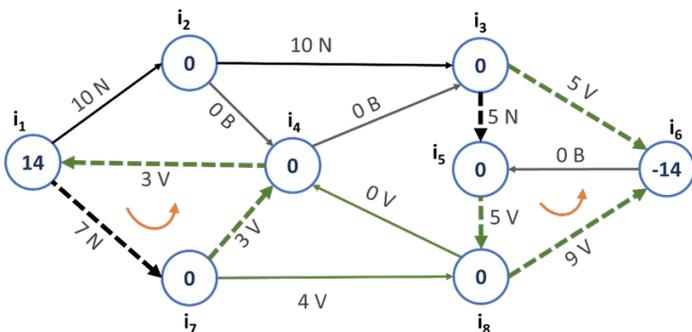
$$P_2: i_6 \leftarrow i_3 \rightarrow i_5 \rightarrow i_8 \rightarrow i_6 \quad \text{Contiene al arco } (i_3, i_6) \text{ que está en } A^+$$

Para ambos circuitos se tiene:

$$\text{Para } P_1: \alpha_1 = \min \{9 - 4, 10 - 0, 3 - 0\} = \{5, 10, 3\} = 3$$

$$\text{Para } P_2: \alpha_2 = \min \{8 - 0, 20 - 0, 20 - 4, 10 - 5\} = \{8, 20, 16, 5\} = 5$$

Resultando el flujo que se muestra en la figura 4.16

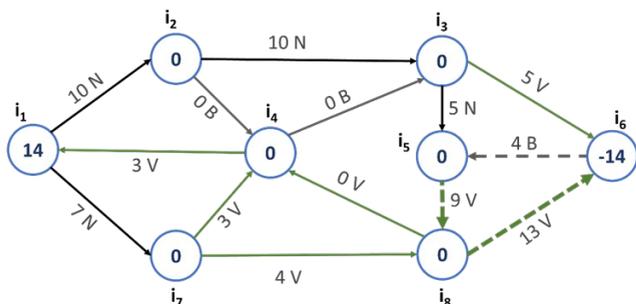


**Figura 4.16**  
Flujo resultante de los circuitos

Iteración 1

**Paso 2**  $A^+ = \phi \quad A^- = \{(i_6, i_5)\}$

**Paso 3** Sea  $P_3: i_6 \rightarrow i_5 \rightarrow i_8 \rightarrow i_6$  que contiene a  $(i_6, i_5)$  con  $\alpha = \min \{20 - 5, 20 - 9, 4 - 0\} = 4$  el flujo resultante es el que se muestra en la figura 4.17 donde además para este flujo  $A^+ = A^- = \phi$  por lo cual se termina el algoritmo con esta solución factible en nodos y arcos.



**Figura 4.17**  
Flujo factible en arcos y nodos

### 4.5 DESCRIPCIÓN DEL ALGORITMO DE ETIQUETADO DE FORD FULKERSON

El método que se describirá a continuación resuelve el problema de flujo máximo a través de etiquetar nodos. A grandes rasgos consiste en lo siguiente: Primero, se etiqueta el nodo origen  $s$  con dos etiquetas  $[s, \infty]$  indicando que en este nodo se dispone de cualquier cantidad de flujo. Después, si  $j$  es un nodo etiquetado y puede enviarse flujo del nodo  $j$  al nodo  $i$ , recibe dos etiquetas  $[+j, x(ji)]$ . La primera etiqueta será de la forma  $+j$  si  $i \in P^+(j)$ , es decir, si puede aumentarse flujo a través del arco  $j$ . Será de la forma  $-j$  si  $i \in P^-(j)$ , es decir, si puede disminuirse el flujo a través del arco  $(j, i)$ . La segunda etiqueta es la cantidad de flujo que puede enviarse de  $j$  a  $i$  y se calculará por lo tanto como el mínimo entre  $x(ji)$  y  $h$  donde:

$$h = \begin{cases} c_{ji}^+ & \text{si } i \in P^+(j) \\ x_{ij} & \text{si } i \in P^-(j) \end{cases}$$

Debe observarse que si  $h = 0$  no puede enviarse flujo de  $j$  a  $i$ , por lo que no se etiquetará con  $[j, x(ji)]$ . Aquí  $c_{ji}^+$  es la capacidad máxima del arco.

Este proceso de asignación de etiquetas a nodos se repetirá mientras sea posible. Si el nodo destino  $t$  recibe etiquetas, entonces dado el modo de etiquetar, existe una trayectoria aumentante de  $s$  a  $t$  con una capacidad incremental igual a  $x(t)$  y por lo tanto se procederá a determinar esta 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 considere el conjunto de arcos que tienen extremo inicial etiquetado y extremo final no etiquetado; este conjunto forma un corte de capacidad igual al valor del último flujo definido y por lo tanto este es máximo.

Note la similitud de este algoritmo con la demostración del teorema de flujo máximo - corte mínimo.

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

## Algoritmo de Ford y Fulkerson

**Propósito:** Determinar el flujo máximo entre el origen y destino en una red  $G$ .

### Descripción

**Paso 1** Iniciar con cualquier flujo factible  $x$

**Paso 2** Etiquetar el origen con  $[s, \infty]$

**Paso 3** Elegir un nodo etiquetado y no examinado; sea  $j$  dicho nodo y sean  $[\pm j, x(ji)]$  sus etiquetas.

**3.1** Para todo  $i \in P^+(j)$  que no esté etiquetado y tal que  $x_{ji} < c_{ji}^+$  asignar la etiqueta  $[+j, x(i)]$ , donde  $x(i) = \min\{x(j), c_{ji}^+ - x_{ji}\}$

**3.2** Para todo  $i \in P^-(j)$  que no esté etiquetado y tal que  $c_{ji}^+ > 0$  asignar la etiqueta  $[-j, x(i)]$ , donde  $x(i) = \min\{x(j), x_{ij} - c_{ij}^-\}$

Se puede decir ahora que el nodo  $j$  ha sido examinado

**Paso 4** Repetir el paso 3 hasta que suceda:

**4.1** El nodo destino  $t$  no tiene etiqueta y todos los nodos etiquetados han sido examinados. Terminar, ya que el flujo factible  $x$  es el máximo

**4.2** El nodo  $t$  recibe etiqueta. Ir al paso 5

**Paso 5** Sea  $y = t$

**5.1** Si la etiqueta de  $y$  es de la forma  $[+z, x(y)]$ ,  
hacer  $x_{zy} = x_{zy} + x(t)$

**5.2** Si la etiqueta de  $y$  es de la forma  $[-z, x(y)]$ ,  
hacer  $x_{yz} = x_{yz} - x(t)$

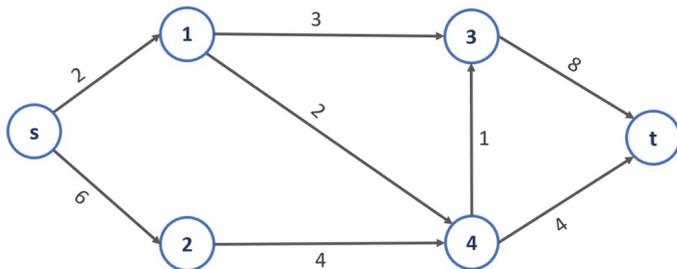
**Paso 6** **6.1** Si  $z = s$ , borrar todas las etiquetas y regresar al paso 2 actualizando el flujo en la trayectoria.

**6.2** Si  $z \neq s$ , hacer  $y = z$  y regresar al paso 5.

## E

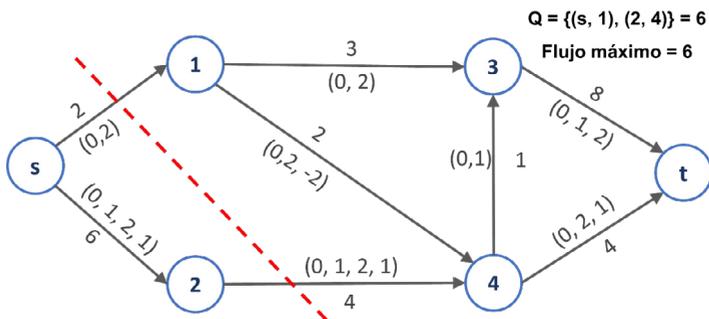
## Ejemplo 4.7

Determine el flujo máximo de  $s$  a  $t$  en la siguiente red usando el algoritmo de Ford y Fulkerson.



**Figura 4.18**  
Red con  
capacidades  
de flujo en los  
arcos

En siguiente figura se muestran los números correspondientes al flujo actualizado en cada arco por cada trayectoria encontrada y que corresponden a las iteraciones 1 a 5.



**Figura 4.19** Red con los flujos de las trayectorias para cada iteración

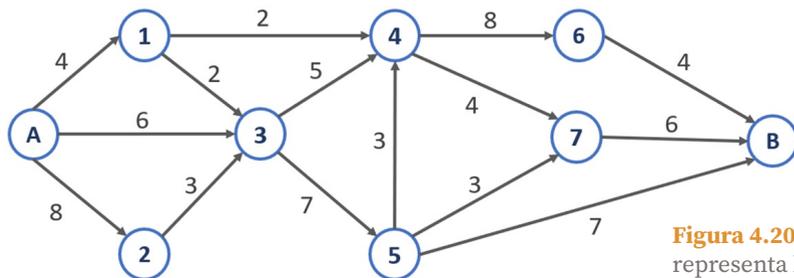
Las correspondientes trayectorias aumentantes son:

- $P_1: s \rightarrow 1 \rightarrow 4 \rightarrow t$
- $P_2: s \rightarrow 2 \rightarrow 4 \rightarrow 3 \rightarrow t$
- $P_3: s \rightarrow 2 \rightarrow 4 \leftarrow 1 \rightarrow 3 \rightarrow t$
- $P_4: s \rightarrow 2 \rightarrow 4 \rightarrow t$

E

Ejercicio

Un gran número de personas viajan en automóvil de la ciudad A hacia 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 hacia 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). Determine dónde deberán colocarse las casetas si se desea incurrir en el mínimo costo. La red que representa la problemática se muestra en la figura 4.20



**Figura 4.20** Red que representa las rutas de A hacia B

Observe que, en términos de redes, se desea obtener un conjunto de arcos (tramos donde deberán construirse las casetas) de manera tal que si eliminamos esos arcos de la red ya no existan caminos de A hacia 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 esta corresponde a la de mínimo costo. Para ello se determinará el flujo máximo de A hacia B mediante el algoritmo de Ford y Fulkerson.

#### 4.6 COMPLEJIDAD COMPUTACIONAL

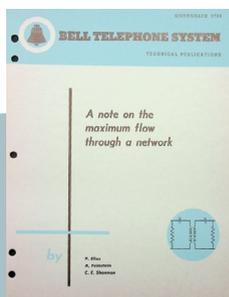
Los cuatro algoritmos vistos en este capítulo tienen la misma complejidad computacional, y se parte del peor caso en el algoritmo de etiquetado, recordemos que, en cada iteración, excepto la última, cuando el sumidero no se puede etiquetar, el algoritmo realiza un aumento.

Es fácil ver que cada aumento requiere tiempo  $O(m)$  porque el método de búsqueda examina cualquier arco o cualquier nodo como máximo una vez. Por lo tanto, la complejidad del algoritmo de etiquetado es  $O(m)$  veces el número de aumentos. ¿Cuántos aumentos puede realizar el algoritmo? Si todas las capacidades del arco son enteras y acotadas por un número finito  $U$ , la capacidad del corte  $(s, N - \{s\})$  es como máximo  $nU$ .

Por lo tanto, el valor de flujo máximo está acotado por  $nU$ .

El algoritmo de etiquetado aumenta el valor del flujo en al menos 1 unidad en cualquier aumento. Como consecuencia, terminará dentro de  $nU$  aumentos, por lo que  $O(nmU)$  es un límite en el tiempo de ejecución del algoritmo de etiquetado. En el anexo B se presenta otro algoritmo con menor tiempo de ejecución.

## 4.7 NOTAS HISTÓRICAS

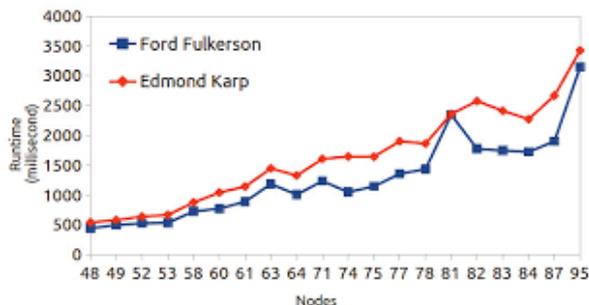


El artículo de Ford y Fulkerson (1956) en el problema de flujo máximo estableció el celebrado teorema de flujo máximo-corte mínimo. Fulkerson y Dantzig (1955) y Elías, Feinstein y Shannon (1956) establecieron independientemente este resultado. Ford y Fulkerson y Elías et al., resolvieron el problema de flujo máximo a través de algoritmos

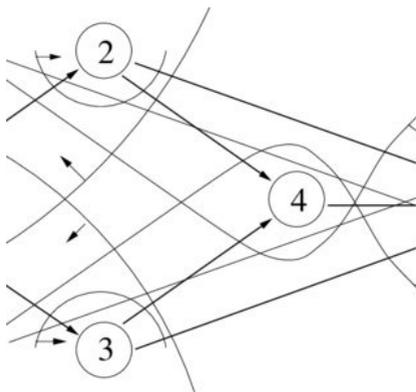
de trayectorias aumentantes, mientras Fulkerson y Dantzig (1955) lo resolvieron usando el método simplex especializado para redes.

Desafortunadamente, el algoritmo de Ford y Fulkerson corre en tiempo pseudopolinomial; más aún, para redes con capacidades irracionales el algoritmo desarrolla una sucesión infinita de trayectorias aumentantes que pueden converger a un valor diferente del valor de máximo flujo. Existen varias versiones mejoradas que superan esta limitación.

En la primera versión del teorema, todos los intervalos de capacidad se toman de la forma  $[0, c]$ , y aparecen nodos sencillos en lugar de conjuntos de nodos  $N^+$  y  $N^-$ . Tales restricciones no representan una pérdida significativa de generalidad, pues se pueden usar trucos para reformular el problema; sin embargo, el uso de intervalos arbitrarios y nodos múltiples fuente y sumidero hacen más fácil su aplicación.



En años más recientes J. Edmonds, y R.M. Karp (1972) demostraron que, si se usa una búsqueda de primera amplitud en la subrutina de pintado de la red, el algoritmo terminará (aun sin conmensurabilidad de los datos), y de hecho los pasos requeridos son del orden  $O(|N| |A|^2)$  (por lo que es para el caso digráfico no peor que  $O(|N|^5)$ ). El efecto de la búsqueda de primera amplitud consiste en seleccionar en cada iteración una trayectoria aumentante con la menor cantidad de arcos posible. Lo mismo se puede cumplir usando una versión de trayectorias múltiples de la subrutina de pintado de la red.



Las facetas de un conjunto poliédrico determinadas por las desigualdades de Gale Hoffman

El teorema de distribución factible fue publicado por primera vez por Gale (1957) en el caso en que  $c^-(j) \leq 0 \leq c^+(j)$  y por Hoffman (1960) en el caso de intervalos de capacidad cerrados arbitrarios. Ambos resultados en realidad datan de un artículo que no se publicó elaborado por ambos autores en 1956.

El algoritmo de rectificación de flujo se puede ver como una simplificación de un método de Minty (1962).

# Monotone networks\*

BY G. J. MINTY†

*Department of Mathematics, Duke University, Durham, N.C., U.S.A.*

*(Communicated by J. L. Synge, F.R.S.—Received 12 June 1959—  
Revised 4 February 1960)*

Fundamental existence and uniqueness theorems for electrical networks of non-linear resistors are proved in an abstract form, as theorems of pure mathematics. The two groups from which the 'currents' and 'voltage drops' are drawn are permitted to be either the real numbers, or discrete subgroups of the reals. It is found that the uniqueness theory is derivable from extremum principles for certain convex functions associated with the networks, and that the existence theory is derivable from a single new theorem of graph theory.

The abstract approach, besides revealing the logical structure of the subject more clearly than the 'concrete' approach, also (1) reveals the mathematical problem of solving a non-linear network to be identical with certain extremum problems arising in non-electrical applications, (2) contributes a numerical method, since the constructions for the discrete case are algorithmic, and (3) permits the application of the theorems to problems of pure mathematics.

Applications are not fully discussed; they will be treated at greater length in the appropriate technical journals.

## 1. INTRODUCTION

A well-known theorem, proved incorrectly by Poincaré (1901) and correctly by Chuard (1922), states that all subdeterminants of the incidence matrix of a directed graph are  $+1$ ,  $-1$ , or  $0$ . This fact has been exploited (implicitly, via the simplex method) to derive integer-valued solutions to certain special linear-programming problems. Efficient algorithms for the solution of such problems have been devised, notably by Ford & Fulkerson (1957*a, b*), using only the addition-operation of the real numbers.

The work of the present treatment began as an attempt to discover the broadest class of problems to which such methods were applicable; however, in its present form, it is an investigation of certain relationships between simply ordered groups and networks (directed graphs) culminating in proofs of existence and uniqueness theorems. Solution algorithms are contained implicitly in the proofs of the existence theorems. An electrical application is carried along with the theory; it has been preferred over the other applications largely for aesthetic reasons.

## 2. PRELIMINARIES

The empty set is called  $\phi$ . The real number system, with all its usual structure, is called  $R$ . An  $n$ -tuple of reals may be written as a vector:  $x = (x_1, \dots, x_n)$ ; the zero vector is denoted by  $\theta$ . The symbol  $\|x\|$  means  $\max_i |x_i|$ , and satisfies the triangle

\* This paper has its origins in work done for Project Michigan (Contract DA-36-039 SC-52654, U.S. Sig. C.); abstracts of various aspects were presented to the American Mathematical Society on 28 November 1958 and 21 November 1959.

† Now at the University of Washington, Seattle, Washington, U.S.A.

1

2

3

4

5

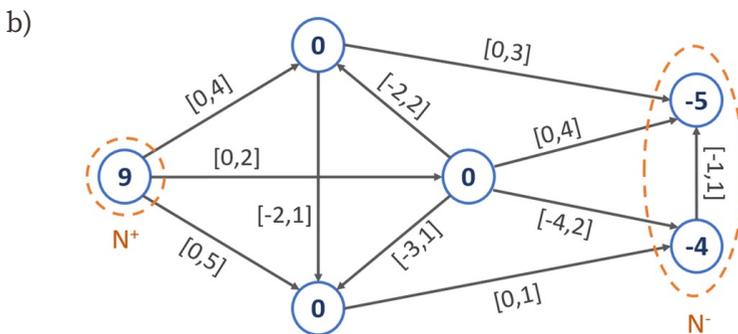
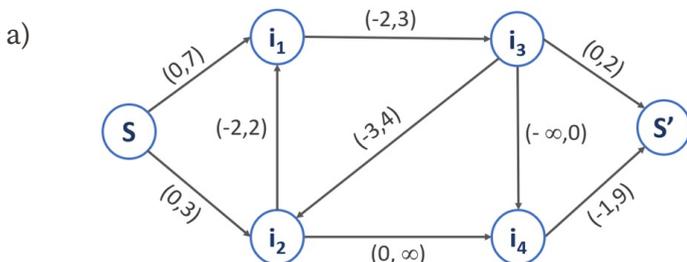
6

7

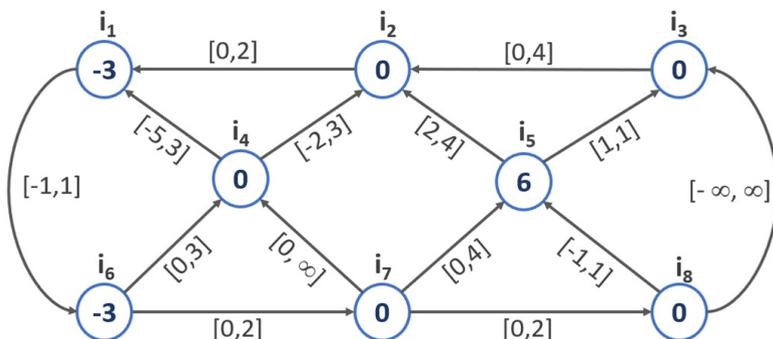
8

### 4.8 EJERCICIOS PROPUESTOS

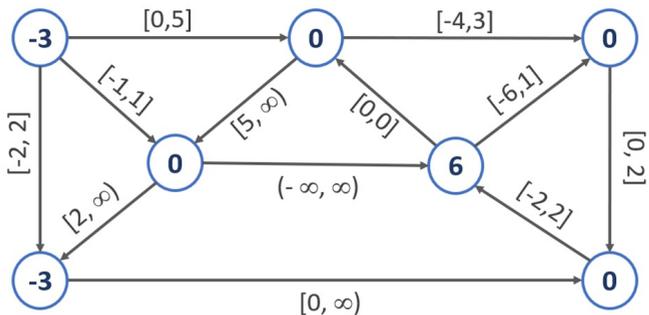
1. Resuelva el problema de flujo máximo-corte mínimo en las redes siguientes:



2. Resuelva el problema de flujo factible usando distribución factible.



3. Resuelva el problema de flujo factible representado en la red siguiente utilizando el algoritmo de rectificación de flujo. Las ofertas son  $i_2 = 5$ ,  $i_7 = -5$ ,  $i_j = 0$  en los demás nodos



1

2

3

4

5

6

7

8

# Árbol de expansión mínima

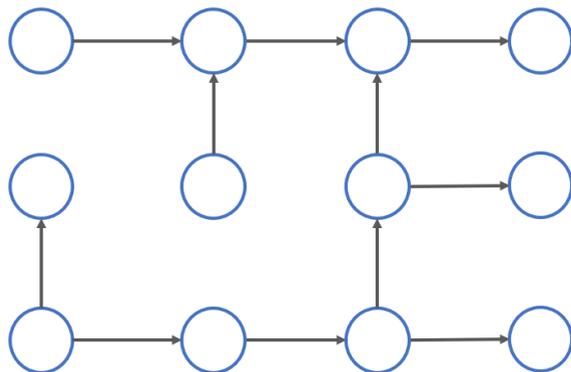
## 5.1 INTRODUCCIÓN

Como ya se ha mencionado en el Capítulo 1, el problema de árbol de expansión mínima juega un papel importante en los problemas de redes de flujo. Como veremos en este capítulo, para resolver el problema de ruta más corta se necesita construir un árbol con raíz; asimismo, para resolver redes de flujo a costo mínimo se usan árboles para encontrar la solución. En este capítulo veremos el modelo de árbol de expansión mínima y definiremos un árbol de expansión  $T$  de una red  $G$  como una subgráfica acíclica conectada que conecta a todos los nodos.

Con mayor exactitud podemos definir un árbol de la siguiente manera:

### **D** Definición 5.1

Una red se dice árbol si es conectada y sin circuitos. Por ejemplo, en la red de la figura 5.1 se muestra un árbol.



**Figura 5.1**  
Ejemplo de un árbol

La definición anterior excluye aquellas redes que constan de un solo nodo.

Hay una serie de características de un árbol que podemos enunciar en el siguiente teorema:

### T Teorema 5.1

Un árbol es una red (o gráfica) caracterizada por cualquiera de los siguientes postulados:

- Conectada y sin circuitos
- Existe una trayectoria única entre cada par de nodos
- No tiene circuitos, pero exactamente uno se forma al añadir un arco
- Es conectada, pero deja de serlo si algún arco se elimina

### D Demostración

a  $\rightarrow$  b) Es inmediato, pues si existe un par de nodos con dos trayectorias distintas que los una, se tendrá un circuito.

- b  $\rightarrow$  c) Se tiene que la red no tiene circuitos. Si  $j \leq (i, i')$  y hay una trayectoria única al añadir un arco, tendremos un circuito único.
- c  $\rightarrow$  d) Suponemos que no es conectada, entonces se añade un arco que une dos componentes no conectados y no se forma un circuito (lo que es imposible ya que por hipótesis al añadir un arco se forma un circuito).
- d  $\rightarrow$  a) Es inmediato que es conectada; supongamos que tiene circuitos, entonces podemos eliminar un arco y la red no se desconecta, pero esto contradice a d) por lo cual no debe tener circuitos.

**Nota:** las equivalencias de árbol no requieren que el número de nodos sea finito.

## E

**Corolario 5.1**

Sea  $G$  una red con  $m$  nodos. Entonces cualquiera de los postulados siguientes caracteriza a un árbol:

- e)  $G$  es un árbol
- f)  $G$  tiene  $m - 1$  arcos y no tiene circuitos
- g)  $G$  tiene  $m - 1$  arcos y es conectada

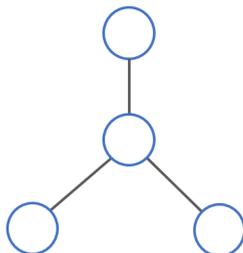
Un nodo se dice terminal si su grado (número de arcos que inciden) es 1.

**LEMA**

En una red la suma de los grados de los nodos es par.

**Proposición 5.1**

Sea  $G$  un árbol. Entonces  $G$  tiene al menos dos nodos terminales. Ver figura 5.2



**Figura 5.2** Nodos terminales en un árbol

Suponga que  $G$  tiene  $m$  nodos y  $m-1$  arcos. Entonces la suma de los grados de los nodos es  $2(m-1)$ .

Si  $G$  no tiene al menos dos nodos terminales, es sencillo concluir que la suma de los grados de los nodos será mayor que  $2(m-1)$ .

**Proposición 5.2**

Sea  $G$  un árbol. Cualquiera de las siguientes operaciones produce otro árbol.

- a) Se elimina un nodo terminal y el arco que incide
- b) Se añade un nodo y se conecta a  $G$  con un arco

**Prueba para saber cuándo una red es un árbol**

Existe una forma de determinar algorítmicamente cuándo una red dada con  $(|N| \leq 2)$  es un árbol. Seleccione cualquier nodo  $s$  y aplique el algoritmo de pintado de la red, con todos los arcos pintados de verde para obtener un enrutamiento máximo  $\Theta$  con base en  $N^+ = \{s\}$ .

Si este es un enrutamiento para todo  $N$  (es decir,  $\Theta(i)$  se define para cualquier  $i \neq s$ ) y usa cada arco (esto es, cada arco  $j$  se designa como  $\Theta(i)$  para algún nodo  $i$ , y es necesariamente único de acuerdo con la definición de enrutamiento) entonces la red es un árbol.

Si el enrutamiento tiene estas propiedades, entonces la red es conec-tada, ya que existe una trayectoria de  $s$  a cualquier nodo, y no puede tener circuitos elementales, pero si  $P$  fuera un circuito elemental, entonces cada arco  $j$  de  $P$  estaría en  $\Theta(i)$  para uno de los dos nodos de  $P$  adyacente a él; ya que un circuito elemental tiene igual número de nodos que de arcos, esto implicaría para cada nodo  $i$  de  $P$  que  $\Theta(i)$  está definido y es un arco de  $P$ , entonces el enrutamiento no propor-ciona una trayectoria que une cualquier nodo de  $P$  con  $s$ . Por lo que la red debe ser un árbol.

Por otro lado, si la red es un árbol y se aplica este procedimiento, cada nodo se alcanzará debido a la conexidad. Además, los nodos y los arcos usados por el enrutamiento constituirán un árbol por el argumento ya dado, y por lo tanto cualquier par de nodos  $i, i'$  se pueden unir por una trayectoria elemental limitada a los arcos que usan el enrutamiento.

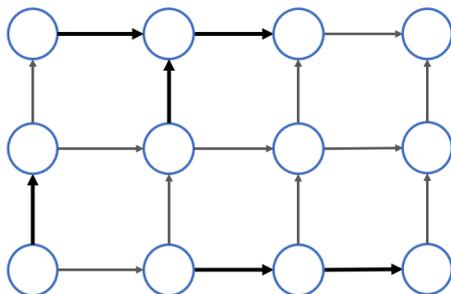
Lo anterior muestra que la noción de un enrutamiento con base en  $s$  es equivalente a la de árbol con raíz, donde el nodo  $s$  es la raíz del árbol. Un árbol con raíz también recibe el nombre de arborescencia.

## Bosques y árboles de expansión

### Definición 5.2

Un conjunto de arcos  $F \subset A$  se dice que forma un bosque en una red  $G$  si cada componente de la subred compuesta por arcos de  $F$  y los nodos

incidentes a ella es un árbol. En otras palabras,  $F$  es un bosque si y solo si no se incluyen circuitos elementales en  $F$ . Por ejemplo, los arcos más gruesos en la figura 5.3 forman un bosque, si incluimos los nodos incidentes se tiene en cada caso un árbol.



**Figura 5.3**  
Ejemplo de un bosque

$F$  forma un bosque en  $G$  si y solo si es el conjunto de arcos usado por algún enrutamiento  $\Theta$  (no necesariamente todos los de  $N$  con base  $N^+ \subset N$ ).

La conexión entre bosques y flujos es la siguiente: Un conjunto de arcos  $F$  forma un bosque si y solo si no existe una circulación distinta de cero cuyo soporte esté incluido en  $F$ , esto es, ninguna circulación  $x$  tal que  $x(j) = 0$  para toda  $j$  que no esté en  $F$ , pero  $x(j) \neq 0$  para al menos una  $j \in F$ .

Las circulaciones  $x$  se caracterizan como soluciones del sistema.

$$\sum_{j \in A} e(i, j) x(j) = 0 \quad \text{para toda } i \in N$$

Sea  $e^j(i) = e(i, j)$ , así  $e^j$  denota a la “columna” de  $E$  correspondiente al arco  $j$ . El sistema entonces se convierte en

$$\sum_{j \in A} x(j) e^j = 0$$

En otras palabras, una circulación distinta de cero puede interpretarse como una elección de coeficientes  $x(j)$  que producen una relación de dependencia lineal entre las columnas correspondientes de  $E$ . De aquí se sigue que un conjunto de arcos  $F$  forma un bosque en  $G$  si y solo si las columnas correspondientes de la matriz de incidencia de  $G$  son linealmente independientes.

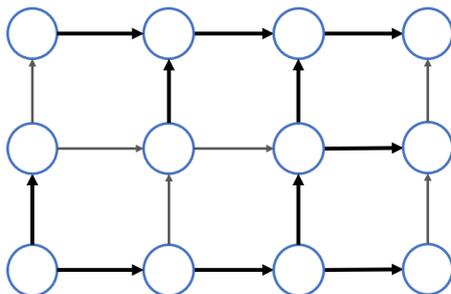
Es natural de este hecho pensar en el estudio de conjuntos de columnas que forma una base para el espacio de las columnas de  $E$  (es decir, el subespacio lineal de  $R^N$  generado por todas las  $e^j$ ,  $j \in A$ ), ya que entre otras cosas este da una descripción del rango de  $E$ .

### Definición 5.3

Un bosque máximo en  $G$  se define como un bosque que no está estrictamente contenido en otro bosque de  $G$ .

### Definición 5.4

Un árbol de expansión de  $G$  es un árbol que pasa por cada nodo de  $G$ , como en la figura 5.4



**Figura 5.4**  
Ejemplo de un árbol de expansión

## T Teorema 5.2

### Teorema de expansión

Para cualquier red  $G$ , las siguientes propiedades de un conjunto de arcos  $F \subset A$  son equivalentes:

1.  $F$  forma un bosque máximo para  $G$ .
2.  $F$  forma un árbol de expansión para cada componente de  $G$  que no sea un nodo único.
3.  $F$  forma un bosque tal que  $|F| = |N| - p$  donde  $p$  es el número de componentes de  $G$ .
4. Las columnas  $e^j$  de la matriz de incidencia  $E$  correspondiente a los arcos  $j \in F$  forman una base para el espacio de las columnas de  $E$ .

## D Demostración

La equivalencia de las propiedades 1 y 4 es clara de las observaciones precedentes acerca de la independencia lineal.

1  $\rightarrow$  2) Note de la maximalidad de  $F$  que cada componente de  $G$  que no sea un nodo aislado debe incluir un arco de  $F$  y por lo tanto uno de los árboles componentes de  $F$ .

Sea  $T$  tal árbol y sea  $S$  su conjunto de nodos, la componente de  $G$  que incluya a  $T$  no puede tener nodos fuera de  $S$ . Si así fuera entonces el corte  $Q = [S, N \setminus S]$  sería no vacío y cualquier arco en  $Q$  se podría agregar a  $F$  sin romper la propiedad de bosque. Así, cada árbol componente del bosque corresponde a una componente de  $G$  que tiene el mismo conjunto de nodos que el árbol, y cada componente que no es un nodo aislado corresponde de esta forma a un árbol.

2  $\rightarrow$  3) Es inmediata del hecho de que el número de arcos en un árbol siempre es igual al número de nodos menos uno, de aquí se generaliza que para cualquier componente  $G'$  de  $G$ :

Número de arcos en un bosque que está en $G'$	=	Número de nodos incidentes al bosque que está en $G'$	-	Número de árboles del bosque que está en $G'$	$\leq$	Número de nodos en $G'$	-1
---	---	---	---	---	--------	-------------------------	----

Entonces cualquier conjunto de arcos  $F$  que forman un bosque en  $G$  satisface:

$$|F| \leq |N| - [\text{número de componentes de } G]$$

y se cumple que  $F$  es máximo, así la propiedad 3 implica la propiedad 1.

## S

## Corolario 5.2

En una red conectada (con al menos un arco) un bosque máximo es lo mismo que un árbol de expansión.

En programación lineal la propiedad 4 tiene un significado especial, ya que en programación los problemas involucran la restricción  $Ex = b$ , que se puede escribir:

$$\sum_{j \in A} x(j)e^j = b$$

Es importante saber cuáles conjuntos  $F \subset A$  son tales que, los elementos  $e^j$  para  $j \in F$  forman una base del espacio de las columnas (suponiendo que contiene a  $b$ ). Cada uno de éstos determina una única  $x$  que satisface la ecuación y teniendo  $x(j) = 0$  para toda  $j$  que no está en  $F$ .

El método simplex depende de generar una secuencia de tales conjuntos  $F$ . Geométricamente, estos conjuntos pueden identificarse con los árboles de expansión para la red, suponiendo que la red es conectada.

Asociado con el concepto de árbol con raíz o arborescencia se encuentra el problema de ruta más corta que se aborda en el capítulo 6 y se presentan algoritmos de solución para las distintas variantes que hay, también se desarrollan ejemplos para cada algoritmo.

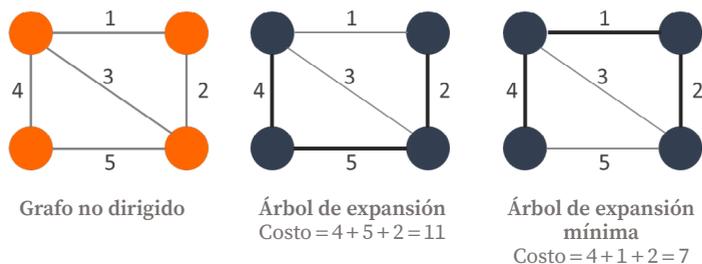
## 5.2 ÁRBOL DE EXPANSIÓN MÍNIMA

Una característica especial de los problemas de árbol de expansión mínima es que los arcos no son dirigidos ( $i, j$ ) o ( $j, i$ ) se refieren al mismo arco.

### Definición 5.5

Dada una red  $G=(N, A)$  no dirigida con  $n=|N|$  nodos y  $m=|A|$  arcos con costos o longitudes  $c_{ij}$  asociados a cada arco  $(i, j) \in A$ , se desea encontrar un árbol de expansión, llamado el árbol de expansión mínima, que tenga el menor costo (o longitud) de los arcos que lo constituyen, medido como la suma de los costos de los arcos.

En la figura siguiente se muestra la diferencia entre red no dirigida, árbol de expansión y árbol de expansión mínima:



**Figura 5.5**  
Diferencia entre red no dirigida, árbol de expansión y árbol de expansión mínima

Los tres algoritmos que veremos en esta sección son el algoritmo de Kruskal, el algoritmo de Prim y el algoritmo de Sollin, todos con la característica de ser algoritmos de tipo “glotón”, en el sentido de que en cada paso aumentan un arco de costo mínimo como candidato de la lista, siempre y cuando este arco no forme ningún circuito con los arcos ya seleccionados. Los tres algoritmos mantienen un bosque que contiene arcos que ya han sido seleccionados y se aumenta uno o más arcos para incrementar el tamaño del bosque.

Para el algoritmo de Kruskal, la lista de candidatos es la red completa; para el algoritmo de Prim, el bosque es un único árbol más un conjunto de nodos aislados y la lista de candidatos contiene todos los arcos entre el árbol único y los nodos que no están en él. El algoritmo de Sollin es un híbrido que mantiene varias componentes en el bosque, como el algoritmo de Kruskal, pero se aumentan varios arcos en cada iteración, seleccionando (como en el algoritmo de Prim) el arco de costo mínimo que conecta cada componente del bosque a los nodos que no están en esa componente.

Ya que los algoritmos tipo glotón se usan en muchos otros contextos de optimización discreta, se puede ver qué generalizaciones del algoritmo de Kruskal resuelven una amplia clase de problemas combinatorios abstractos como los de optimización en matroides.

La programación matemática tiene otra forma útil de ver el problema de árbol de expansión mínima: se puede formular un problema de árbol de expansión mínima como un modelo de programación entera y usar argumentos de programación lineal para establecer otra prueba de la validación del algoritmo de Kruskal. Esta discusión sirve para varios propósitos:

1

2

3

4

5

6

7

8

1. Da otra visión útil de los árboles de expansión mínima.
2. Ilustra una técnica de prueba, vía programación lineal que ha demostrado ser poderosa en el campo de la optimización combinatoria.
3. Proporciona un puente entre el problema de árbol de expansión mínima y poliedros combinatorios (o poliedros enteros).

### 5.3 Algoritmo de Kruskal

**Propósito:** Determinar el árbol de expansión cuyo “costo o peso” es mínimo en una red conectada  $G$  con  $N$  nodos y función de costo  $C: A \rightarrow R$  conocida.

#### Descripción

- Paso 1** (Inicio) Ordene el conjunto de arcos en forma creciente respecto a la función de costos. Sean  $j_1, j_2, \dots, j_m$  los arcos ordenados, hacer  $k=0, 1, 2, \dots, n=1, I=1$  y  $\underline{A}=\phi$
- Paso 2** (Añadir arco) Si el arco  $j_i$  no forma circuito con el conjunto de arcos de  $\underline{A}$  entonces  $\underline{A} = \underline{A} \cup \{j_r\}$  y hacer  $k = k + 1$  y vaya a 3. De otra manera siga a 3 y no aumente  $k$ .
- Paso 3** (Criterio de Terminación) Si  $k < n - 1$  hacer  $l = l + 1$  y regresar a 2. En caso contrario  $T = [N, \underline{A}]$  es el árbol de expansión mínima.

Ejemplo 5.1

Considere la red de la figura 5.6 y determine el árbol de peso mínimo mediante el algoritmo de Kruskal.

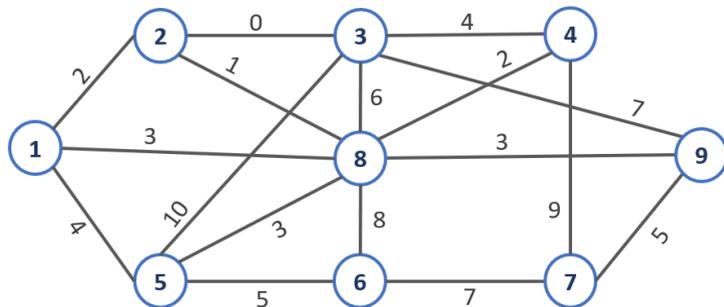


Figura 5.6 Red con pesos en los arcos

**Paso 1** Se ordenan los arcos:

- |                   |                   |                   |                   |
|-------------------|-------------------|-------------------|-------------------|
| $j_1 = (2, 3)$    | $j_2 = (2, 8)$    | $j_3 = (1, 2)$    | $j_4 = (8, 4)$    |
| $j_5 = (1, 8)$    | $j_6 = (5, 8)$    | $j_7 = (8, 9)$    | $j_8 = (1, 5)$    |
| $j_9 = (3, 4)$    | $j_{10} = (5, 6)$ | $j_{11} = (7, 9)$ | $j_{12} = (3, 8)$ |
| $j_{13} = (6, 7)$ | $j_{14} = (3, 9)$ | $j_{15} = (8, 6)$ | $j_{16} = (4, 7)$ |
| $j_{17} = (3, 5)$ |                   |                   |                   |

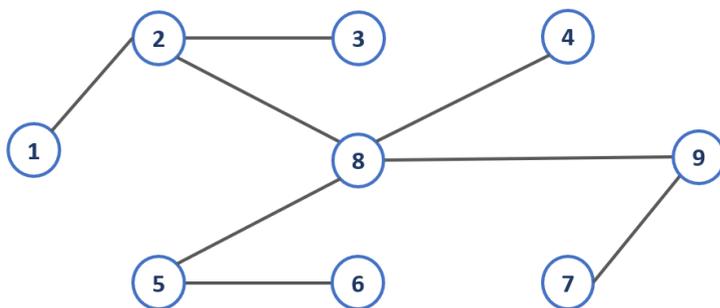
**Paso 2**  $\underline{A} = \{j_1\}$ ,  $k=1$ . Como  $k < n - 1 = 8$ , aún no se tiene el árbol.

En la tabla 5.1 siguiente se presenta un resumen de las operaciones realizadas en cada iteración. En la primera columna aparece el número de iteración, en la segunda, el arco que se agrega al conjunto  $\underline{A}$  para ir formando el árbol; por último, en la tercera columna se tiene el número de arcos que ya se han incluido en el árbol hasta esa iteración.

**Tabla 5.1** Iteraciones del algoritmo de Kruskal para el ejemplo 5.1

Número de iteración (I)	Arco agregado a A	Valor de K
1	$j_1 = (2,3)$	1
2	$j_2 = (2,8)$	2
3	$j_3 = (1,2)$	3
4	$j_4 = (8,4)$	4
5	ninguno, $j_5$ forma un circuito	4
6	$j_6 = (5,8)$	5
7	$j_7 = (8,9)$	6
8	ninguno, $j_8$ forma un circuito	6
9	ninguno, $j_9$ forma un circuito	6
10	$j_{10} = (5,6)$	7
11	$j_{11} = (7,9)$	8

Puesto que en la iteración 11 se tiene  $k = 8 = n - 1$ , se ha obtenido la solución óptima; la que está dada por la red de la figura 5.7 donde  $\underline{A} = \{j_1, j_2, j_3, j_4, j_6, j_7, j_{10}, j_{11}\}$



**Figura 5.7** Árbol de expansión mínima

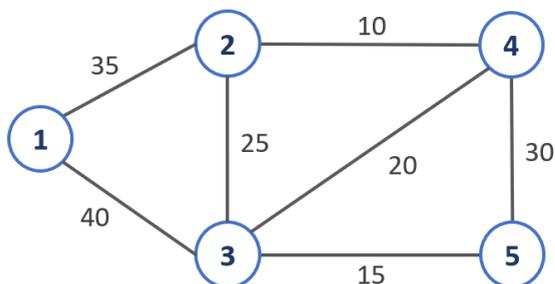
Este árbol de expansión tiene un peso de:

$$0 + 1 + 2 + 2 + 3 + 3 + 5 + 5 = 21$$

## E

## Ejemplo 5.2

Considere la red de la figura 5.8 y determine el árbol de peso mínimo mediante el algoritmo de Kruskal.



**Figura 5.8** Red con pesos en los arcos

**Paso 1** Se ordena la lista de arcos.

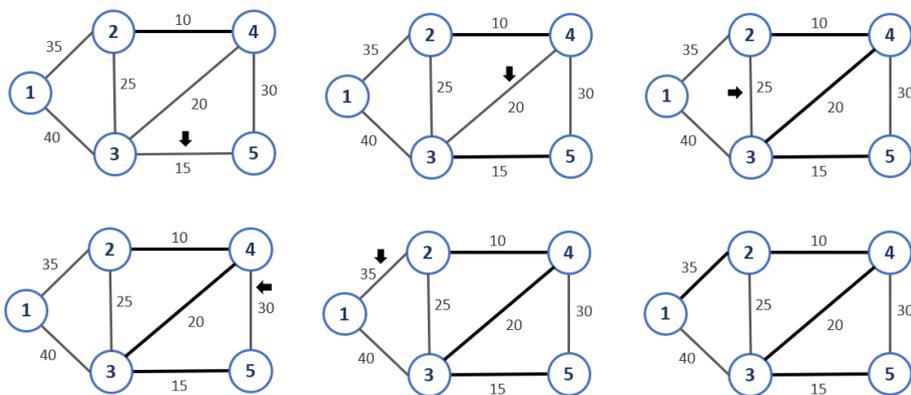
$j$	Arco	Valor
1	(2, 4)	10
2	(3, 5)	15
3	(3, 4)	20
4	(2, 3)	25
5	(4, 5)	30
6	(1, 2)	35
7	(1, 3)	40

**Pasos 2 y 3.** Se resumen las iteraciones del algoritmo para el ejemplo 5.2 en la tabla siguiente

**Tabla 5.2** Resumen de las iteraciones para el ejemplo 5.2

Paso 2		Paso 3	
No. de iteración	Arco por agregar	¿Hace ciclo?	Valor de K
1	(2,4)	No	1
2	(3,5)	No	2
3	(3,4)	No	3
4	(2,3)	Sí	3
5	(4,5)	Sí	3
6	(1,2)	No	4

La construcción del árbol de expansión mínima se presenta en la figura 5.9



**Figura 5.9** Construcción del árbol de expansión mínima

### Complejidad computacional del algoritmo

El tiempo de corrida del algoritmo de Kruskal se compone del tiempo para ordenar los arcos y el tiempo para detectar ciclos. Para una red arbitrariamente grande, el ordenamiento requiere  $O(m \log m) = O$

$(m \log n^2) = O(m \log n)$  de tiempo. El tiempo para detectar un ciclo depende del método que se use en este paso. Por ejemplo, la lista que se tiene en el ejemplo 2 y el segundo paso es (2, 4) y (3, 5). Se denotan estos conjuntos de nodos por los árboles  $N_1$  y  $N_2$  se guardan estos conjuntos en dos listas ligadas. Mientras se examina un arco  $(k, l)$  se busca en estas listas ligadas y se verifica si  $k$  y  $l$  están en la misma lista, si ambos nodos están en listas distintas se agrega el arco si no, se elimina. Si se aumenta el arco se unen las listas que contienen a  $k$  y  $l$  en una sola lista; esta estructura de datos requiere un tiempo  $O(n)$ . Usando esta estructura de datos el algoritmo de Kruskal corre en  $O(nm)$ .

## 5.4 ALGORITMO DE PRIM

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

A diferencia del algoritmo de Kruskal, la red construida en cada iteración es conexa y además tiene  $n - 1$  arcos, luego, esta red es un árbol expandido de la red 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 de expansión cuyo costo o peso es mínimo en una red conectada  $G$  con  $N$  nodos y función de peso o costo  $C : A \rightarrow R$  conocida.

### Descripción

**Paso 1** (Inicio) Sea  $x_0$  (arbitrario) elemento de  $N$  y  $k=0$ . Sea

$$N_0 = \{x_0\} \qquad A_0 = \phi$$

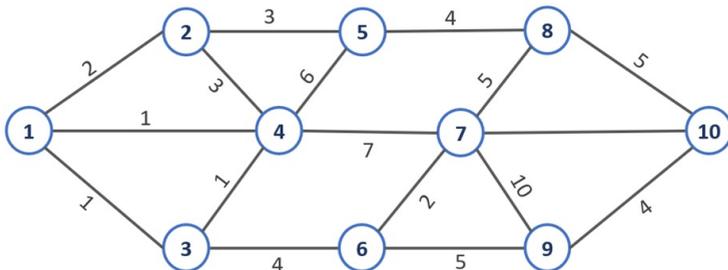
**Paso 2** (Añadir un arco). Sea  $F_k$  el conjunto de arcos de  $A$  que tienen exactamente un extremo en  $N_k$ . Sea  $j_k$  el arco de costo mínimo en  $F_k$  y denote por  $x_k$  el extremo de  $j_k$  que no pertenece a  $F_k$ . Hacer

$$N_{k+1} = N_k \cup \{x_k\} \qquad A_{k+1} = A_k \cup \{j_k\}$$

**Paso 3** Hacer  $k=k+1$ . Si  $k < n-1$  regrese a 2. En caso contrario, termine. La red  $T_{n-1} = [N_{n-1}, A_{n-1}]$  representa el árbol de expansión mínima de  $G$ .

### Ejemplo 5.3

Aplique el algoritmo de Prim y determine el árbol de expansión mínima en la red conectada de la figura 5.10



**Figura 5.10**  
Red con pesos en los arcos

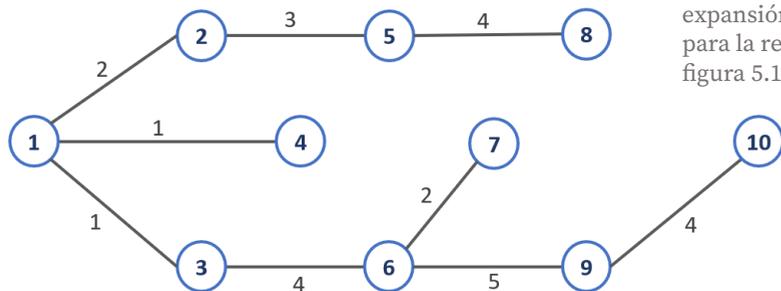
**Paso 1** Comenzaremos con  $x_0 = \{7\}$ .  $A_0 = \phi$

En la tabla 5.3 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 de la  $k$ -ésima iteración, en la tercera el arco de peso mínimo de  $c_k(j_k)$  y por último en las columnas cuarta y quinta respectivamente, los nodos y arcos que irán formando el árbol de expansión mínima de la red.

**Tabla 5.3** Resumen de las iteraciones para el ejemplo 5.3 usando el algoritmo de Prim

Número de iteración (K)	$F_K$	$j_K$	$N_K$	$A_K$
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	$\{(6,9), (7,9), (8,10)\}$	(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	$\{(8,10), (9,10), (7,8), (7,9)\}$	(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		(9,10)	$\{7,6,3,1,4,2,5,8,9,10\}$	$\{(6,7), (3,6), (1,3), (1,4), (1,2), (2,5), (5,8), (6,9), (9,10)\}$

En la iteración  $k = n - 1 = 9$  se observa que el árbol de peso mínimo generado por el algoritmo es como el que se muestra en la figura 5.11



**Figura 5.11** Árbol de expansión mínima para la red de la figura 5.10

Este árbol de expansión tiene un peso de:

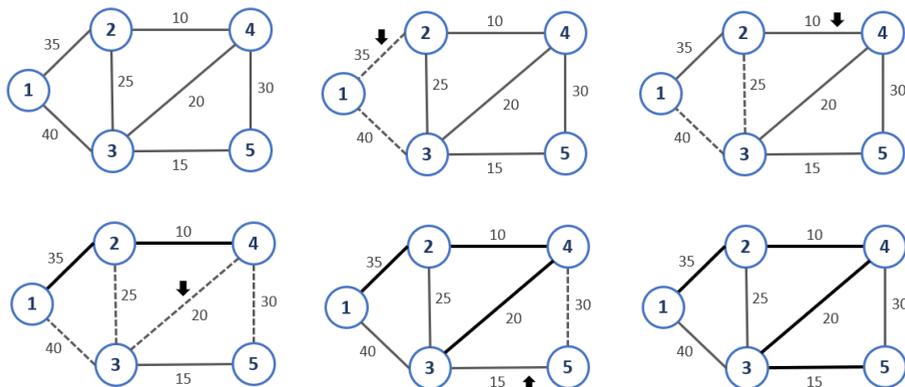
$$1 + 1 + 2 + 3 + 4 + 4 + 2 + 5 + 4 = 26$$

E

**Ejemplo 5.4**

La siguiente serie de redes muestran paso a paso el uso del algoritmo de Prim para resolver el problema de encontrar un árbol de expansión mínima.

**Figura 5.12** Construcción del árbol de expansión mínima



## Complejidad computacional del algoritmo

Para analizar el tiempo de ejecución del algoritmo de Prim, consideramos cada una de las  $n - 1$  iteraciones que el algoritmo desarrolla como si aumentara un arco a la vez al árbol, hasta que se tiene un árbol de expansión con  $n - 1$  arcos. En cada iteración el algoritmo selecciona el arco de costo mínimo en el corte  $[S, \underline{S}]$ . Si podemos revisar la lista completa de arcos para identificar el arco de costo mínimo, esta operación requiere un tiempo de ejecución de  $O(m)$ , lo que nos da una cota de tiempo de  $O(nm)$  para el algoritmo. Por lo cual podemos afirmar que el algoritmo se ejecuta en un tiempo  $O(mn)$ .

El cuello de botella en el algoritmo de Prim es la identificación del arco de costo mínimo en el corte  $[S, \underline{S}]$ . Podemos mejorar la eficiencia en este paso manteniendo dos índices para cada nodo  $j \in \underline{S}$ :

1. Una etiqueta para la distancia  $d(j)$  que representa el costo mínimo de los arcos en el corte incidente al nodo  $j$  que no está en  $S$ , es decir

$$d(j) = \min \{c_{ij} : (i, j) \in [S, \underline{S}]\}$$

2. Una etiqueta predecesora  $\text{pred}(j)$  que representa el otro punto final del arco de costo mínimo en el corte incidente al nodo  $j$ . Por ejemplo, en la tercera iteración del ejemplo 4.4 los tres arcos  $(1, 3)$ ,  $(2, 3)$  y  $(4, 3)$  en el corte son incidentes al nodo 3. Entre esos arcos el arco  $(4, 3)$  tiene el costo mínimo de 20.

Entonces  $d(3) = 20$  y  $\text{pred}(3) = 4$ . Para el mismo caso  $d(5) = 30$  y  $\text{pred}(5) = 4$ . Si mantenemos esos índices podemos encontrar fácilmente el costo mínimo de un arco en el corte; simplemente calculamos

$$\min \{d(j) \mid j \in S\}$$

si el nodo  $i$  alcanza este mínimo,  $(\text{pred}(i), i)$  es el arco de costo mínimo en el corte. Observe que, si movemos el nodo  $i$  desde  $S$  hasta  $\bar{S}$ , necesitamos actualizar las etiquetas distancia y predecesor solo para los nodos adyacentes al nodo  $i$ .

## 5.5 ALGORITMO DE SOLLIN

Podemos usar las condiciones de optimalidad para derivar otro algoritmo para el problema de árbol de expansión mínima. Este algoritmo lo podemos ver como una versión híbrida de los algoritmos de Kruskal y de Prim. Como en el algoritmo de Kruskal, el algoritmo de Sollin mantiene una colección de árboles de expansión de los nodos  $N_1, N_2, N_3, \dots$  y aumenta arcos a esa colección. Sin embargo, en cada iteración aumenta arcos de costo mínimo que emanan de esos árboles, una idea tomada del algoritmo de Prim. Como resultado se obtiene un algoritmo que usa estructuras de datos sencillas y corre en un tiempo  $O(m \log n)$ . El algoritmo de Sollin desarrolla repetidamente las dos siguientes operaciones básicas.

**Vecino más cercano:**  $(N_k, i_k, j_k)$ . Esta operación toma como inicio un árbol que une a los nodos  $N_k$  y determina un arco  $(i_k, j_k)$  de costo mínimo entre todos los arcos que emanan de  $N_k$ , es decir

$$C_{i_k j_k} = \min \{ C_{ij} \mid (i, j) \in A, i \in N_k, y j \notin N_k \}$$

Para desarrollar esta operación necesitamos revisar todos los arcos en las listas de adyacencia de los nodos en  $N_k$  y encontrar el arco de costo mínimo entre esos arcos que tenga un punto final que no esté en  $N_k$ .

**Acoplar  $(i_k, j_k)$ :** Esta operación toma como inicio dos nodos  $i_k$  y  $j_k$  y si los dos nodos están en distintos árboles, entonces une esos dos árboles en un solo árbol.

## Algoritmo de Sollin

**Propósito:** Resolver el problema de árbol de expansión mínima.

### Descripción

**Paso 1** Para cada  $i \in N$  haga  $N_i = \{i\}$ ,  $T^* = \emptyset$

**Paso 2** Mientras  $|T^*| < (n-1)$  entonces para cada árbol  $N_k$  haga  $(N_k, i_k, j_k)$ : donde el arco  $(i_k, j_k)$  es el arco de costo mínimo entre todos los que emanan de  $N_k$  es decir:

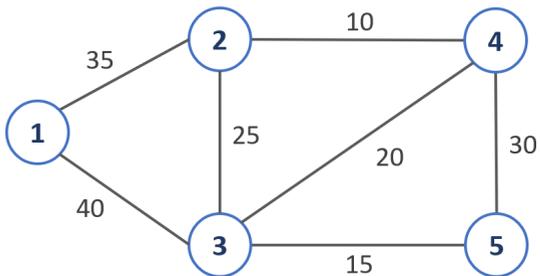
$$\min c_{ij} = \{(i, j) \mid (i, j) \in A, i \in N_k, j \notin N_k\}$$

y vaya al paso 3

**Paso 3** Si los nodos  $i_k$  y  $j_k$  están en distintos árboles una  $(i_k, j_k)$  y actualice  $T^* = T^* \cup \{(i_k, j_k)\}$  y regrese al paso 2.

### Ejemplo 5.5

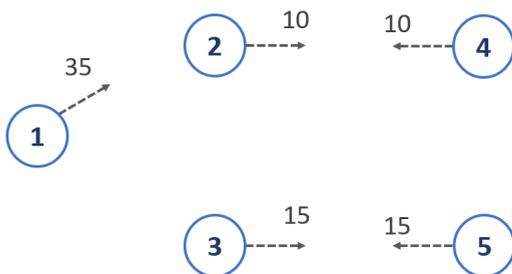
En la siguiente red encuentre el árbol de expansión mínima usando el algoritmo de Sollin.



**Figura 5.13**  
Red con pesos en los arcos

Se comienza con cinco árboles que conforman un bosque. Cada árbol consta de un solo nodo y el arco es el de menor costo que conecta cada nodo.

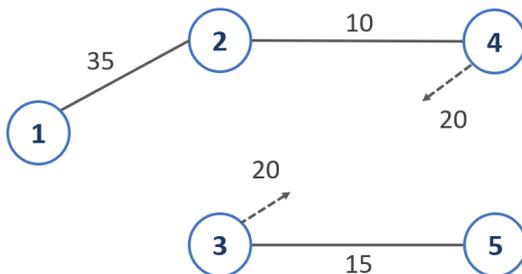
**Paso 1**



**Figura 5.14**  
Primer paso para la construcción del árbol

**Paso 2**

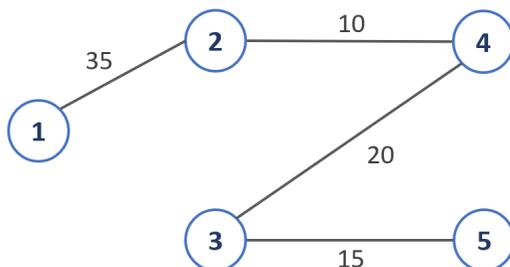
Se unen nodos reduciendo el número de árboles a solo dos. El arco de menor costo que emana de esos árboles es el (3,4).



**Figura 5.15**  
Segundo paso para la construcción del árbol

**Paso 3**

Se une ese arco y obtenemos el árbol.



**Figura 5.16**  
El árbol de expansión mínima resultante

## Complejidad computacional del algoritmo

Para poder analizar el tiempo de ejecución del algoritmo necesitamos discutir la estructura de datos necesaria para implantarlo. Mostraremos que el algoritmo desarrolla  $O(\log n)$  ejecuciones del ciclo del paso 2, y que se pueden desarrollar las operaciones vecino más cercano y acoplamiento en un tiempo  $O(m)$ . Estos resultados establecen una cota de tiempo de  $O(m \log n)$  para el algoritmo.

Almacenamos los nodos del árbol en una lista circular doblemente ligada, lo que nos permite visitar cada nodo del árbol comenzando en cualquier nodo. Asignamos una etiqueta numérica con cada nodo en la red; la etiqueta satisface las siguientes dos propiedades:

1. Nodos del mismo árbol que tienen la misma etiqueta.
2. Nodos de diferentes árboles con diferentes etiquetas. Al inicio del algoritmo, asignamos la etiqueta  $i$  a cada nodo  $i \in N$ .

Usando esta estructura de datos podemos verificar fácilmente cuándo un arco  $(i, j)$  tiene sus puntos finales en el mismo árbol. Contestamos esta pregunta verificando las etiquetas de los nodos  $i$  y  $j$ .

Esta observación implica que podemos desarrollar la operación de vecino más cercano para cada árbol en el bosque en un tiempo total de  $O(\sum_{i \in N} |A(i)|) = O(m)$ .

Desarrollamos las operaciones de acoplamiento en el ciclo del paso 2-3 usando el siguiente esquema iterativo: En cada iteración seleccionamos un árbol no examinado, por ejemplo,  $N_1$ , y consideramos el arco de costo mínimo  $(i_1, j_1)$  que emana de  $N_1$  (el nodo  $i_1$  está en  $N_1$  y el  $j_1$  puede o no estar en  $N_1$ ). Suponga que los nodos en  $N_1$  tienen la etiqueta  $\alpha$ . Si el nodo  $j_1$  también tiene la etiqueta  $\alpha$ , la iteración termina.

En caso contrario buscamos los nodos del árbol, por ejemplo,  $N_2$  que contenga al nodo  $j_1$  y les asignamos la etiqueta  $\alpha$ . Después, consideramos los arcos de costo mínimo  $(i_2, j_2)$  que emanan de  $N_2$ . Si el nodo  $j_2$  tiene etiqueta  $\alpha$ , la iteración termina; en caso contrario, buscamos en los nodos del árbol  $N_3$  que contenga al nodo  $j_2$  y les asignamos la etiqueta  $\alpha$ . Repetimos este proceso hasta que la iteración termina. Note que en cada iteración podemos asignar a los nodos de varios árboles la etiqueta del primer árbol. Cuando una iteración termina iniciamos una nueva iteración seleccionando otro árbol no examinado. Terminamos este proceso iterativo cuando hayamos examinado todos los árboles. De esta descripción se puede ver que este método asigna una etiqueta a cada nodo y se ejecuta en un tiempo  $O(n)$ .

Cada ejecución en el ciclo del algoritmo requiere un tiempo  $O(m)$ , obtenemos entonces una cota del número de ejecuciones del ciclo. Cada ejecución del ciclo reduce el número de árboles en el bosque por un factor de al menos dos porque acoplamos cada árbol en un árbol mayor. Esta observación tiene como consecuencia que desarrollaremos  $O(\log n)$  ejecuciones del ciclo. Se tiene entonces el siguiente teorema:

1

2

3

4

5

6

7

8

## Teorema 5.3

El algoritmo de Sollin tiene un tiempo de ejecución de  $O(m \log n)$ .

En este capítulo hemos descrito tres algoritmos para resolver el problema del árbol de expansión mínima. Todos ellos son fáciles de implantar y tienen excelentes tiempos de ejecución, además son muy eficientes en la práctica.

La tabla 5.4 siguiente resume la complejidad computacional de estos tres algoritmos:

**Tabla 5.4** Complejidad computacional de Kruskal, Prim y Sollin

	Tiempo de ejecución	Observaciones
Kruskal	$O(m + n \log n)$	<ol style="list-style-type: none"> <li>1. Examina arcos en orden no decreciente de sus costos más el tiempo necesario para ordenar <math>m</math> arcos y los incluye en un árbol de expansión si el arco que se aumenta no forma un ciclo con los arcos seleccionados.</li> <li>2. La prueba del algoritmo usa condiciones de optimalidad para una trayectoria.</li> <li>3. Es un algoritmo atractivo si los arcos ya se encuentran ordenados en forma creciente.</li> </ol>
Prim	$O(m + n \log n)$	<ol style="list-style-type: none"> <li>1. Mantiene un árbol de expansión en un subconjunto de nodos <math>S</math> y aumenta un arco de costos mínimos en el corte <math>[S, \underline{S}]</math>.</li> <li>2. La prueba del algoritmo usa condiciones de optimalidad para cortes.</li> <li>3. Se puede implantar usando una variedad de estructuras de apilamiento. Esta cota de ejecución se refiere al tiempo de la estructura de datos de Fibonacci.</li> </ol>
Sollin	$O(m \log n)$	<ol style="list-style-type: none"> <li>1. Mantiene una colección de árboles, en cada iteración aumenta un árbol de costo mínimo que emana de cada árbol.</li> <li>2. La prueba del algoritmo usa condiciones de optimalidad en cortes.</li> </ol>

## 5.6 ANÁLISIS DE SENSIBILIDAD

Una vez presentados los métodos de solución para el problema del árbol de expansión mínima, es interesante observar el efecto de eliminar un arco de la gráfica original. Es claro que, si este arco no pertenece a la solución óptima, la solución óptima del problema de encontrar el árbol de peso mínimo en la red resultante de remover dicho arco es la misma que para la red original. Sin embargo, si el arco 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.

Suponga que se ha obtenido el árbol expandido de peso mínimo  $T = [X, A^*]$  de una red conexa  $G = [X, A]$ , también suponga que, por alguna razón, es necesario eliminar un arco de la red  $G$  que, además, pertenece al árbol  $T$ . Considere, ahora, el problema de encontrar el árbol expandido de peso mínimo de la red resultante de eliminar dicho arco.

Sea  $j' = (i, i')$  el arco que debe eliminarse y sea  $G'$  la red resultante de tal eliminación, es decir,  $G' = [X, A - \{j'\}]$ . Entonces, el problema consiste en encontrar el árbol requerido. A continuación, desarrollamos un procedimiento para obtenerlo, aprovechando que ya se cuenta con el árbol  $T$ .

Al eliminar  $(i, i')$  de  $T$ , este se descompone en dos componentes conexas. Sean estas  $T_1 = [X_1, A_1]$  y  $T_2 = [X_2, A_2]$ . Observe que un extremo del arco  $j'$  está en  $X_1$  y el otro en  $X_2$ . Suponga, sin pérdida de generalidad, que  $i$  está en  $X_1$  y que  $i'$  está en  $X_2$ . Sea  $U$  el conjunto de arcos de  $G$  distintos de  $j'$  que tienen un extremo en  $X_1$  y otro en  $X_2$ . Observe 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  $G$ . Suponga

entonces, que,  $U$  no es vacío, sea  $u^*$  el arco de  $U$  tal que  $p(u^*) \leq p(u)$  para todo arco  $u \in U$ , y una constante  $p$ . La solución del problema se establece en la siguiente proposición:

### P Proposición 5.3

La red  $T^* = [X, A_1 \cup A_2 \cup \{u^*\}]$  es un árbol de expansión mínima de  $G'$ .

### D Demostración

Sean las redes conexas  $G_1 = [X_1, A_1]$  y  $G_2 = [X_2, A_2]$  donde  $A_i$  es el conjunto de arcos de  $A$  que tienen ambos extremos en  $X_i$  ( $i=1,2$ ). Se demostrará primero que  $T_1$  y  $T_2$  son los árboles de expansión mínima de  $G_1$  y  $G_2$  respectivamente.

Suponga que  $T_1$  no es un árbol de expansión mínima de  $G_1$ . Sea  $N = [X_1, U_1]$  el árbol de expansión mínima de dicha gráfica, se tiene entonces que  $p(N) < p(T_1)$ .

Sea  $T' = [X, U_1 \cup U_2 \cup \{j'\}]$ , note que  $T'$  es un árbol de expansión de  $G$ , ya que  $T'$  es acíclica debido a que el arco  $j'$  tiene un extremo en  $X_1$  y otro en  $X_2$  y las redes  $N$  y  $T_2$  son acíclicas.

$$\text{Ahora bien, } p(T') = p(U_1) + p(A_2) + p(j') = p(N) + p(A_2) + p(j')$$

$$\text{por otro lado } p(T) = p(A_1) + p(A_2) + p(j') = p(T_1) + p(A_2) + p(j')$$

$$\text{por lo tanto } p(T') < p(T)$$

Lo que contradice que  $T$  sea un árbol de expansión mínima de  $G$ . Por lo tanto,  $T_1$  es un árbol de expansión de peso mínimo de  $G_1$ . Análogamente, se prueba que  $T_2$  es un árbol de expansión mínima de  $G_2$ .

Por otro lado,  $G'$  puede expresarse como la red  $[X, A_1 \cup A_2 \cup j']$ , y puesto que  $p(u^*) \leq p(u)$  para todo arco  $u \in U$ , se tiene que  $T^*$  es un árbol de expansión mínima de  $G'$  y con esto se concluye la prueba. Este procedimiento lo podemos aplicar en el siguiente problema:

Suponga que se ha encontrado el árbol de peso mínimo  $T = [X, A^*]$  de la red  $G = [X, A]$  y suponga también que se modifica el peso de un arco  $j \in A^*$ . Se desea encontrar ahora el árbol de peso mínimo en la gráfica  $G$  con la función de peso, asociada a los arcos de  $G$ , modificada para el arco  $j$ .

Si el nuevo peso de  $j$  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  $j$  es mayor que su peso anterior. Es en este último caso cuando puede aplicarse el método expuesto anteriormente. Para encontrar la nueva solución, la única modificación que deberá hacerse al procedimiento será no excluir al arco  $j$  del conjunto  $U$ .

E

Ejemplo 5.6

Considere la red del ejemplo 5.1 y su árbol de expansión mínima. Suponga que se elimina el arco (2,8). La gráfica resultante  $G'$  se muestra en la figura 5.17

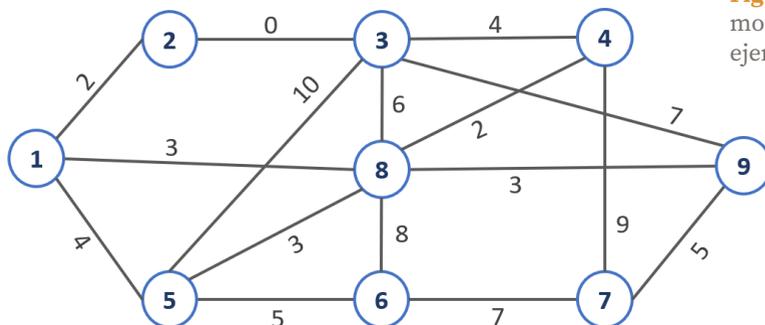
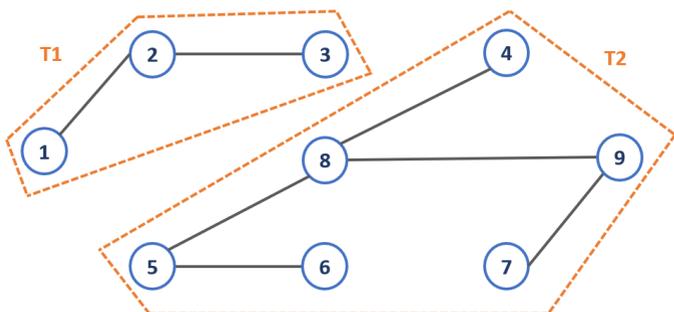


Figura 5.17 Red modificada del ejemplo 5.1

Determine el árbol de expansión mínima en esta nueva red. Las componentes conexas  $T_1$  y  $T_2$  en que se divide  $T$  al eliminar  $(2, 8)$  se muestran en la figura 5.18

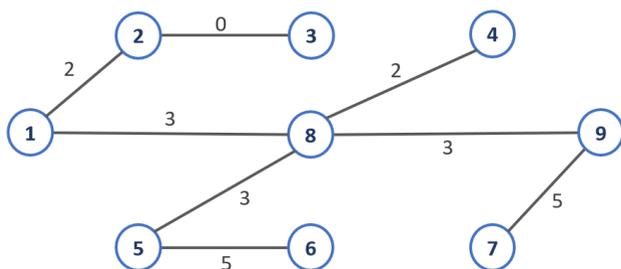


**Figura 5.18**  
Componentes conexas de  $T$

El conjunto  $U$  estará formado entonces, por los arcos siguientes, con sus respectivos pesos:

Arco	Peso	Arco	Peso
(1, 5)	4	(3, 5)	10
(1, 8)	3	(3, 8)	6
(3, 4)	4	(3, 9)	7

De aquí se tiene que  $u^* = (1, 8)$ . Por lo tanto, el árbol de expansión mínima de  $G'$  se muestra en la figura 5.19



**Figura 5.19**  
Árbol de expansión mínima construido

Cuyo peso está dado por:

$$0 + 2 + 2 + 3 + 3 + 3 + 5 + 5 = 23 = p(T) - p(2, 8) + p(1, 8)$$

## 5.7 NOTAS HISTÓRICAS



R.L. Graham

Los primeros algoritmos de árbol de expansión mínima se desarrollaron en 1926. El artículo de Graham and Hell (1985) presenta una reseña histórica de los algoritmos de árbol de expansión mínima.

Borůvka (1926) y Jarnick (1930) independientemente desarrollaron y resolvieron el problema. Más tarde otros investigadores redescubrieron esos algoritmos.



Pavol Hell

El algoritmo de Borůvka es un algoritmo codicioso para encontrar un árbol de expansión mínimo en un gráfico, o un bosque de expansión mínimo en el caso de un gráfico que no está conectado.



Otakar Borůvka

Kruskal (1956), Loberman y Weinberger (1957) independientemente desarrollaron el algoritmo que lleva el nombre de Kruskal.



Vojtěch Jarník



Joseph B. Kruskal

1

2

3

4

5

6

7

8



Robert Clay Prim

Prim (1957) desarrolló el algoritmo que lleva su nombre.

Sollin presentó su algoritmo, discutido en un seminario en 1961; nunca se publicó. Claude Berge, que estaba presente en el seminario, reportó este algoritmo en su libro *Berge and Ghouila-Houri* (1962).

El algoritmo fue redescubierto por Choquet en 1938; nuevamente por Florek, Łukasiewicz, Perkal, Steinhaus y Zubrzycki en 1951; y nuevamente por Georges Sollin en 1965. Más tarde, diversos investigadores descubrieron que el algoritmo de Prim es similar al de Jarnick.

Una versión mejorada del algoritmo de Kruskal fue desarrollada por Tarjan en 1984 y se ejecuta en  $O(m \alpha(n, m))$  más el tiempo requerido para ordenar los arcos.  $\alpha(n, m)$  es la función de Ackermann que para efectos prácticos es menor que 6.

Gabow, Galil, Spencer y Tarjan en 1986 presentaron una variedad del algoritmo de Prim que corre en un tiempo  $O(m \log \beta(m, n))$  con la función  $\beta(m, n)$  definida como  $\beta(m, n) = \min \{i \mid \log^{(i)}(m/n) \leq 1\}$ . En esta expresión  $\log^{(i)} x = \log \log \dots \log x$  en  $\log i$  iterado  $i$  veces. Así,  $\beta(m, n)$  es una función que crece muy lentamente. Por ejemplo, si  $m/n = 2^{264,000}$  entonces  $\beta(m, n) = 6$



Robert Endre Tarjan

Actualmente, el algoritmo más rápido para resolver el problema del árbol de expansión mínima es el de Kruskal modificado por Tarjan (1984) si los arcos ya están ordenados.

## 5.8 EJERCICIOS PROPUESTOS

1. Sea  $G$  una gráfica no dirigida. Conteste o demuestre lo siguiente:

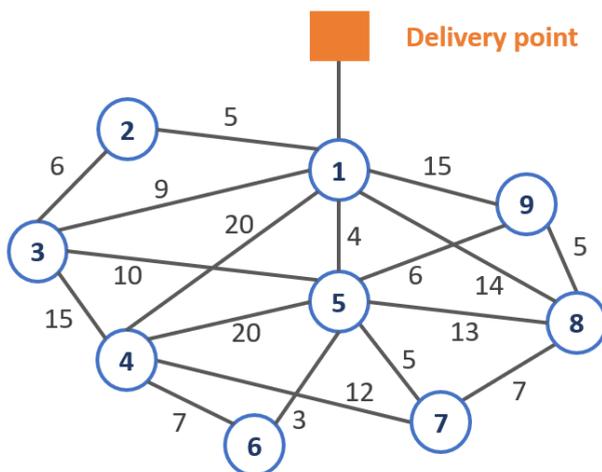
- $G$  es un árbol si...
- Si  $G$  es un árbol con  $m$  nodos ( $m \geq 2$ ) entonces  $G$  tiene al menos dos nodos terminales
- Si  $G$  es un árbol y se elimina un nodo terminal, se tiene otro árbol.

2. Sea  $G$  una gráfica no dirigida con  $m$  nodos. Demuestre que los siguientes postulados son equivalentes.

- $G$  es un árbol
- $G$  tiene  $m - 1$  arcos y no tiene circuitos.
- Si a  $G$  se le añade un arco se forma un circuito único

Sugerencia: use la fórmula de Euler

3. Encuentre el árbol de expansión mínima en la siguiente red:



4. Considere el siguiente algoritmo: arregle los arcos en  $A$  en orden arbitrario y comience con un árbol de expansión vacío. Examine cada arco en  $A$ , uno por uno, y siga los pasos siguientes:
- Aumente un arco  $(i, j)$  a  $T$ , si  $T$  contiene un circuito  $W$ .
  - Borre de  $T$  un arco de costo máximo de dicho circuito.

Demuestre que cuando este algoritmo ha examinado todos los arcos el árbol final es un árbol de expansión mínima. ¿Es posible implementar este algoritmo de manera eficiente como el de Kruskal? ¿Por qué?

# Problema de ruta más corta

## 6.1 INTRODUCCIÓN

El problema de la ruta más corta es un problema clásico de redes y de una gran aplicación en diferentes áreas no solo de la ingeniería sino de otras disciplinas, en este capítulo se presentan métodos de solución para los siguientes problemas de rutas más cortas de una red:

1. Ruta más corta entre dos nodos específicos  $s$  y  $t$
2. Rutas más cortas entre un nodo específico  $s$  y todo nodo  $i$  de la red
3. Ruta más corta entre todo par de nodos

Para ejemplificar considere lo siguiente:

### Ejemplo 6.1

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 representar en una red  $G = [N, A, d]$  donde:

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

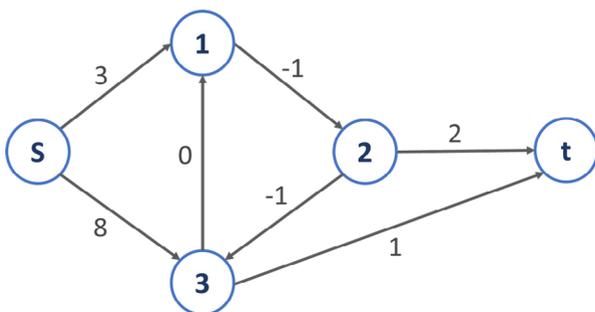
$A = \{\text{Tramos de carretera entre las ciudades}\}$

$d: A \rightarrow R$ , donde para toda  $a$  elemento de  $A$ ,  $d(a)$  es la longitud o distancia del tramo de carretera  $a$ .

En general en una red  $G = [N, A, d]$ , 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 longitudes de los arcos que la forman; aquella ruta tal que su longitud sea mínima se llama la ruta más corta o camino más corto.

El problema entonces de la terminal de autobuses es encontrar la ruta más corta entre dos nodos específicos: los que representan a las ciudades  $s$  y  $t$ . Observe que en este caso las longitudes definidas son no negativas; sin embargo, el problema de encontrar la ruta más corta entre dos nodos 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 no ser acotado puesto que para cualquier ruta entre  $s$  y  $t$  que contenga al circuito negativo existe otra mejor, a saber, aquella que contiene una vez más al circuito. Por ejemplo, en la siguiente red:



**Figura 6.1**

Una red con costos negativos

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 considera 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  $9 - 2M$ , 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 nodos específicos tenga solución, deberá cumplirse:

- i) Existe alguna trayectoria entre  $s$  y  $t$
- ii) No existen circuitos negativos tales que haya un camino de  $s$  a algún nodo del circuito y otro de algún nodo del circuito a  $t$ .

Suponga 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 les da el servicio. A este problema puede asociarse, de nuevo, la red definida anteriormente.

Antes de analizar cuándo tiene solución este problema se definirán algunos conceptos de gran utilidad.

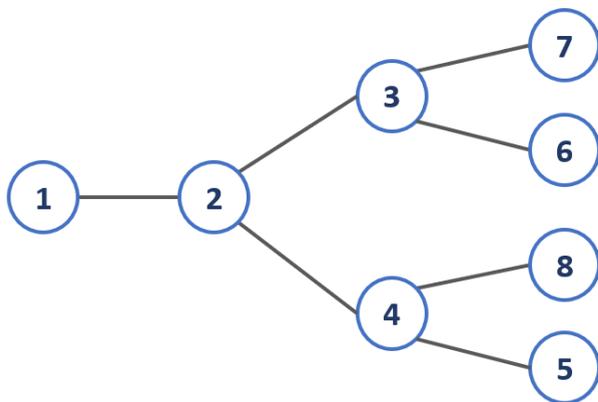
### **D** Definición 6.1

Sea  $G = [N, A]$  una red dirigida y sea  $s$  un nodo en  $N$ ; entonces a  $s$  se le llama raíz de  $G$ , si existe una trayectoria de  $s$  a  $i$  para toda  $i$  en  $N$ .

## D

**Definición 6.2**

Una *arborescencia* es un árbol que tiene una raíz. La siguiente red es una arborescencia de raíz 1:



**Figura 6.2**  
Arborescencia  
de raíz 1

## D

**Definición 6.3**

Sea  $G = [N, A]$  una gráfica dirigida. Una *arborescencia de  $G$*  es un árbol expandido de  $G$  que contiene un nodo que es raíz.

En una arborescencia de raíz  $s$  el camino o trayectoria de  $s$  a  $i$ , para toda  $i$  elemento de  $N$ , es único.

## D

**Definición 6.4**

Considere ahora una red  $G$ . Una *arborescencia de rutas más cortas de  $G$*  es aquella arborescencia tal que la única ruta de  $s$  a  $i$ , para toda  $i$  elemento de  $N$ , es una ruta más corta de  $s$  a  $i$ .

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  $G = [N, A, d]$ .

Haciendo una analogía con el problema de la ruta más corta entre dos nodos específicos puede concluirse que para que exista la arborescencia de rutas más cortas de raíz  $s$  en una red cualquiera  $G$ , ésta deberá cumplir que:

- i) Existen caminos de  $s$  a  $i$ , para toda  $i$  en  $N$ . Es decir, que  $s$  sea la raíz de la red.
- ii) No existen circuitos negativos en la red  $G$ , ya que de presentarse estos el problema sería no acotado (por la misma razón expuesta para el problema de ruta más corta entre dos nodos específicos).

Finalmente, suponga 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  $G$  definida anteriormente, se deben encontrar entonces, las rutas más cortas entre todo par de nodos en la red  $G$ .

Este último problema es una generalización inmediata de los anteriores. Por esto se deduce que, para que exista solución en cualquier red  $G$ , deberá cumplirse lo siguiente:

- i) Existe, al menos, una trayectoria entre todo par de nodos
- ii) No existen circuitos negativos en la red  $G$

Algunas propiedades de las arborescencias se dan a continuación.

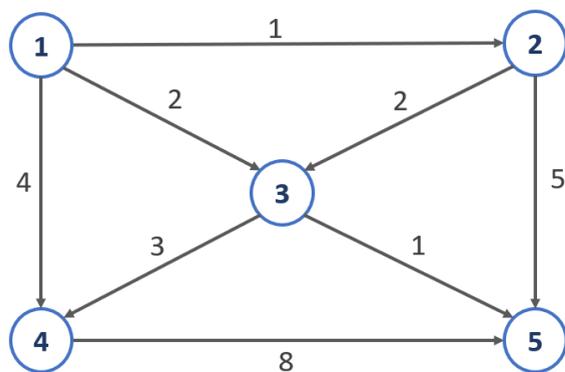
## 6.2 CARACTERIZACIÓN DE UNA ARBORESCENCIA

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 redes.

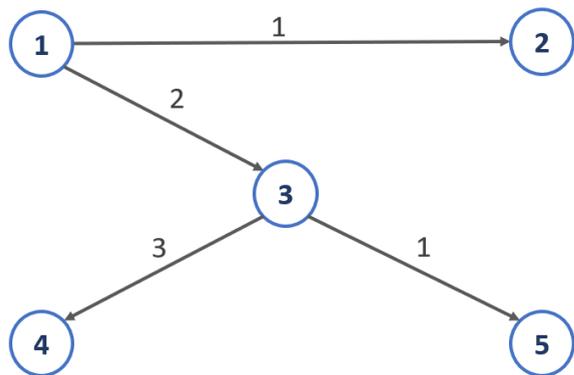
Es importante hacer notar que, en general, la arborescencia de rutas más cortas es distinta del árbol de expansión mínima en una red, puesto que, mientras el segundo concepto es no orientado el primero solo es aplicable a redes dirigidas.

Aun en el caso en que la red sea dirigida, un árbol no necesariamente es arborescencia ya que esta puede no tener raíz. Por ello, 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 esta sea de rutas más cortas. Por ejemplo, en la red de la figura 6.3.



**Figura 6.3** Red con costos positivos en arcos

Un árbol de peso mínimo está dado en la figura 6.4 el cual resulta ser una arborescencia de raíz 1. Sin embargo, la única ruta en la arborescencia entre los nodos 1 y 4 tiene una longitud de 5 unidades y existe otra ruta de longitud menor entre esos dos nodos en la red: la ruta 1, 4 tiene una longitud de 4 unidades. Por lo tanto, puede concluirse que la arborescencia no es de rutas más cortas.



**Figura 6.4**  
Árbol de  
peso mínimo

### T Teorema 6.1

Sea  $G$  una red con  $n$  nodos. Suponga que  $n \geq 2$ . Los postulados siguientes son equivalentes y caracterizan una arborescencia:

- $G$  es un árbol y tiene un nodo  $x_0$  que es raíz
- Para todo nodo  $i$  existe un camino único de  $x_0$  a  $i$
- $G$  tiene al nodo  $x_0$  que es raíz y si se elimina un arco entonces  $x_0$  ya no es raíz
- $G$  es conexa
- $G$  es acíclica
- $G$  tiene como raíz a  $x_0$  y es acíclica
- $G$  tiene como raíz a  $x_0$  y posee  $n - 1$  arcos

### D Demostración

**(a implica b).** Sea  $i$  un nodo de  $G$ . Puesto que  $x_0$  es raíz de  $G$  entonces existe una trayectoria de  $x_0$  a  $i$ . Suponga que existen, al menos, dos trayectorias de  $x_0$  a  $i$ . Entonces  $G$  tendría un ciclo y esto contradice que  $G$  es un árbol. Luego, el camino de  $x_0$  a  $i$  es único.

**(b implica c).** Puesto que existe una trayectoria entre  $x_0$  e  $i$ , para todo nodo  $i$ , entonces  $x_0$  es raíz. Sea  $a = (i_k, j_k)$  un arco de  $G$  y suponga que  $x_0$  es raíz de la red  $F' = [N, A - \{a\}]$  entonces existe una trayectoria de  $x_0$  a  $j_k$  en  $G'$  (note que esta trayectoria no contiene a  $a$ ). Puesto que  $x_0$  es raíz de  $G$ , existe una trayectoria de  $x_0$  a  $i_k$ ; esta trayectoria y el arco  $a$  forman una trayectoria de  $x_0$  a  $j_k$ ; entonces existen en  $G$  dos trayectorias distintas de  $x_0$  a  $j_k$ , 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 conexo. Suponga 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, lo cual es una contradicción por lo tanto  $g(x_0)$  es igual a 0. Por otro lado, por ser  $x_0$  raíz, se tiene que  $g(x) \geq 1$  para todo nodo  $x \neq x_0$ . Suponga que  $g(x) > 1$ , es decir, que existen al menos dos arcos  $(i_k, x)$  y  $(j_k, x)$  entonces al suprimir cualquiera de estos dos arcos  $x_0$  sigue siendo raíz de  $G$ , por lo tanto,  $g(x) = 1$ , para toda  $x \neq x_0$ .

**(d implica e).** La red  $G$  es conexa y tiene  $n - 1$  arcos puesto que  $g(x_0) = 0$  y  $g(x) = 1$  para toda  $x \neq x_0$ . 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 toda  $x \neq x_0$ .

**(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.

### 6.3 MÉTODOS DE SOLUCIÓN Y JUSTIFICACIÓN

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  $R = [N, A, d]$ . El primero de ellos solo 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 solo a estos casos. Por otro lado, el método presentado para redes con esta característica constituye el primer caso del algoritmo general posteriormente expuesto. Es importante mencionar que estos algoritmos tienen la ventaja de que, además de proporcionar la solución óptima cuando existe, la detectan cuando esta 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 nodos.

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

Otra observación importante que debe hacerse es que, como se verá posteriormente, el problema de la ruta más corta resulta ser 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 en este capítulo serán de gran utilidad.

## Arborescencia de rutas más 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 fue desarrollado por Dijkstra (1959) y está considerado como uno de los métodos más eficientes para resolver este problema.

Este método se basa en la asignación de etiquetas “permanentes” a los nodos para los cuales ya se conocen las longitudes de las rutas más cortas de la raíz a ellos. Sea  $S$  este conjunto de nodos. Las etiquetas de los nodos de  $S$  representan precisamente las longitudes de las rutas más cortas buscadas. Los nodos restantes se etiquetan “temporalmente” en una cota superior de la longitud más corta de la raíz al nodo etiquetado.

En la primera iteración el conjunto  $T$  contendrá únicamente al nodo raíz; es decir, solo la raíz estará etiquetada permanentemente. Las etiquetas temporales se mejoran continuamente y en cada iteración se agrega exactamente un nodo  $x$  a  $S$ ; este nodo es aquel tal que la longitud desde la raíz es la más corta posible.

Puesto que todos los arcos tienen costos no-negativos, siempre puede encontrarse una ruta más corta de la raíz a  $x$  que pase solo por nodos de  $S$ ; en este caso la etiqueta de  $x$  representa la longitud de la ruta más corta correspondiente. Una vez que todos los nodos estén en  $S$ , las etiquetas de todos los nodos serán 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 solo la ruta más corta entre dos nodos específicos, se obtendrá la solución cuando se etiquete “permanentemente” el nodo final del camino buscado.

1

2

3

4

5

6

7

8

## Algoritmo de Dijkstra

**Propósito:** Obtener la arborescencia de las rutas más cortas de raíz  $s$  en una red  $G = [N, A, d]$  con costos no negativos en los arcos.

### Descripción

**Paso 1** (Iniciación de etiquetas). Sea  $d(s) = 0$  y márquese 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  $x \in \Gamma^+(p)$  que tenga etiqueta temporal, actualizar etiquetas de acuerdo con:

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

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 solo 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 nodos tienen etiquetas permanentes, terminar; ésta es la longitud deseada del camino y el conjunto de arcos  $\{a(x), x\}$  forman la arborescencia de caminos más cortos.

En otro caso, ir al paso 2.

1

2

3

4

5

6

7

8

## Justificación del algoritmo

Primero observe 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 nodos 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 este.

Note que, si el algoritmo termina en el paso 3, la red generada tendrá  $n-1$  arcos y a  $s$  como nodo raíz; por esta razón dicha red 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 nodos son las longitudes de las rutas más cortas de  $s$  a  $x$ , para todo  $x \in X$ . Esto es claro en la primera iteración. Suponga también que es válido en la  $k$ -ésima iteración.

Sea  $S$  el conjunto de nodos con etiquetas permanentes y  $S'$  el conjunto de nodos 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 nodos de  $S$ . En efecto, en cada iteración solo se etiqueta permanentemente un nodo; por lo tanto, solo es necesaria la comparación efectuada en el paso 2. En particular esto sucede para  $x^*$  (nodo con la mínima etiqueta temporal).

Suponga ahora que la ruta más corta de la raíz a  $x^*$  no contiene solo nodos de  $S$ . Sea  $y$  el primer nodo, 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 porción del camino de  $s$  a  $x^*$ , que une a  $y$  con  $x^*$ , es no negativa. Sea  $D$  esta longitud. Note que la porción del camino de  $s$  a  $x^*$ , que une a  $s$

con  $y$ , es un camino que contiene solamente nodos de  $S$ . Pero  $d(y)$  es la longitud de una ruta más corta que contiene todos sus nodos 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 solo nodos 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, observe que si  $d(x^*)$  es igual a infinito (paso 2) en alguna iteración, entonces existe algún nodo ( $x^*$  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.

## E

## Ejemplo 6.2

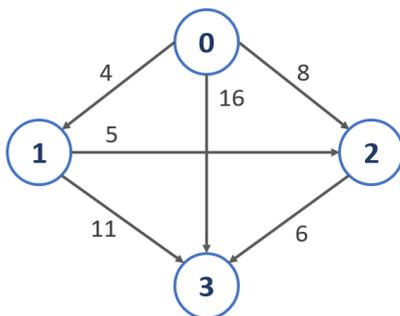
Suponga que en un aeropuerto se considera la posible adquisición de un equipo que, a causa de ciertas modificaciones, será inútil dentro de 3 años. Los costos de utilización y de reventa del equipo usado son conocidos. El equipo puede ser utilizado durante uno, dos o tres años, revenderse al final del 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 considere la red  $G = [N, A, C]$  donde  $N = \{0, 1, 2, 3\}$  y cada elemento de  $C$  indica el final de un año;  $(i, j) \in A$  si y solo 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 del año  $i$ , usarlo y revenderlo al final del año  $j$ .

Una forma esquemática de la red es:



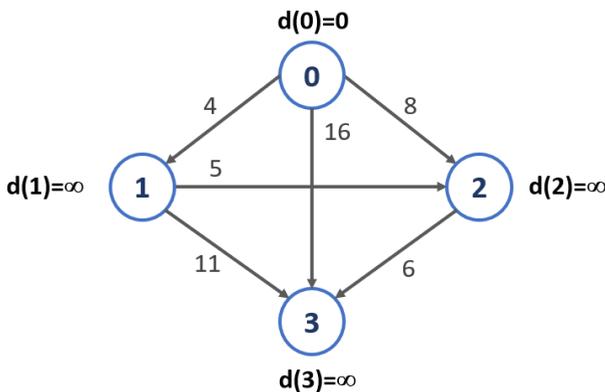
**Figura 6.5**  
Red con  
costos por  
año

Observe 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 el 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 nodos 0 y 3 de la red. Para esto se aplicará el algoritmo de Dijkstra considerando  $s = 0$  y  $t = 3$ .

**Paso 1**  $d(0) = 0$  (etiqueta permanente);  $d(x) = \infty$ , para  $x = 1, 2, 3$  (etiquetas temporales),  $p = 0$ . Ver figura 6.6.



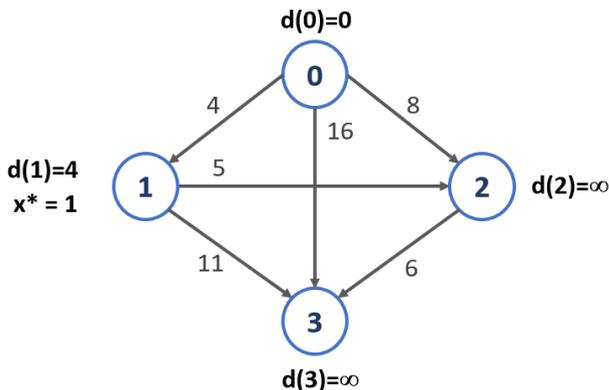
**Figura 6.6**  
Red con etiquetas

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

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

Sea  $x^*$  tal que  $d(x) = \min \{ d(1), d(2), d(3) \} = d(1)$

**Paso 3** De donde  $x^*=1$  (nodo con mínima etiqueta temporal). Se marca  $d(1)$  como permanente;  $p=1$ . ( $p \neq t=3$ ). Ver figura 6.7.



**Figura 6.7**  
Red con  
nodo 1  
etiquetado

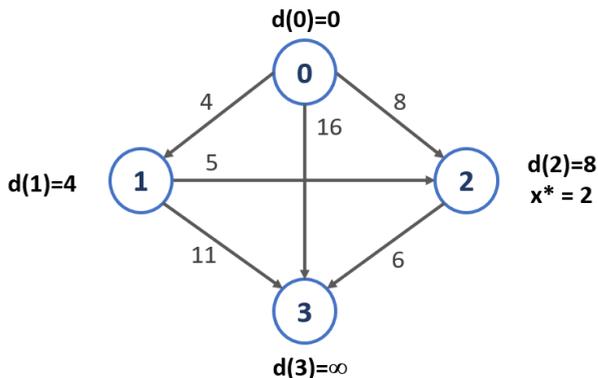
Iteración 1

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

$$d(2) = \min \{8, 4 + 5\} = 8 \quad a(2) \text{ no cambia}$$

$$d(3) = \min \{16, 4 + 11\} = 15 \quad a(3) = 1$$

De donde  $x^*=2$  (nodo con mínima etiqueta temporal). Se marca  $d(2)$  como permanente;  $p=2$ . ( $p \neq t$ ). Ver figura 6.8.



**Figura 6.8**  
Red con  
nodo 2  
etiquetado

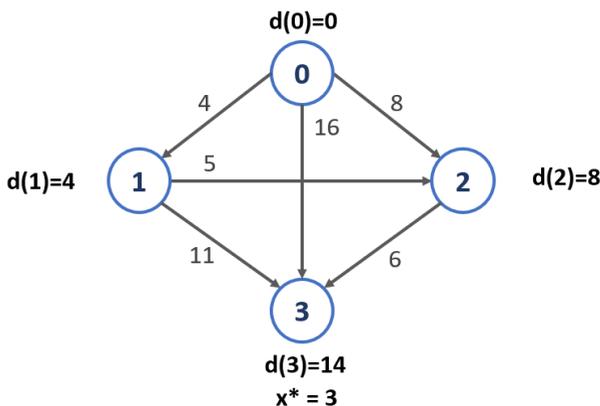
Iteración 2

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

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

$$a(3) = 2$$

Ver figura 6.9.



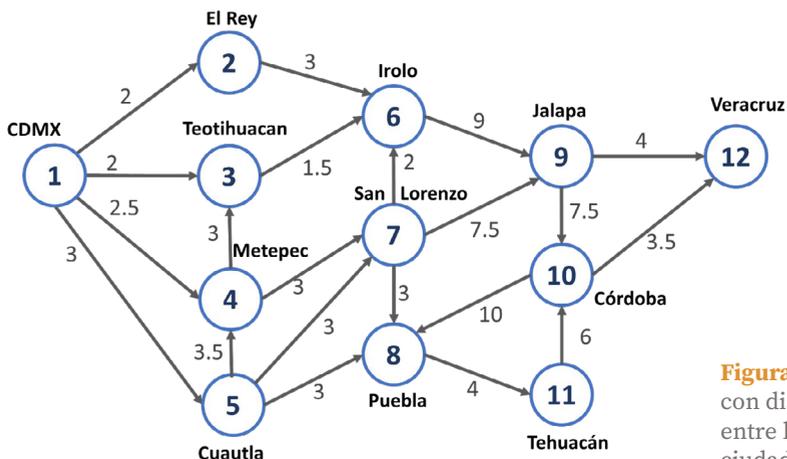
**Figura 6.9**  
Red con nodo 3 etiquetado

De donde  $x^* = 3$  (nodo 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 un 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 6.3

Considere 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  $(i, j)$ :



**Figura 6.10** Red con distancias entre las ciudades

Suponga que se desea encontrar las rutas más rápidas para los ferrocarriles que presten servicio desde la Ciudad de México (Cd. Mx.) hasta cada una de las ciudades del mapa.

En términos de redes se desea determinar la arborescencia de rutas más cortas de raíz Cd. Mx. Para resolver este problema usaremos algoritmo de Dijkstra puesto que los costos en los arcos son no negativos.

- Paso 1**  $d(\text{Cd. Mx.}) = 0$  (etiqueta permanente)  
 $d(x) = \infty$  para toda  $x$  en  $N$  tal que  $x \neq \text{Cd. Mx.}$   
 (etiquetas temporales)  
 $p = \text{Cd. Mx.}$

**Paso 2** Actualización de etiquetas:  $\Gamma^+(p) = (\text{El Rey}, \text{Teotihuacan}, \text{Metepec}, \text{Cuautla})$ :

$$\begin{array}{llll} d(\text{El Rey}) & = \min \{ \infty, 2 \} & = 2; & a(\text{El Rey}) & = \text{Cd. Mx.} \\ d(\text{Teotihuacan}) & = \min \{ \infty, 2 \} & = 2; & a(\text{Teotihuacan}) & = \text{Cd. Mx.} \\ d(\text{Metepec}) & = \min \{ \infty, 2.5 \} & = 2.5; & a(\text{Metepec}) & = \text{Cd. Mx.} \\ d(\text{Cuautla}) & = \min \{ \infty, 3 \} & = 3; & a(\text{Cuautla}) & = \text{Cd. Mx.} \end{array}$$

de donde  $x^* = \text{El Rey}$ , puesto que este es el nodo con mínima etiqueta temporal (los empates se rompen arbitrariamente). Se marca  $d(\text{El Rey})$  como permanente,  $p = \text{El Rey}$  (aún existen etiquetas temporales).

### Iteración 1

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

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

de donde  $x^* = \text{Teotihuacan}$ , puesto que este es el nodo con mínima etiqueta temporal. Se marca  $d(\text{Teotihuacan})$  como permanente,  $p = \text{Teotihuacan}$  (aún existen etiquetas temporales).

### Iteración 2

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

$$\begin{array}{l} d(\text{Irolo}) = \min \{ 5, 2 + 1.5 \} = 3.5; \\ a(\text{Irolo}) = \text{Teotihuacan} \end{array}$$

de donde  $x^* = \text{Metepec}$ , puesto que este es el nodo con mínima etiqueta temporal. Se marca  $d(\text{Metepec})$  como permanente,  $p = \text{Metepec}$  (aún existen etiquetas temporales).

Iteración 3

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

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

$$a(\text{San Lorenzo}) = \text{Metepec}$$

de donde  $x^* = \text{Cuautla}$ , puesto que este es el nodo con mínima etiqueta temporal. Se marca  $d(\text{Cuautla})$  como permanente,  $p = \text{Cuautla}$  (aún existen etiquetas temporales).

Iteración 4

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

$$d(\text{San Lorenzo}) = \min \{ 5.5, 3 + 3 \} = 5.5;$$

$$a(\text{San Lorenzo}) = \text{Metepec}$$

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

$$a(\text{Puebla}) = \text{Cuautla}$$

de donde  $x^* = \text{Irolo}$ , puesto que este es el nodo con mínima etiqueta temporal. Se marca  $d(\text{Irolo})$  como permanente.

$$p = \text{Irolo} \text{ (aún existen etiquetas temporales).}$$

Iteración 5

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

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

$$a(\text{Jalapa}) = \text{Irolo}$$

$x^*$  = San Lorenzo, puesto que este es el nodo con mínima etiqueta temporal. Se marca  $d$  (San Lorenzo) como permanente.

$p$  = San Lorenzo (aún existen etiquetas temporales).

### Iteración 6

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

$$d(\text{Jalapa}) = \min \{12.5, 5.5 + 7.5\} = 12.5;$$

$$a(\text{Jalapa}) = \text{Irolo}$$

$$d(\text{Puebla}) = \min \{6, 5.5 + 3\} = 6;$$

$$a(\text{Puebla}) \text{ no se modifica}$$

$x^*$  = Puebla, puesto que este es el nodo con mínima etiqueta temporal. Se marca  $d$ (Puebla) como permanente.

$p$  = Puebla (aún existen etiquetas temporales).

### Iteración 7

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

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

$$a(\text{Tehuacán}) = \text{Puebla}$$

$x^*$  = Tehuacán, puesto que este es el nodo con mínima etiqueta temporal. Se marca  $d$ (Tehuacán) como permanente.

$p$  = Tehuacán (aún existen etiquetas temporales).

### Iteración 8

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

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

$$a(\text{Córdoba}) = \text{Tehuacán}$$

$x^*$  = Jalapa, puesto que este es el nodo con mínima etiqueta temporal.

Se marca  $d(\text{Jalapa})$  como permanente.

$p$  = Jalapa (aún existen etiquetas temporales).

### Iteración 9

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

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

$$a(\text{Veracruz}) = \text{Jalapa}$$

$$d(\text{Córdoba}) = \min \{ 16, 12.5 + 7.5 \} = 16;$$

$$a(\text{Córdoba}) = \text{Tehuacán}$$

$x^*$  = Córdoba, puesto que este es el nodo con mínima etiqueta temporal. Se marca  $d(\text{Córdoba})$  como permanente.

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

### Iteración 10

Actualización de etiquetas:  $\Gamma^+(p) = (\text{Puebla}, \text{Veracruz})$

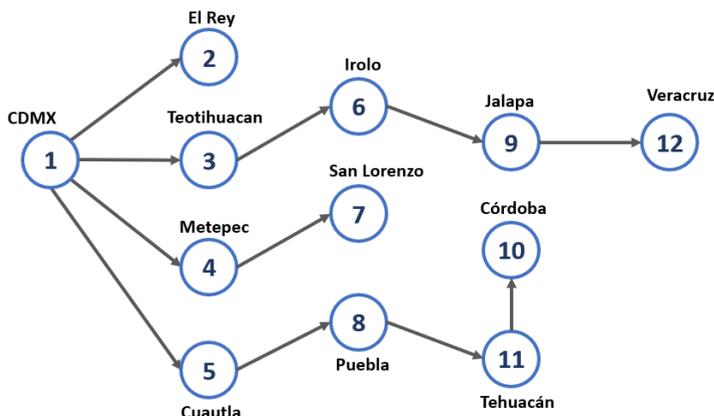
$$d(\text{Veracruz}) = \min \{ 16.5, 16 + 3.5 \} = 16.5;$$

$$a(\text{Veracruz}) = \text{Jalapa}$$

$x^*$  = Veracruz, puesto que este es el nodo con mínima etiqueta temporal. Se marca  $d(\text{Veracruz})$  como permanente.

Todos los nodos tienen etiqueta permanente, detenerse.

La arborescencia de rutas más cortas con raíz Cd.Mx. está formada por el conjunto de arcos en la siguiente red:



**Figura 6.11**  
Arborescencia de  
rutas más cortas

El único camino entre el Cd. Mx. y  $x$  para todo nodo  $x$  en  $N$ , en esta arborescencia es la ruta más rápida para seguir por ferrocarril que proporciona servicio entre las ciudades Cd. Mx. y  $x$  y su longitud es el número asociado en el nodo  $x$  (etiqueta permanente).

### Ruta más corta entre todo par de nodos

Una manera de resolver el problema de rutas más cortas entre todo par de nodos en una red  $G$  consiste en encontrar la arborescencia de rutas más cortas de raíz  $x$  para toda  $x$  elemento de  $N$ . Sin embargo, existen procedimientos más eficientes como el que se expone en esta sección. Este procedimiento fue 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 nodos de la red  $1, 2, 3, \dots, n$  y se utilizará una matriz  $C$  de dimensión  $n$  para calcular las longitudes de las rutas más cortas entre cada par de nodos; al terminar de aplicar el algoritmo, la longitud de la ruta más corta entre los nodos  $i$  y  $j$  estará 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 pueda admitir a los primeros  $k$  nodos, o a alguno de ellos, como nodos 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 en la diagonal serán igual a cero. Con esto quedan calculadas las longitudes de las rutas más cortas entre todo par de nodos  $i$  y  $j$ , que no contengan ningún nodo como nodo intermedio.

Al inicio 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  $k-1$  nodos, o a alguno de ellos, como nodos 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  $k-1$  nodos como nodos 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  $k$  nodos, o a algunos de ellos, como nodos 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  $n$  nodos como nodos intermedios o a alguno de ellos; es decir, se habrá calculado la longitud de la ruta más corta entre  $i$  y  $j$ .

Debe observarse que si, al finalizar el algoritmo, alguna entrada de  $C$  es igual a  $\infty$ , esto querrá decir que no existe ruta alguna entre los nodos correspondientes. Por otro lado, si algún elemento de la diagonal de, por ejemplo  $(i, i)$ , 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. Este algoritmo por lo anterior es importante porque detecta circuitos negativos.

## Algoritmo de Floyd

**Propósito:** Obtener las rutas más cortas entre todo par de nodos en una red  $G$  con  $n$  nodos.

### Descripción

**Paso 1** Construya la matriz  $C$  de dimensión  $n$  cuyos elementos  $c_{ij}$  son como sigue:

**Paso 2** Haga  $k = k + 1$  para toda  $i \neq k$  tal que  $c_{ij} \neq \infty$  y para todo  $j \neq k$  tal que  $c_{kj} \neq \infty$ , haga:

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

$$k = 0$$

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

- Paso 3**
- i) Si  $c_{ij} < 0$  para alguna  $i$ , termine. En este caso existe un circuito negativo que contiene al nodo  $i$  y por lo tanto no hay solución.
  - ii) Si  $c_{ij} \geq 0$ , para toda  $i$  y  $k = n$  termine  $c_{ij}$  es la longitud del camino mas corto de  $i$  a  $j$ .
  - iii) Si  $c_{ij} \geq 0$  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  $n$ , donde el elemento  $a_{ij}$  de esta matriz será el predece-

sor del nodo  $j$  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 nodos  $i, j \in N$ .

La matriz  $A$  se modificará en el paso 2 de la  $k$ -ésima iteración de acuerdo con la siguiente asignación:

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

### Justificación del algoritmo

El algoritmo de Floyd termina en exactamente  $n$  iteraciones, donde  $n$  es el número de nodos de la red, a menos que termine en el inciso i) del paso 3, en cuyo caso 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 nodo intermedio entre los nodos  $i$  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 1 y la ruta entre 1 y  $j$  ( $c_{i1} + c_{1j}$ ); de este modo se obtiene la longitud de la ruta más corta entre los nodos  $i$  y  $j$ , que no contiene ningún nodo intermedio o que contiene al nodo 1 como intermedio.

Suponga que al final de la iteración  $k-1$ ,  $c_{ij}$  representa la longitud de la ruta más corta entre  $i$  y  $j$ , que contiene los primeros  $k-1$  nodos como nodos 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  $k-1$  nodos como nodos 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  $k$  nodos 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 nodos  $i$  y  $j$ .

E

Ejemplo 6.3

Considere la siguiente red y determine las rutas más cortas entre todo par de nodos; se usará el algoritmo de Floyd para la determinación de las rutas:

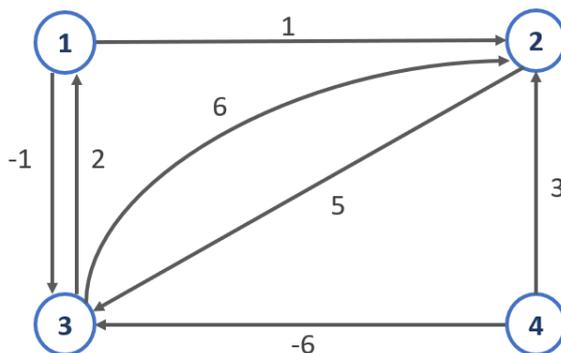


Figura 6.12 Red con distancias en los arcos

Las matrices C y A iniciales son las siguientes:

Paso 1

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

**Paso 2** Sea  $k = 1$ . Se actualizan los valores de los elementos:

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

Entonces las matrices resultantes de esta iteración son las siguientes:

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

**Paso 3**

**iii)**  $C_{ij} \geq 0 \quad \forall i, k < n$ , ir al paso 2

Iteración 1

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

$$\begin{aligned} C_{13} &= \min \{-1, 1 + 5\} = -1 && \text{no se modifica } a_{13} \\ C_{33} &= \min \{0, 3 + 5\} = 0 && \text{no se modifica } a_{33} \\ C_{43} &= \min \{-6, 3 + 5\} = -6 && \text{no se modifica } a_{43} \end{aligned}$$

Luego, las matrices  $C$  y  $A$  no se modificaron durante la segunda iteración.

Iteración 2

Se asigna  $k = 3$  y se actualizan los elementos:

$$\begin{aligned} C_{12} &= \min \{1, -1 + 3\} = 1 && \text{no se modifica } a_{12} \\ C_{21} &= \min \{\infty, 5 + 2\} = 7 && a_{21} = a_{31} = 3 \\ C_{41} &= \min \{\infty, -6 + 2\} = -4 && a_{41} = a_{31} = 3 \\ C_{42} &= \min \{3, -6 + 3\} = -3 && a_{42} = a_{32} = 3 \end{aligned}$$

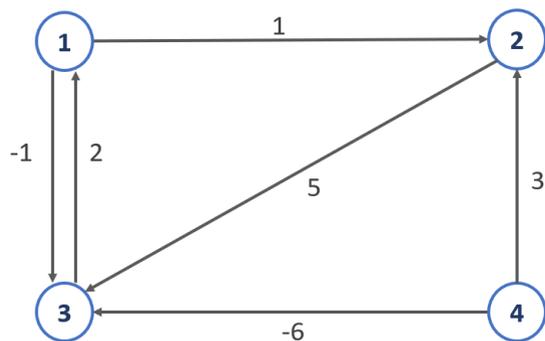
Entonces las matrices resultantes de esta iteración son las siguientes:

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

### Iteración 3

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 nodo  $j$ , en la ruta más corta de  $i$  a  $j$ , siempre y cuando  $C_{ij} \neq \infty$ .

Observe que, puesto que  $C_{i4} = \infty$  (para  $i=1,2,3$ ), entonces no existe ninguna ruta entre los nodos  $i$  y 4 (para  $i=1, 2, 3$ ); esto puede verificarse fácilmente en la red siguiente.



**Figura 6.13**  
Matriz de rutas más cortas en todo par de nodos

El Algoritmo de Floyd se puede reformular de la siguiente manera:

## Algoritmo de Floyd

PASO GENERAL K. Defina el renglón  $k$  y la columna  $k$  como renglón pivote y columna pivote. Aplique la triple operación a cada elemento  $d_{ij}$  en  $D_{k-1}$  para todo  $i$  y  $j$ . Si la condición  $d_{ik} + a_{kj} < d_{ij}$  ( $i \neq k, j \neq k, i \neq j$ ) se satisface, haga los cambios siguientes:

- a) Crear  $D_k$  reemplazando  $d_{ij}$  en  $D_{i-1}$  con  $d_{ik} + d_{kj}$
- b) Crear  $S_k$  reemplazando  $s_{ij}$  en  $S_{k-1}$  con  $k$

Esto se puede ver fácilmente como sigue: El renglón  $k$  y la columna  $k$  son la columna y el pivote actuales.

El renglón  $i$  representa cualquiera de los renglones  $1, 2, \dots, k-1$

El renglón  $p$  representa cualquiera de los renglones  $k+1, k+2, \dots, n$

La columna  $j$  representa a las columnas  $1, 2, \dots, k-1$ .

La columna  $q$  representa a las columnas  $k+1, k+2, \dots, n$ .

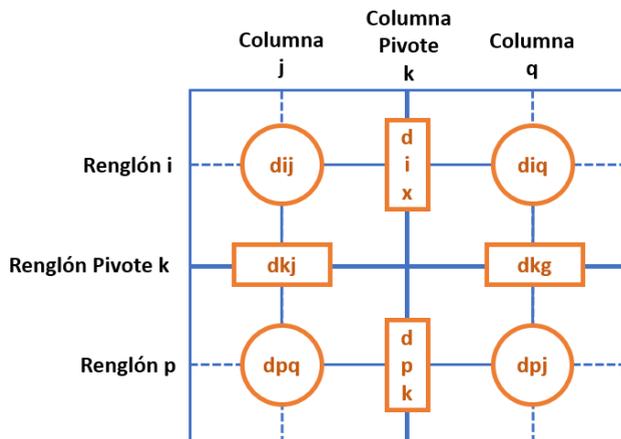
1. Si  $d_{ik} + d_{kj} < d_{ij}$ , reemplace  $d_{ij}$  con  $d_{ik} + d_{kj}$  y cambie  $s_{ij}$  por  $k$ .
2. Si  $d_{ik} + d_{kq} < d_{iq}$ , reemplace  $d_{iq}$  con  $d_{ik} + d_{kq}$  y cambie  $s_{iq}$  por  $k$ .
3. Si  $d_{pk} + d_{kj} < d_{pj}$ , reemplace  $d_{pj}$  con  $d_{pk} + d_{kj}$  y cambie  $s_{pj}$  por  $k$ .
4. Si  $d_{pk} + d_{kq} < d_{pq}$ , reemplace  $d_{pq}$  con  $d_{pk} + d_{kq}$  y cambie  $s_{pq}$  por  $k$ .

Después de  $n$  pasos, podemos determinar la ruta más corta entre los nodos  $i$  y  $j$  de las matrices  $D_n$  y  $S_n$  usando las siguientes reglas:

1. De  $D_n$ ,  $d_{ij}$  da la distancia más corta entre los nodos  $i$  y  $j$ .
2. De  $S_n$ , determinar el nodo intermedio  $k = s_{ij}$ , el cual proporciona

la ruta  $i \rightarrow k \rightarrow j$ . Si  $s_{ik} = k$ , detener; todos los nodos intermedios de la ruta han sido hallados. De otra forma, repetir el procedimiento entre los nodos  $i$  y  $k$ .

Esto se puede ver en la figura 6.14 como sigue:

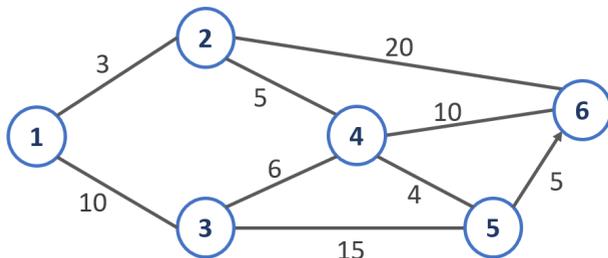


**Figura 6.14**  
Esquema general de las operaciones del algoritmo

E

**Ejemplo 6.4**

Para la siguiente red, encuentre las rutas más cortas entre cada par de nodos. Las distancias están dadas en los arcos. El arco (5, 6) es direccional, por lo que no se permite flujo del nodo 6 al nodo 5. Todos los otros arcos permiten flujos en ambas direcciones.



**Figura 6.15**  
Red con un solo arco dirigido

Las matrices  $D_0$  y  $S_0$  proporcionan la representación inicial de la red. Note que  $d_{65} = \infty$  debido a que no se permite el tráfico del nodo 6 al nodo 5. El renglón y la columna pivotes son dados por el primer renglón y la primera columna (marcados).  $k=1$ .

$$D_0 \quad \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \\ \left[ \begin{array}{cccccc} 0 & 3 & 10 & \infty & \infty & \infty \\ 3 & 0 & \infty & 5 & \infty & 20 \\ 10 & \infty & 0 & 6 & 15 & \infty \\ \infty & 5 & 6 & 0 & 4 & 10 \\ \infty & \infty & 15 & 4 & 0 & 5 \\ \infty & 20 & \infty & 10 & \infty & 0 \end{array} \right]$$

En la matriz  $D_0$ , los elementos  $d_{23}$  y  $d_{32}$  son los únicos que pueden ser mejorados. Por lo tanto, para obtener  $D_1$  y  $S_1$  de  $D_0$  y  $S_0$ :

1. Reemplace  $d_{23}$  con  $d_{21} + d_{13} = 3 + 10 = 13$  y haga  $s_{23} = 1$ .
2. Reemplace  $d_{32}$  con  $d_{31} + d_{12} = 10 + 3 = 13$  y haga  $s_{32} = 1$ .

Estos cambios se muestran en  $D_1$  y  $S_1$ .

Iteración 1

Haga  $k=2$ . Los cambios en los elementos de  $D_1$  se muestran en  $D_2$  y  $S_2$ .

$$D_1 \quad \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \\ \left[ \begin{array}{cccccc} 0 & 3 & 10 & \infty & \infty & \infty \\ 3 & 0 & \mathbf{13} & 5 & \infty & 20 \\ 10 & \mathbf{13} & 0 & 6 & 15 & \infty \\ \infty & 5 & 6 & 0 & 4 & 10 \\ \infty & \infty & 15 & 4 & 0 & 5 \\ \infty & 20 & \infty & 10 & \infty & 0 \end{array} \right]$$

Iteración 2

Fije el renglón y la columna pivotes en  $k=3$ .

$$D_2 \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \left[ \begin{array}{cccccc} 0 & 3 & 10 & \mathbf{8} & \infty & \mathbf{23} \\ 3 & 0 & 13 & 5 & \infty & 20 \\ 10 & 13 & 0 & 6 & 15 & \mathbf{33} \\ \mathbf{8} & 5 & 6 & 0 & 4 & 10 \\ \infty & \infty & 15 & 4 & 0 & 5 \\ \mathbf{23} & 20 & \mathbf{33} & 10 & \infty & 0 \end{array} \right]$$

$$S_2 \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \left[ \begin{array}{cccccc} 1 & 2 & 3 & \mathbf{2} & 5 & \mathbf{2} \\ 1 & 2 & 1 & 4 & 5 & 6 \\ 1 & 1 & 3 & 4 & 5 & \mathbf{2} \\ \mathbf{2} & 2 & 3 & 4 & 5 & 6 \\ 1 & 2 & 3 & 4 & 5 & 6 \\ \mathbf{2} & 2 & \mathbf{2} & 4 & 5 & 6 \end{array} \right]$$

Iteración 3

Fije el renglón y la columna pivotes en  $k=4$ .

$$D_3 \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \left[ \begin{array}{cccccc} 0 & 3 & 10 & 8 & \mathbf{25} & \mathbf{23} \\ 3 & 0 & 13 & 5 & \mathbf{28} & 20 \\ 10 & 13 & 0 & 6 & 15 & 33 \\ 8 & 5 & 6 & 0 & 4 & 10 \\ \mathbf{25} & \mathbf{28} & 15 & 4 & 0 & 5 \\ 23 & 20 & 33 & 10 & \mathbf{48} & 0 \end{array} \right]$$

$$S_3 \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \left[ \begin{array}{cccccc} 1 & 2 & 3 & 2 & \mathbf{3} & \mathbf{2} \\ 1 & 2 & 1 & 4 & \mathbf{3} & 6 \\ 1 & 1 & 3 & 4 & 5 & 2 \\ 2 & 2 & 3 & 4 & 5 & 6 \\ \mathbf{3} & \mathbf{3} & 3 & 4 & 5 & 6 \\ 2 & 2 & 2 & 4 & \mathbf{3} & 6 \end{array} \right]$$

Iteración 4

Fije el renglón y la columna pivotes en  $k=5$ .

$$D_4 \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \left[ \begin{array}{cccccc} 0 & 3 & 10 & 8 & \mathbf{12} & \mathbf{18} \\ 3 & 0 & \mathbf{11} & 5 & \mathbf{9} & \mathbf{15} \\ 10 & \mathbf{11} & 0 & 6 & \mathbf{10} & \mathbf{16} \\ 8 & 5 & 6 & 0 & 4 & 10 \\ \mathbf{12} & \mathbf{9} & \mathbf{10} & 4 & 0 & 5 \\ \mathbf{18} & \mathbf{15} & \mathbf{16} & 10 & \mathbf{14} & 0 \end{array} \right]$$

$$S_4 \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \left[ \begin{array}{cccccc} 1 & 2 & 3 & 2 & \mathbf{4} & \mathbf{4} \\ 1 & 2 & 3 & 2 & \mathbf{4} & \mathbf{4} \\ 1 & \mathbf{4} & 3 & 4 & \mathbf{4} & \mathbf{4} \\ 2 & 2 & 3 & 4 & 5 & 6 \\ \mathbf{4} & \mathbf{4} & \mathbf{4} & 4 & 5 & 6 \\ \mathbf{4} & \mathbf{4} & \mathbf{4} & 4 & \mathbf{4} & 6 \end{array} \right]$$

Iteración 5

Fije el renglón y la columna pivotes en  $k=5$ . Como muestra la matriz  $D_5$ , ya no es posible mejorar ninguna de las rutas. Por lo tanto,  $D_5$  y  $S_5$  son las matrices óptimas para la red.

$$D_5 \quad \begin{matrix} & 1 & 2 & 3 & 4 & 5 & 6 \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix} & \begin{bmatrix} 0 & 3 & 10 & 8 & 12 & 17 \\ 3 & 0 & 11 & 5 & 9 & 14 \\ 10 & 11 & 0 & 6 & 10 & 15 \\ 8 & 5 & 6 & 0 & 4 & 9 \\ 12 & 9 & 10 & 4 & 0 & 5 \\ 18 & 15 & 16 & 10 & 14 & 0 \end{bmatrix} \end{matrix}$$

$$S_5 \quad \begin{matrix} & 1 & 2 & 3 & 4 & 5 & 6 \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix} & \begin{bmatrix} 1 & 2 & 3 & 2 & 4 & 5 \\ 1 & 2 & 4 & 4 & 4 & 5 \\ 1 & 4 & 3 & 4 & 4 & 4 \\ 2 & 2 & 3 & 4 & 5 & 6 \\ 4 & 4 & 4 & 4 & 5 & 4 \\ 5 & 5 & 5 & 5 & 4 & 6 \end{bmatrix} \end{matrix}$$

Estas matrices contienen toda la información necesaria para determinar la ruta más corta entre todo par de nodos en la red. Por ejemplo, la distancia más corta entre el nodo 1 y el nodo 6 es  $d_{16} = 17$ . Para determinar la ruta, sea cada segmento  $(i, j)$  una ruta directa si  $s_{ij} = j$ , de otra manera  $i$  y  $j$  están ligadas por al menos otro nodo intermedio.

Como  $s_{16} = 5$  y  $s_{56} = 6$ , la ruta está dada inicialmente por  $1 \rightarrow 5 \rightarrow 6$ . Como  $s_{15} \neq 5$ , el segmento  $(1, 5)$  no es una ruta directa y necesitamos determinar sus nodos intermedios; dados  $s_{15} = 4$  y  $s_{45} = 5$ , la ruta  $1 \rightarrow 5$  es reemplazada por  $1 \rightarrow 4 \rightarrow 5$  y como  $s_{14} = 2$  y  $s_{24} = 4$ ,  $1 \rightarrow 4$  es reemplazada por  $1 \rightarrow 2 \rightarrow 4$ . Finalmente, tenemos  $s_{12} = 2$  y no existen más nodos intermedios, lo que implica que  $1 \rightarrow 2 \rightarrow 4 \rightarrow 5 \rightarrow 6$  es la ruta óptima y la distancia asociada a la ruta es 17.

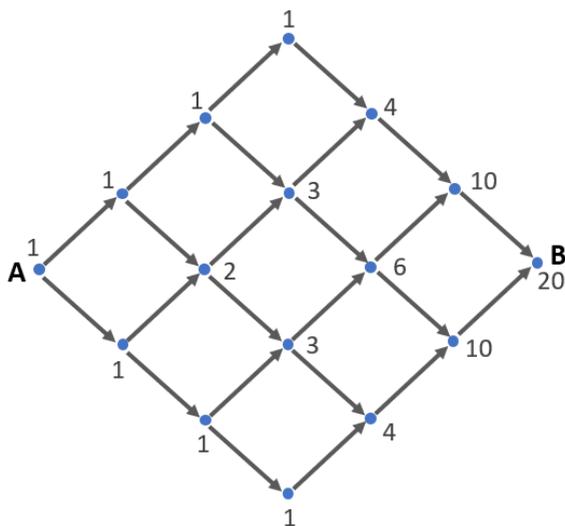
Otra manera de resolver el problema de ruta más corta es usando programación dinámica como se muestra en la siguiente sección.



Todas las calles son de un solo sentido y los números representan el esfuerzo (que usualmente es tiempo, pero algunas veces costo o distancia) requerido para atravesar cada bloque o cuadra individual. Si usted vive en *A* y quiere llegar a *B* con el mínimo esfuerzo total. Se puede resolver este problema enumerando todas las trayectorias posibles de *A* a *B*. Sumando los esfuerzos bloque por bloque para cada trayectoria y positivamente escogiendo la menor suma.

Sin embargo, hay 20 trayectorias posibles de *A* a *B* (¿Por qué?).

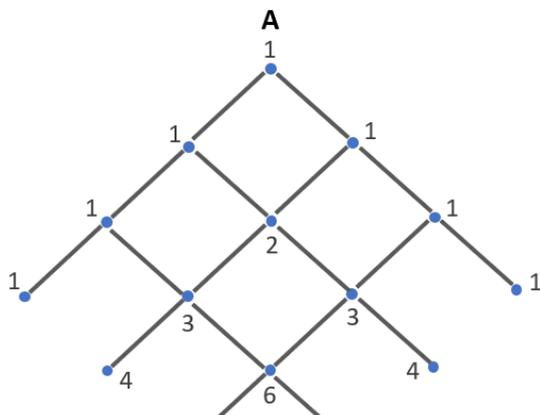
Si enumeramos las calles de la siguiente manera:



**Figura 6.17**  
Enumeración  
de calles

Se pone un 1 en *A* pues para llegar ahí solo hay una posibilidad (de hecho, se va a salir de ahí). Después se pone un 1 en cada esquina inmediata porque solo hay una forma de ir a ellas desde *A*, dos en la siguiente esquina pues hay dos formas de llegar a ella y así en cada esquina se pone la suma de los dos números más cercanos situados sobre el o los caminos que conduzcan hasta ella.

Si cortamos a la mitad esta figura y la rotamos:



**Figura 6.18**  
La red cortada  
y rotada

Los números de este triángulo son los números del triángulo de Pascal; entonces, el algoritmo para contar el número de trayectorias mínimas que bajan desde la cúspide no es otro que el método usual para construir el triángulo de Pascal.

Entonces para una red de  $n \times m$  el número de trayectorias posibles de  $A$  a  $B$  es igual al número de combinaciones de  $n + m$  elementos tomados de  $n$  en  $n$ .

$$\frac{n + m}{n}$$

Si  $n = m$  se tiene:

$$\frac{(2n)!}{n! n!} = \frac{(2(3))!}{3! 3!} = \frac{6!}{3! 3!} = \frac{6 \times 5 \times 4}{3 \times 2} = \frac{120}{6} = 20$$

Si se toma una red de  $30 \times 30$  son 2000 años.

Se tiene entonces 20 caminos de  $A$  a  $B$  y 5 sumas que producen la suma de 6 números a lo largo de la trayectoria, así 100 sumas podrían producir las 20 sumas de las trayectorias a ser comparadas. Ya que una comparación produce el menor de 2 números, una comparación adicional (de ese número con un tercero) produce el menor de 3, y así sucesivamente. Las enumeraciones completan esta solución del problema.

Sin embargo, se puede resolver este problema usando programación dinámica.

### La solución de programación dinámica

Para desarrollar el problema en términos de programación dinámica, razonamos de la siguiente forma: “No sé si voy a avanzar diagonalmente hacia arriba o diagonalmente hacia abajo desde  $A$ , pero si sólo conozco 2 números adicionales, —es decir el esfuerzo total requerido para ir de  $C$  a  $B$  por la mejor trayectoria (el mínimo esfuerzo) y el esfuerzo requerido de ir de  $D$  a  $B$  por la mejor trayectoria— puedo entonces hacer la mejor selección desde  $A$ ”.

Denotamos el mínimo esfuerzo desde  $C$  a  $B$  por  $S_C$  y el mínimo esfuerzo desde  $D$  a  $B$  por  $S_D$ , se puede entonces aumentar el esfuerzo de ir de  $A$  a  $C$  a  $S_C$ , obteniendo así el esfuerzo requerido de la mejor trayectoria de ir de  $A$  a  $B$  usando  $C$ . Lo mismo se hace con  $S_D$  y finalmente se comparan estas 2 sumas para encontrar el mínimo esfuerzo y la mejor decisión.

Por supuesto, todo esto presupone el conocimiento de los números  $S_C$  y  $S_D$ . Sin embargo, una de las ideas centrales de la programación dinámica se ha establecido ya que solo el esfuerzo a través de las mejores

trayectorias desde  $C$  y desde  $D$  hacia  $B$  son relevantes para los cálculos y el esfuerzo a lo largo de las otras 9 trayectorias respectivas desde  $C$  y  $D$  a  $B$  nunca necesitan calcularse. Esta observación a menudo se conoce como el principio de optimalidad y se establece como sigue:

### Principio de Optimalidad

La mejor trayectoria de  $A$  a  $B$  tiene la propiedad de que cualquiera que sea la decisión inicial en  $A$ , la trayectoria remanente a  $B$ , comenzando desde el siguiente punto después de  $A$ , debe ser la mejor trayectoria desde ese punto a  $B$ .

Sea  $S_C$  y  $S_D$  como ya se había mencionado, podemos citar el principio de optimalidad para justificar la fórmula:

$$S_A = \min \begin{cases} 1 + S_C \\ 0 + S_D \end{cases}$$

Donde  $S_A$  es el mínimo esfuerzo de ir de  $A$  a  $B$  y el símbolo “min [ ]” significa “el menor entre  $x$  e  $y$ ”.

La segunda idea importante es: Mientras los dos números  $S_C$  y  $S_D$  son desconocidos para nosotros, inicialmente podemos calcular  $S_C$  si conocemos  $S_E$  y  $S_F$  (los esfuerzos mínimos desde  $E$  y  $F$  para llegar a  $B$  respectivamente) y remitiéndonos al principio de optimalidad escribimos:

$$S_C = \min \begin{cases} 5 + S_E \\ 4 + S_F \end{cases}$$

Así mismo:

$$S_D = \min \begin{cases} 7 + S_F \\ 4 + S_G \end{cases}$$

Así continuamos con  $S_E$ ,  $S_F$  y  $S_G$  si no los conocemos, los podemos calcular si  $S_H$ ,  $S_I$ ,  $S_J$  y  $S_K$  se conocen y continuamos con este razonamiento hasta llegar a  $O$  y  $P$ , trabajando hacia atrás desde  $O$  y  $P$  hasta  $A$  tenemos:

$$S_E = \min \begin{cases} 2 + S_H = 2 + 10 \\ 1 + S_I = 1 + 8 \end{cases} = 9$$

$$S_F = \min \begin{cases} 1 + S_I = 1 + 8 \\ 2 + S_J = 2 + 6 \end{cases} = 8$$

$$S_G = \min \begin{cases} 5 + S_J = 5 + 6 \\ 4 + S_K = 4 + 7 \end{cases} = 11$$

$$S_C = \min \begin{cases} 5 + S_E = 5 + 9 \\ 4 + S_F = 4 + 8 \end{cases} = 12$$

$$S_D = \min \begin{cases} 7 + S_F = 7 + 8 \\ 3 + S_G = 3 + 11 \end{cases} = 12$$

$$S_A = \min \begin{cases} 1 + S_C = 1 + 12 \\ 0 + S_D = 0 + 14 \end{cases} = 13$$

$$S_L = 5 + S_O = 7$$

$$S_H = 3 + S_L = 10$$

$$S_M = \min \begin{cases} 2 + S_O = 2 + 2 \\ 8 + S_P = 8 + 1 \end{cases} = 4$$

$$S_N = 4 + S_P = 5$$

$$S_I = \min \begin{cases} 3 + S_L = 3 + 7 \\ 4 + S_M = 4 + 4 \end{cases} = 8$$

$$S_J = \min \begin{cases} 2 + S_M = 2 + 4 \\ 2 + S_N = 2 + 5 \end{cases} = 6$$

$$S_K = 2 + S_N = 2 + 5 = 7$$

Como la mejor trayectoria de  $S_A = 13$  desarrollamos una suma en cada punto  $H$ ,  $L$ ,  $O$ ,  $K$ ,  $N$  y  $P$  (que son 6 puntos) donde solo fue posible una decisión y desarrollamos 2 sumas y una comparación por cada uno de los 9 puntos remanentes donde las 2 decisiones iniciales fueron posibles. Esto da 24 sumas y 9 comparaciones, lo cual contrasta con 100 sumas y 19 comparaciones usando enumeración exhaustiva (o fuerza bruta).

Una vez que conocemos el esfuerzo total queremos saber cuál es la trayectoria, esta se obtiene fácilmente fijándonos en las 2 primeras decisiones y buscando el mínimo en nuestro cálculo previo.

Si denotamos  $x$  un nodo inicial cualquiera y  $Px$  el nodo posterior al nodo  $x$  en la trayectoria óptima desde  $x$  a  $B$  entonces se tiene:

$$\begin{array}{ll} P_M = O & \text{ya que } 2 + S_O \text{ fue menor que } 8 + S_P \\ P_I = M & \text{ya que } 4 + S_M < 3 + S_L \text{ y así se continua.} \end{array}$$

Así, se tiene que:

$$\begin{array}{llll}
 P_O = B & P_P = B & & \\
 P_L = O & P_M = O & P_N = P & \\
 P_H = L & P_I = M & P_J = M & P_K = N \\
 P_R = I & P_F = J & P_G = J \text{ o } K & \\
 P_C = F & P_D = G & & \\
 P_A = C & & & 
 \end{array}$$

La mejor trayectoria es:

$$A \rightarrow C \rightarrow F \rightarrow J \rightarrow M \rightarrow O \rightarrow B$$

Sumando las distancias a lo largo de la trayectoria se tiene

$$1 + 4 + 2 + 2 + 2 + 2 = 13$$

que es igual a  $S_A$ .

## Terminología

La regla que asigna los valores a varios subproblemas es la función de valor óptimo (en este caso  $S$ ).

El subíndice de  $S$  (por ejemplo,  $S_A$ ) es el argumento de la función  $S$  y cada argumento se refiere a un subproblema en particular (por la definición de  $S$ , en este caso  $A$  significa la mejor trayectoria de  $A$  a  $B$  que es la deseada).

La regla que asocia la primera mejor decisión con cada subproblema (la función  $P$ ) es la función de política óptima.

El principio de optimalidad produce una fórmula o un conjunto de fórmulas relativas a varios valores de  $S$ . Esta fórmula es la relación de recurrencia. Finalmente, el valor de la función de valor óptimo  $S$  para ciertos argumentos se supone obvia del tipo de problema y de la definición de  $S$  que no requieren cálculos. Estos valores obvios se conocen como condiciones de frontera a  $S$ .

En este caso: Sea

$S_x$  = trayectoria más corta de  $x$  a  $B$

$X = A, C, D, \dots$

$S_B = 0$  (Condición de frontera)

Ecuaciones recursivas.

$C_{xy}$  = longitud del nodo  $x$  al  $y$

$C_{xz}$  = longitud del nodo  $x$  al  $z$

Además, se tiene la longitud mínima de todo nodo al nodo  $B$ .

## 6.5 COMPLEJIDAD COMPUTACIONAL DE LOS ALGORITMOS

### *Tiempo de ejecución del algoritmo de Dijkstra*

Ahora estudiamos la complejidad del peor caso del algoritmo de Dijkstra. Podríamos ver el tiempo computacional para el algoritmo de Dijkstra asignado a las siguientes dos operaciones básicas:

1. *Selección de nodos.* El algoritmo realiza esta operación  $n$  veces y cada operación de este tipo requiere que se escaneé cada nodo etique-

tado temporalmente. Por lo tanto, el tiempo total de selección de nodos es  $n + (n-1) + (n-2) + \dots + 1 = O(n^2)$ .

2. *Actualizaciones de distancia.* El algoritmo realiza esta operación  $|A(i)|$  veces para el nodo  $i$ . En general, el algoritmo realiza esta operación  $\sum_{i \in N} |A(i)| = m$  veces. Dado que cada operación de actualización de distancia requiere  $O(1)$  tiempo, el algoritmo requiere  $O(m)$  de tiempo total para actualizar todas las etiquetas de distancia.

Esto se resume en el siguiente teorema:

### **T** Teorema 6.2

El algoritmo de Dijkstra resuelve el problema de la ruta más corta en un tiempo  $O(n^2)$ .

El límite de tiempo  $O(n^2)$  para el algoritmo de Dijkstra es el mejor posible para redes completamente densas [es decir,  $m = \Omega(n^2)$ ], pero se puede mejorar para redes dispersas. Tenga en cuenta que los tiempos requeridos por la selección de nodos y las actualizaciones de distancia no están equilibrados.

La selección de nodos requiere un tiempo total de  $O(n^2)$ , y las actualizaciones de distancia requieren solo un tiempo de  $O(m)$ . Los investigadores han intentado reducir el tiempo de selección de nodos sin aumentar sustancialmente el tiempo de actualización de las distancias. En consecuencia, utilizando estructuras de datos inteligentes, han sugerido varias implementaciones del algoritmo. Estas implementaciones han reducido drásticamente el tiempo de ejecución del algoritmo en la práctica o han mejorado su complejidad en el peor de los casos, para más información consultar Ahuja, Magnanti, Orlin (1993).

El algoritmo de Floyd-Warshall claramente realiza  $n$  iteraciones principales, una para cada  $k$ , y dentro de cada iteración principal realiza  $O(1)$  cálculos para cada par de nodos. En consecuencia, se ejecuta en un tiempo  $O(n^3)$ . Hemos establecido así el siguiente resultado.

### **T** Teorema 6.3

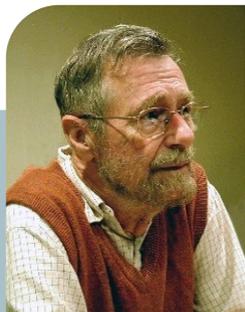
El algoritmo de Floyd-Warshall calcula las rutas más cortas entre todo par de nodos en un tiempo  $O(n^3)$ .

El algoritmo genérico de corrección de etiquetas de todos los pares y su implementación específica como el algoritmo de Floyd-Warshall son algoritmos de manipulación de matrices. Mantienen una matriz de distancias tentativas de ruta más corta entre todos los pares de nodos y realizan actualizaciones repetidas de esta matriz. Los principales inconvenientes de este enfoque son sus importantes requisitos de almacenamiento y su pobre complejidad en el peor de los casos dada la densidad de la red, excepto en los casos donde las redes son completamente densas. Los algoritmos de manipulación de matrices requieren  $\Omega(n^2)$  de espacio de almacenamiento intermedio, lo que podría hacer prohibitiva su aplicación en algunas situaciones. A pesar de estas desventajas, los algoritmos de manipulación de matrices han demostrado ser métodos computacionales populares para resolver problemas de ruta más corta entre todo par de nodos.

## 6.6 NOTAS HISTÓRICAS

El problema de la ruta más corta y sus generalizaciones tienen una voluminosa literatura. Como una guía al respecto hasta antes de 1984, remitimos al lector a la extensa bibliografía compilada por Deo y Pang

(1984). En este escrito se presentan algunas referencias seleccionadas; referencias adicionales pueden encontrarse en los artículos de Ahuja, Magnanti y Orlin (1989, 1991).



Edsger Dijkstra

El primer algoritmo de etiquetado fue sugerido por Dijkstra (1959) e, independientemente, por Dantzig (1960) y Whiting y Hillier (1960). La instrumentación original del algoritmo de Dijkstra corre en un tiempo de  $O(n^2)$ , el cual es el tiempo óptimo de corrida para redes completamente densas [aquellas con  $m = \Omega(n^2)$ ] debido a que cualquier algoritmo debe examinar todos los arcos.

Además, el uso de apilamientos nos permite obtener mejores tiempos de corrida para redes esparcidas. El uso del apilamiento- $d$  del algoritmo de Dijkstra con  $d = \max\{2, \lceil m/n \rceil\}$  corre en un tiempo de  $O(m \log_d n)$  y fue hecho por Johnson (1977a). El uso del apilamiento de Fibonacci hecho por Fredman y Tarjan (1984) corre en un tiempo de  $O(m + n \log n)$ . Johnson (1982) sugirió el uso del algoritmo de Dijkstra  $O(m \log \log C)$  basado en un trabajo anterior de Boas, Kaas y Sijlstra (1977). El algoritmo escalable de Gabow (1985) es otro eficiente algoritmo de ruta más corta.



Stephen Warshall

El algoritmo de Floyd-Warshall, publicado por Floyd (1962), se basó en el algoritmo de Warshall (1962).



Robert W. Floyd

1

2

3

4

5

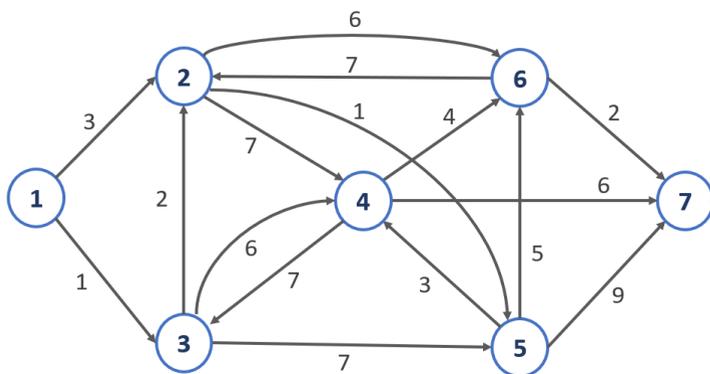
6

7

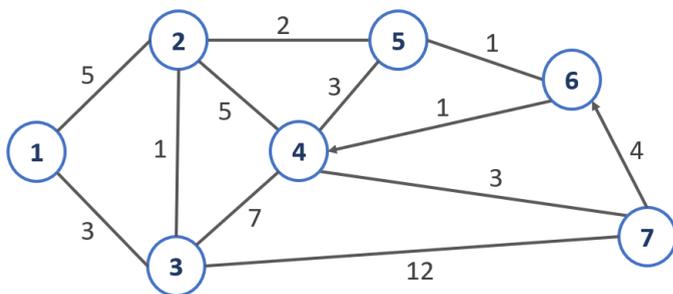
8

## 6.7 EJERCICIOS PROPUESTOS

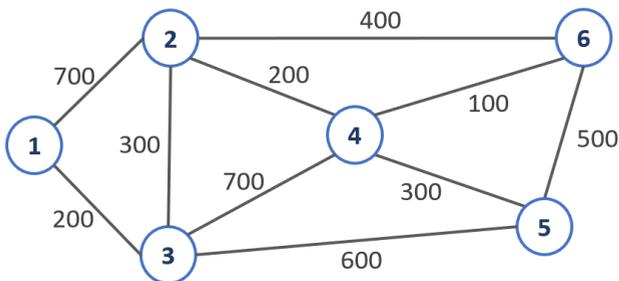
- Usando el algoritmo de Dijkstra encuentre las rutas más cortas entre el nodo 1 y todos los demás nodos en la siguiente red:



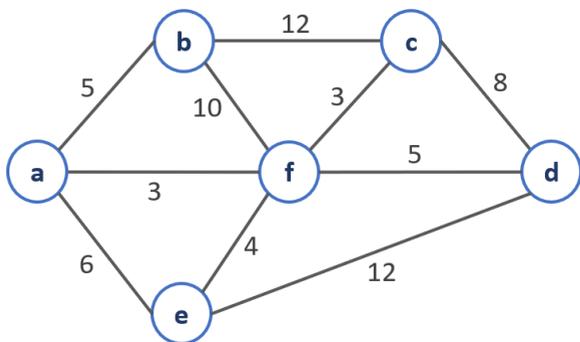
- Aplicando el algoritmo de Floyd encuentre la ruta más corta de:
  - el nodo 1 al nodo 7
  - del nodo 7 al nodo 1
  - del nodo 6 al nodo 7



- La compañía Telcel tiene seis áreas geográficas. Las distancias satelitales en kilómetros se dan en la red siguiente. Telcel necesita determinar las rutas más eficientes de mensajes que se deben establecer entre cada par de áreas en la red.



4. Suponga que un autobús debe recorrer cada uno de los 6 poblados indicados a continuación, visitando cada poblado una vez y regresando al punto de partida. El tiempo de recorrido en este caso es simétrico es decir  $t_{ij} = t_{ji}$ . ¿Qué secuencia de visita se debe llevar a cabo con objeto de minimizar el tiempo de recorrido total? Este problema se conoce como el “problema del agente viajero”, investigue métodos de solución y resuelva.



# Ruta crítica (Pert-CPM)

## 7.1 INTRODUCCIÓN

Los métodos CPM (método de la ruta crítica o del camino crítico, *Critical Path Method*) y PERT (técnica de evaluación y revisión de programa, *Program Evaluation and Review Technique*) se basan en redes, y tienen por objetivo auxiliar a la planeación, programación y control de proyectos. Se define un proyecto como un conjunto de actividades interrelacionadas, en la que cada actividad consume tiempo y recursos. El objetivo del CPM y del PERT es contar con un método analítico para programar las actividades.

El método PERT es una técnica estadística que analiza los tiempos de las tareas de un determinado proyecto.

Esta técnica emplea una distribución de probabilidad con base en tres tiempos a cada actividad (tiempo optimista, medio y pesimista). Se utiliza mayormente en proyectos de investigación donde no se tiene data histórica de referencia para definir los tiempos de cada tarea.

La técnica PERT fue desarrollada en la década de 1950 por la Marina de los Estados Unidos.<sup>1</sup>

El método CPM (por sus siglas en inglés *Critical Path Method*), también conocido como método de la ruta crítica, surgió en 1957, implementado como una herramienta por J. E. Kelly de Remington Rand y M. R. Walker de DuPont para ayudar en la construcción y el mantenimiento de las plantas químicas de DuPont.

La técnica CPM parte de la suposición de que los tiempos de las actividades son fijos. Este tiempo lo denomina tiempo normal o estándar que se utiliza para realizar los cálculos. Este tiempo viene asociado con el costo normal de la actividad. Además del tiempo normal, considera el tiempo de aceleración, que se define como la duración más corta necesaria para terminar la actividad. Este tiempo de aceleración se encuentra asociado con el costo de aceleración de la actividad.

Estas asignaciones de tiempos y costos normales y acelerados están relacionadas con acciones que podemos tomar en la gestión de un proyecto como añadir más personal, equipos, entre otros. Al asignar más recursos, aumentarán los costos.

Las dos técnicas, CPM y PERT, que se desarrollaron en forma independiente, difieren en que en el CPM se suponen duraciones determinísticas de actividad, mientras que en PERT se suponen duraciones probabilísticas.

---

<sup>1</sup> <https://www.plandemejora.com/metodo-pert-cpm-en-la-gestion-de-proyectos/>

## 7.2 DIFERENCIAS ENTRE PERT Y CPM<sup>2</sup>

En la tabla 7.1 siguiente se muestran las principales diferencias entre ambos métodos:

**Tabla 7.1** Diferencias entre los métodos PERT y CPM

PERT	CPM
Se usa para proyectos con tiempos inciertos.	Se usa para proyectos con tiempos conocidos.
Es un modelo estocástico.	Es un modelo determinista.
Está orientado a la gestión efectiva del tiempo del proyecto.	Está orientado a la gestión efectiva de los costos.
Es adecuado para proyectos de investigación y desarrollo.	Es adecuado para proyectos de construcción o proyectos donde se tienen los antecedentes documentados.
Se usa cuando la naturaleza del trabajo no es repetitiva.	Se usa cuando implica trabajo de naturaleza repetitiva.

Como vimos en los puntos anteriores, los métodos PERT y CPM tienen algunas diferencias entre los enfoques en que abordan los proyectos; sin embargo, tienen los mismos objetivos. De la misma forma, el análisis empleado por ambas técnicas es muy similar, por ejemplo, ambos utilizan un diagrama de redes para representar las actividades.

En la práctica, tanto PERT y CPM se complementan y pueden ser utilizados en conjunto. Asimismo, cada vez es más frecuente que en la bibliografía se haga referencia a ambos métodos como uno solo denominado método PERT CPM.

<sup>2</sup> <https://www.plandemejora.com/metodo-pert-cpm-en-la-gestion-de-proyectos/>

De esta forma, con la aplicación del método PERT CPM podemos responder las siguientes interrogantes:

¿Cuál es la duración del proyecto?

¿Cuáles son las actividades críticas del proyecto?

¿Cuáles son las actividades no críticas?

¿Cuál es la probabilidad de terminar el proyecto en una fecha específica?

Para una fecha en particular, ¿el proyecto está a tiempo, retrasado o adelantado con respecto al programa?

Para una fecha dada, ¿el dinero gastado es igual, menor, o mayor que la cantidad presupuestada?

¿Se dispone de suficientes recursos para terminar el proyecto a tiempo?

Si el proyecto debe terminar en menor tiempo,

¿cuál es la mejor manera de lograr esta meta al menor costo?

## Ventajas

El método PERT CPM tiene las siguientes ventajas:

1. Es muy útil en el control y la programación de grandes proyectos.
2. Tiene conceptos claros y su complejidad matemática es mínima.
3. Los diagramas de redes PERT CPM facilitan la visualización de las relaciones que hay entre las actividades del proyecto.
4. El análisis del camino crítico y los tiempos de holgura nos ayudan a detectar las actividades prioritarias.
5. Las gráficas y la documentación del proyecto indican quién es el responsable de las diferentes tareas. (En este punto es muy útil el diagrama de Gantt).

1

2

3

4

5

6

7

8

6. Su alcance abarca diferentes tipos de proyectos en varios sectores.
7. Permite supervisar tanto los tiempos como los costos.

### Limitaciones

Como toda herramienta, el método PERT CPM tiene ciertas limitaciones, entre ellas:

1. Se requiere definir claramente las actividades del proyecto. Eso implica independizarlas y definir adecuadamente sus relaciones, lo que en la práctica puede ser complicado.
2. Las relaciones de precedencia entre las tareas deben especificarse e incorporarse a la red de forma conjunta.
3. El proceso para estimar los tiempos tiende a no ser objetivo; por el contrario, puede fallar debido a gerentes que temen ser demasiado optimistas o no lo suficientemente pesimistas.
4. Existe un riesgo inherente de darle demasiada importancia a la ruta crítica. Como administradores de los proyectos, se debe saber que las actividades no críticas pueden convertirse en críticas si no se manejan adecuadamente.

### 7.3 TERMINOLOGÍA DE PERT/CPM

La primera etapa del proceso de PERT/CPM consiste en identificar todas las tareas o actividades asociadas con el proyecto y sus interrelaciones. Por ejemplo, se considera un caso básico de un proyecto de ajuste general de un motor, como se muestra a continuación.

## Ejemplo 7.1

En la tabla siguiente se listan las actividades, así como las relaciones de precedencia para el ajuste de un motor.

**Tabla 7.2** Actividades y relaciones de precedencia para ajuste de motor

Código de Actividad	Descripción de la actividad	Predecesores inmediatos
A	Quitar y desarmar el motor	-
B	Limpiar y pintar la base	A
C	Rebobinar la armadura	A
D	Reemplazar los anillos	A
E	Ensamblar e instalar el motor en la base	B, C, D

Para este ejemplo solo se requieren cinco actividades, pero el número de actividades varía de acuerdo con cada proyecto. El ajuste de un motor puede requerir unas cuantas actividades, mientras que la construcción de una planta de energía o una refinería de petróleo podría implicar varios cientos. En cualquier caso, el punto clave es tener, en esta etapa de planeación una lista precisa y exhaustiva de actividades y las relaciones correctas de precedencia entre ellas, ya que todos los cálculos futuros y los programas finales del proyecto dependen de esas actividades.

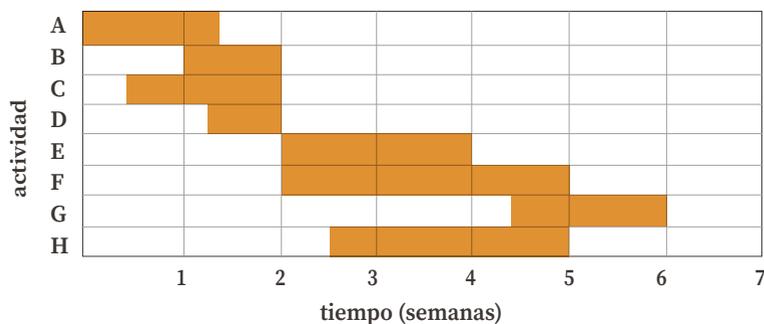
Además de las actividades del proyecto se puede ver en la tabla 7.2 las que son predecesoras del proyecto, que se refieren a la actividad inmediatamente anterior, lo que significa que, para una actividad determinada deben terminarse todas las precedentes inmediatas antes de poder comenzar esa actividad. En el ejemplo 7.1 las actividades B, C y D no pueden comenzar hasta que la actividad A se haya terminado; esto significa que antes de limpiar y pintar la base, antes de rebobinar la armadura y antes de reemplazar los anillos, debe retirarse y desar-

mar el motor. Este ejemplo indica que la actividad E no puede comenzar mientras no se hayan terminado las actividades B, C y D.

Muchas veces son evidentes las relaciones de precedencia, sin embargo es importante que sean completas y precisas. La omisión, así como también la inclusión de una actividad precedente puede distorsionar en gran medida las relaciones generales entre las actividades.

## 7.4 ESTRUCTURA DE RED

Una vez que se tiene una lista completa y precisa de actividades y sus predecesoras se pueden ver gráficamente sus relaciones. Antes del desarrollo de PERT se usaban diagramas de barras diseñadas por H. L. Gantt y que se conocen como diagramas de Gantt para ilustrar la duración y relaciones de tiempo de un proyecto de manera gráfica. Como se puede observar en el diagrama de barras de la tabla 7.3 donde se ilustra un proyecto de ocho actividades, la técnica es conceptualmente correcta pero no puede mostrar de manera clara las relaciones de precedencia, por ejemplo ¿las actividades E y F dependen de la terminación de las actividades B y D? es imposible responder a esta y otras preguntas usando un diagrama de barras, sin embargo, un diagrama de red puede ilustrar la relación apropiada.

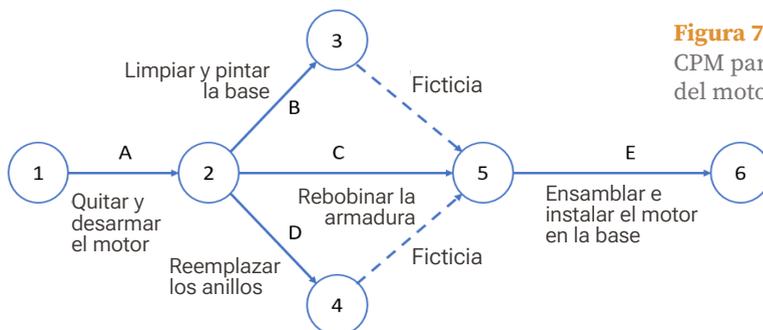


**Tabla 7.3**  
Diagrama de barras

En la figura 7.1 siguiente se resumen los pasos de estas técnicas, CPM y PERT. Primero se definen las actividades del proyecto, sus relaciones de precedencia y sus necesidades de tiempo. A continuación, el proyecto se traduce en una red que muestre las relaciones de precedencia entre las actividades. El tercer paso implica cálculos específicos de redes, que forman la base de desarrollo del programa del proyecto en función del tiempo.

Durante la ejecución del proyecto, podría no cumplirse el programa que estaba planeado, provocando que algunas actividades se atrasen o se adelanten. En este caso será necesario actualizar el programa para que refleje la realidad. Esta es la razón de incluir un bucle, lazo o ciclo de retroalimentación entre la fase del programa y la fase de red como se ve en la figura 7.1.

La figura siguiente es un diagrama de red o diagrama de red PERT/CPM para el ejemplo 7.1 de ajuste del motor. La red consta de círculos llamados nodos y flechas llamadas arcos que los unen. En una red PERT/CPM los arcos representan actividades y los nodos representan eventos. Las actividades implican tiempo, y como se verá más adelante por lo general consumen recursos en forma de mano de obra, materiales o dinero. Los eventos no consumen ni tiempo ni recursos, sino que más bien, sirven como “puntos de referencia del proyecto” y representan los puntos lógicos de conexión para asociar las diversas actividades.



**Figura 7.1.** Red PERT/CPM para ensamblado del motor

Al comparar la tabla 7.2 con la figura 7.1 se confirma que la red ilustra las relaciones apropiadas de precedencia. La actividad A precede a las actividades B, C y D, y la actividad E no puede comenzar sino hasta que se hayan terminado esas tres actividades. Examinando los datos de la tabla y la figura se puede inferir que si se proporcionan las relaciones de precedencia la elaboración de los diagramas de red es bastante simple. Esto es correcto, sin embargo, no debe intentarse dibujar el diagrama de red sin listar primero todas las actividades, sin haberlas ordenado en alguna forma lógica, y sin haber identificado en la medida de lo posible las relaciones de precedencia.

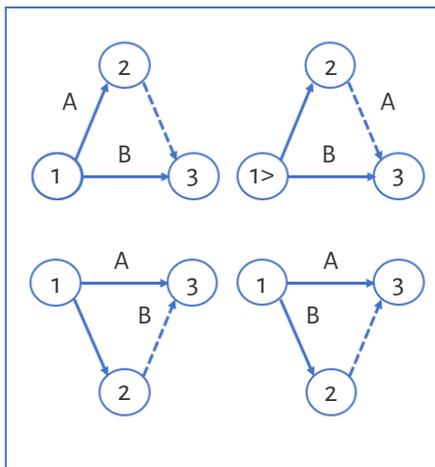
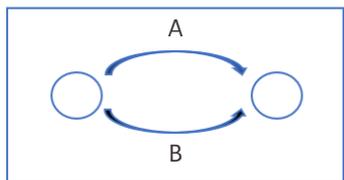
## 7.5 ELABORACIÓN DE LA RED

Cada actividad del proyecto se representa con un arco que apunta en la dirección de avance del proyecto. Los nodos de la red establecen las relaciones de precedencia entre las diferentes actividades del proyecto. Para configurar la red se dispone de dos reglas:

**Regla 1** Cada actividad se representa con un arco, y uno solo.

**Regla 2** Cada actividad se debe identificar con dos nodos distintos.

La figura 7.2 siguiente muestra cómo se puede usar una actividad ficticia para representar dos actividades concurrentes, A y B. Por definición, la actividad ficticia, que normalmente se representa con un arco de línea interrumpida, no consume tiempo o recursos. La inserción de una actividad ficticia en una de las cuatro formas que se ve en la figura 7.2 mantiene la concurrencia de A y B, y también proporciona nodos finales únicos para las dos actividades (para satisfacer la regla 2).



**Figura 7.2** Uso de actividades ficticias en la red

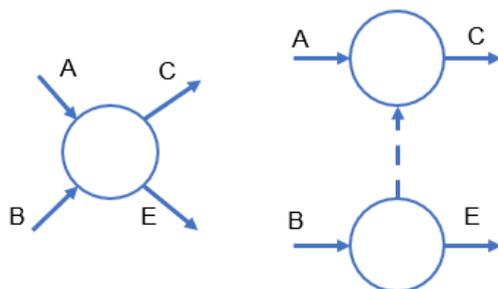
**Regla 3** Para mantener las relaciones de precedencia correctas, se deben contestar las siguientes preguntas cuando se agrega a la red cada actividad:

- a) ¿Qué actividades deben anteceder inmediatamente a la actividad actual?
- b) ¿Qué actividades deben seguir inmediatamente a la actividad actual?
- c) ¿Qué actividades deben efectuarse en forma concurrente o simultánea con la actividad actual?

Para contestar estas preguntas se podrá necesitar el uso de actividades ficticias, para asegurar las precedencias correctas entre las actividades. Por ejemplo, considere el siguiente segmento de un proyecto:

1. La actividad C comienza de inmediato después de haber terminado A y B.
2. La actividad E se inicia después de que solo terminó la actividad B.

La parte (a) de la figura 7.3 muestra la representación incorrecta de esta relación de precedencia, porque pide que A y B terminen antes de poder iniciar E. En la parte B se corrige la situación con el uso de la actividad ficticia.



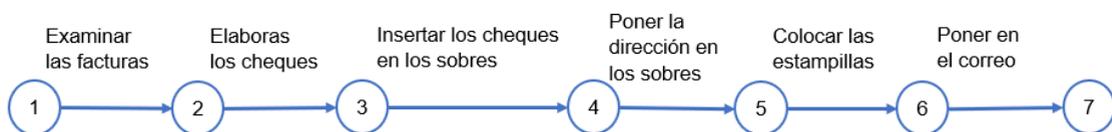
**Figura 7.3** Otro caso de actividades ficticias

Además de las reglas descritas se pueden considerar una serie de pasos que faciliten la construcción de esta. Los pasos se listan a continuación.

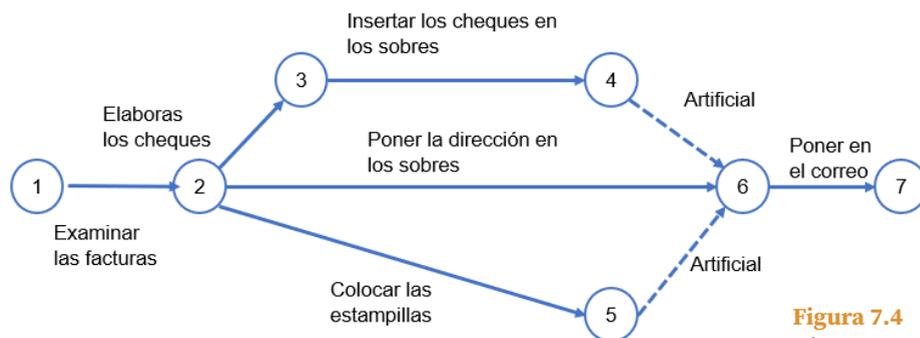
1. Antes de que pueda comenzar una actividad, todas las actividades precedentes deben haber terminado.
2. Las flechas indican solo precedencia lógica, ni su longitud ni su dirección tienen significado alguno.
3. Cada flecha (actividad) debe comenzar y terminar en un nodo de evento.
4. Ningún par de nodos de la red puede estar directamente conectado por más de una flecha.
5. Cuando se enumeran los nodos es aconsejable, y en particular en una red grande, utilizar múltiplos de 10 para que sea fácil incorporar cualesquiera cambios o adiciones futuros.
6. Todas las flechas de la red deben estar dirigidas, más o menos de izquierda a derecha.
7. La clasificación de las actividades (es decir, el listado de las actividades del proyecto) no debe ser más detallado que lo que se requiera para representar un plan de acción lógico y claramente definido.

## Ejemplo 7.2

Uno de los errores comunes que se cometen en la lógica de las redes es colocar las actividades en la red con base en algún sentido del tiempo (es decir, considerando el momento en que es probable que ocurran). En la figura 7.4 parte a), se presenta una red secuencial, donde es probable que se consideraran esta serie de tiempos y actividades al pagar una factura. Aunque la red puede ilustrar la forma en que podrían ordenarse las actividades, equivocadamente dice que “poner la dirección en los sobres” no puede comenzar sino hasta que “se elaboren los cheques” haya sido terminada, lo cual por supuesto no es cierto.



(a) Diagrama secuencial de red para pagar facturas



**Figura 7.4**  
Diagramas de red para proyecto pago de facturas

En la parte b) de la figura 7.4 se presenta la relación apropiada entre las actividades. Aquí no existe restricción sobre la anotación de las direcciones en los sobres (o sobre la colocación de las estampillas).

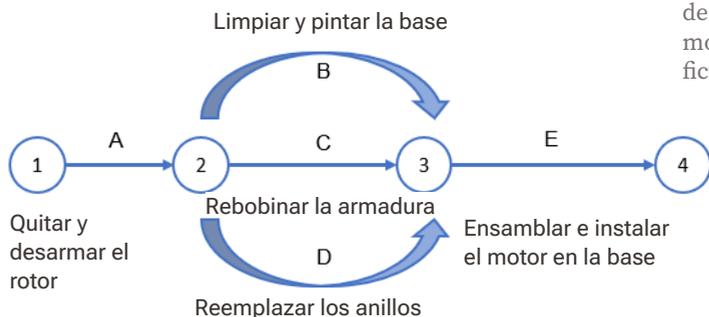
El diagrama simplemente dice que antes de que se pueda poner la carta en el correo, el cheque debe estar en el sobre y este debe tener la dirección y las estampillas.

A partir de este ejemplo simple, resulta muy evidente que las actividades no deben colocarse en serie a menos que sea absolutamente necesario. En cualquier caso, en que puedan ejecutarse en forma simultánea dos o más actividades, esta posibilidad debe estar reflejada en la red, aun cuando no haya sido esta la práctica anterior. Usando este procedimiento se tiene como resultado una mayor flexibilidad en la planeación, así como una reducción en el tiempo de terminación del proyecto.

## 7.6 MÁS SOBRE LAS ACTIVIDADES FICTICIAS

Al examinar la figura 7.1 se observa que dos actividades que no han sido identificadas se representan mediante flechas punteadas. A estas actividades se les denomina actividades ficticias y consumen cero tiempos y cero recursos. Se utilizan las actividades ficticias para mostrar relaciones correctas entre actividades y/o para evitar tener que conectar en forma directa dos nodos a través de más de una flecha, lo que violaría el punto 4 al construir redes. Si se omitieran las actividades ficticias de la figura 7.1, la red resultante tendría la forma de la figura 7.5, donde se ilustran las mismas relaciones entre las actividades, pero da como resultado unir dos nodos con varias actividades, lo cual debe evitarse al construir una red.

**Figura 7.5** Diagrama de red para ajuste de motor sin actividades ficticias



### Ejemplo 7.3

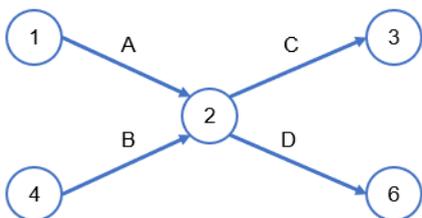
Para ilustrar mejor la necesidad de las actividades artificiales, considere un proyecto básico de instalación de una máquina donde el proyecto consiste en la instalación y capacitación del operador. La capacitación puede comenzar una vez que se contrate al operador y se instale la máquina. La capacitación no depende de la inspección de la máquina, sin embargo, se requiere la inspección una vez que se ha instalado la máquina. La tabla 7.4 ilustra las actividades y sus relaciones de precedencia.

**Tabla 7.4** Actividades para la instalación de la máquina

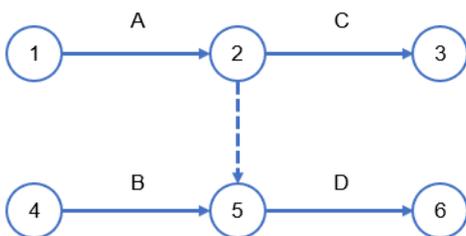
Código de Actividad	Descripción de la actividad	Predecesores inmediatos
A	Instalar la máquina	-
B	Contratar al operador	-
C	Capacitar al operador	A, B
D	Inspeccionar la máquina	A

En la figura 7.6 parte a) se muestra una representación inapropiada y en la parte b) la manera apropiada usando actividades ficticias ya que de otra manera es imposible representar correctamente las relaciones

de precedencia. En la parte a) se ve que las actividades A y B preceden a las actividades C y D, pero la actividad de inspeccionar la máquina o sea la actividad D aparece como dependiente de ellas dos cuando solo le precede la actividad A. El problema se corrige usando la actividad ficticia que aparece en la parte b). Aunque es posible que no en todas las redes se requieran actividades ficticias, en proyectos grandes y complejos se pueden necesitar varias de ellas.



(a) Diagrama inapropiado, dependencia falsa



(b) Diagrama apropiado de red para el proyecto de instalación de una máquina; se requiere una actividad ficticia

**Figura 7.6** Uso de actividad ficticia para una relación apropiada de precedencia

## 7.7 ANÁLISIS DE UNA RED PERT/CPM

Una vez construida la red, el paso siguiente consiste en identificar un programa compatible de actividades que permita la terminación del proyecto en una cantidad mínima de tiempo, para lo cual se requiere identificar los tiempos inicial y final de cada actividad, las relaciones de tiempo entre las actividades y las actividades críticas que deben

terminarse de acuerdo con el programa. El análisis que se hace es básicamente determinístico ya que se consideran los tiempos esperados de las actividades, posteriormente se verá cómo se puede incorporar incertidumbre en los tiempos de las actividades.

## E

**Ejemplo 7.4****Compañía Sharp**

La compañía Sharp fabrica una línea completa de productos para afeitar. Recientemente, un competidor presentó una nueva rasuradora con hoja doble, que en los últimos seis meses ha absorbido una parte significativa de un mercado que la Sharp había tenido por años. Los administradores de la Sharp han decidido introducir un nuevo producto para lo cual se han identificado las tareas que se requieren para diseñar, desarrollar y comercializar el nuevo producto y el tiempo esperado que se requiere para llevar cada una de ellas. Con base en esta información se desea conocer:

- El tiempo total que se requiere desde el inicio del proyecto hasta que el producto nuevo se encuentre en las manos del distribuidor.
- Las fechas específicas de inicio y terminación de cada tarea.
- Las tareas críticas, es decir las que deben terminarse a tiempo para que el proyecto concluya en una fecha específica.

En la tabla 7.5 siguiente se describen las actividades, sus predecesores inmediatos y el tiempo esperado de terminación.

1

2

3

4

5

6

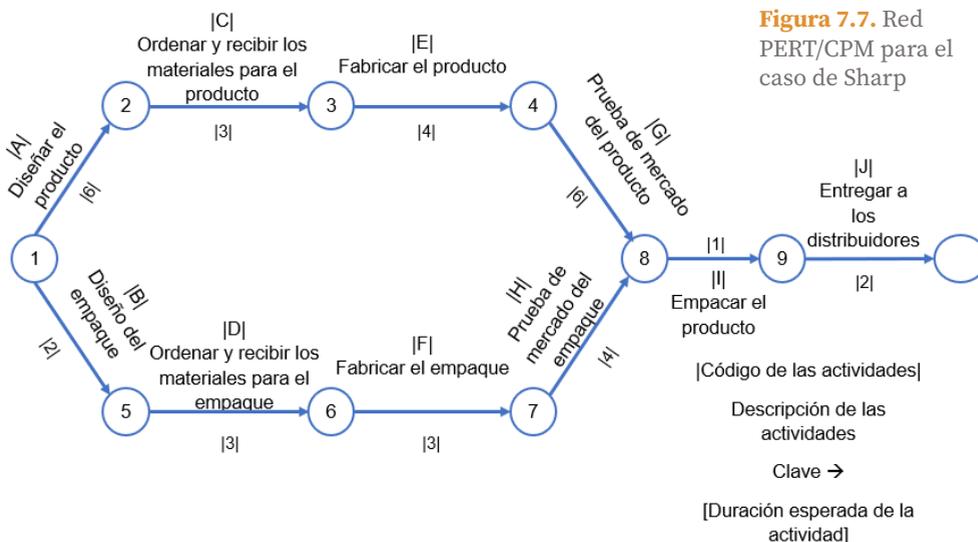
7

8

**Tabla 7.5.** Datos para el proyecto de la Sharp

Código de Actividad	Descripción de la actividad	Predecesores inmediatos	Tiempo esperado para terminar (semanas)
A	Diseñar el producto	-	6
B	Diseñar el empaque	-	2
C	Ordenar y recibir los materiales para el producto	A	3
D	Ordenar y recibir los materiales para el empaque	B	3
E	Fabricar el producto	C	4
F	Fabricar el empaque	D	3
G	Empacar el producto	E	6
H	Prueba de mercado del producto	F	4
I	Prueba de mercado del empaque	G, H	1
J	Entregar a los distribuidores	I	2

La figura 7.7 muestra la red PERT/CPM para la compañía Sharp, se ha colocado el código de la actividad y su descripción correspondiente encima de cada arco y debajo de cada uno de ellos se ha anotado el tiempo estimado de la actividad.



Una vez que se ha elaborado la red PERT/CPM se puede concentrar la atención en determinar la fecha esperada de terminación del proyecto y el programa de actividades. La fecha general de terminación del proyecto es importante si los administradores están compitiendo con otra empresa en el proyecto o si los administradores operan con base en incentivos por fechas de terminación. El programa de actividades es importante porque señala las actividades críticas del proyecto.

Si se suman todos los tiempos esperados de las actividades de la tabla 7.5, se tienen 34 semanas como duración del proyecto, sin embargo, de la figura 7.7 se sabe que esta no es la cantidad de tiempo que se requerirá para terminar el proyecto, las actividades A y B pueden llevarse a cabo en forma simultánea y la actividad C puede ejecutarse en paralelo con la actividad D (dependiendo cuándo terminen la actividad A y la B). En la figura 7.7 puede no ser evidente la verdadera duración del proyecto, pero con un poco de trabajo se puede calcular el valor correcto.

Se calcula la duración del proyecto determinando la ruta crítica para la red. Toda red tiene dos o más rutas críticas. Se define una ruta crítica como la secuencia de actividades que se llevan a cabo al pasar del evento (nodo) inicial al evento (nodo) final de la red. En el ejemplo de la Sharp el evento inicial es el nodo 1 y el evento final el nodo 10.

Las actividades A, C, E, G, I y J forman una ruta que conecta los nodos 1, 2, 3, 4, 8, 9 y 10 de la red en tanto que las actividades B, D, F, H, I y J forman una segunda ruta que conecta los nodos 1, 5, 6, 7, 8, 9 y 10. Puesto que la terminación de un proyecto requiere que se terminen todas las rutas de la red, la duración de la ruta más larga de la red, es decir el tiempo que se requiere para terminarla es la ruta crítica. Para este ejemplo la ruta A – C – E – G – I requiere 22 semanas en tanto que la ruta

B–D–F–H–I–J requiere solo 15 semanas. Por lo tanto, la ruta A–C–E–G–I es la ruta crítica. Si se demora cualquier actividad sobre la ruta crítica, se demora todo el proyecto. A las actividades que se encuentran sobre la ruta crítica se les llaman actividades críticas. Para reducir el tiempo total del proyecto, 22 semanas en este caso, los administradores deben reducir la duración de una o más actividades críticas.

Para cualquier red se puede:

- Identificar todas las rutas de la red
- Calcular la duración de cada ruta
- Elegir la ruta más larga o crítica.

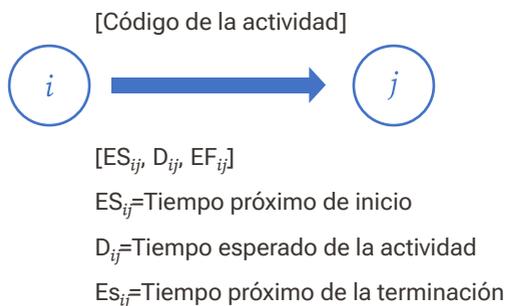
Sin embargo, esta no es una manera eficiente de analizar la red, por lo cual un procedimiento más eficiente consiste en calcular límites de tiempo para cada actividad como tiempos próximos de inicio, lejanos de inicio, próximos de terminación y lejanos de terminación. A partir de estos datos calcular la ruta crítica. Los tiempos próximos de inicio y de terminación se pueden calcular haciendo una revisión hacia adelante de la red. Los tiempos lejanos de inicio y terminación se determinan usando una revisión hacia atrás de la red.

### **Revisión hacia adelante: Cálculo de los tiempos próximos de inicio y terminación**

Para comenzar con el análisis de la revisión hacia adelante, es necesario definir algunos términos e identificar el procedimiento y notación que se usará, más adelante a través de otro ejemplo se verá que existen otras notaciones para los cálculos de tiempos, sin embargo el procedimiento es el mismo. El tiempo próximo de inicio de una actividad es el tiempo más próximo posible en que una actividad puede comen-

zar. En los cálculos se usará  $ES_{ij}$  para designar el tiempo próximo de inicio, donde  $i$  y  $j$  son los nodos iniciales y finales asociados con la actividad. Por ejemplo, en la figura 7.5:  $ES_{12}$  denota el tiempo próximo de inicio para la actividad “diseño del producto” es decir la actividad A, y  $ES_{15}$  denota el tiempo próximo de inicio de la actividad “diseño de empaque” es decir, la actividad B.

El tiempo próximo de terminación para una actividad se denota por  $EF_{ij}$ , es el tiempo próximo de inicio más el tiempo que se requiere para terminar la actividad. Para la actividad A de la figura 7.7 este tiempo sería:  $EF_{12} = ES_{12} + D_{12}$ , donde  $D_{12} = 6$ , el tiempo esperado de la actividad. Si el tiempo próximo de inicio de A es igual a 0, es decir  $ES_{12} = 0$  entonces  $EF_{12} = ES_{12} + D_{12} = 0 + 6 = 6$ . Los cálculos no se muestran en la red, más bien los resultados de aplicar las relaciones y se usa la siguiente notación:

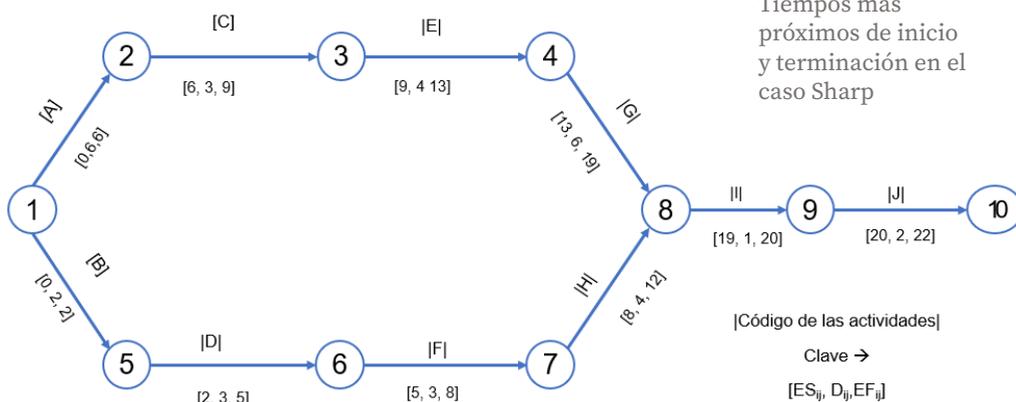


El procedimiento normal para analizar una red consiste en comenzar en el nodo inicial y suponer como lo hemos hecho que se tiene un tiempo inicial de cero. Se supone que todas las actividades comienzan tan pronto como es posible, es decir, tan pronto como se han terminado todas las actividades precedentes asociadas. Puesto que las actividades A y B no tienen predecesoras  $ES_{12} = 0$  y  $ES_{15} = 0$  por lo tanto sus correspondientes tiempos de terminación son  $EF_{12} = 6$  y  $EF_{15} = 2$ .

De este modo se procede con las actividades C y D respectivamente. El tiempo más próximo de inicio de la actividad C es igual al tiempo más próximo de terminación de la actividad A es decir  $ES_{23} = EF_{12} = 6$  y su tiempo más próximo de terminación es  $EF_{23} = ES_{23} + D_{23} = 6 + 3 = 9$ . De la misma manera se calcula para la actividad D y se tiene:

$$ES_{56} = EF_{15} = 2; EF_{56} = ES_{56} + D_{56} = 2 + 3 = 5$$

Continuando con este análisis hacia delante pueden calcularse los tiempos próximos de inicio y después los tiempos próximos de terminación para cada una de las actividades. La figura 7.8 es el diagrama completo de red que ilustra los valores de  $ES_{ij}$  y  $EF_{ij}$



**Figura 7.8.** Tiempos más próximos de inicio y terminación en el caso Sharp

El avance por la red y la elaboración de los cálculos son tareas bastante simples para nodos que tienen una sola actividad precedente, lo cual fue el caso para cada una de las actividades que se examinaron antes. Sin embargo, en los casos en que existen varias actividades precedentes el tiempo más próximo de inicio para esta actividad es igual al mayor valor de los tiempos próximos de terminación para todas las actividades precedentes. Por ejemplo, la actividad I es precedida por

las actividades G y H por lo tanto el tiempo más próximo de iniciación para la actividad I es 19, es decir  $ES_{g_9} = 19$ .

Examinando los cálculos de tiempos para la figura 7.8 se observa que el tiempo más próximo de terminación para la actividad final J es de 22 semanas que fue el mismo valor encontrado con anterioridad.

### Revisión hacia atrás: cálculo de los tiempos lejano de inicio y de terminación

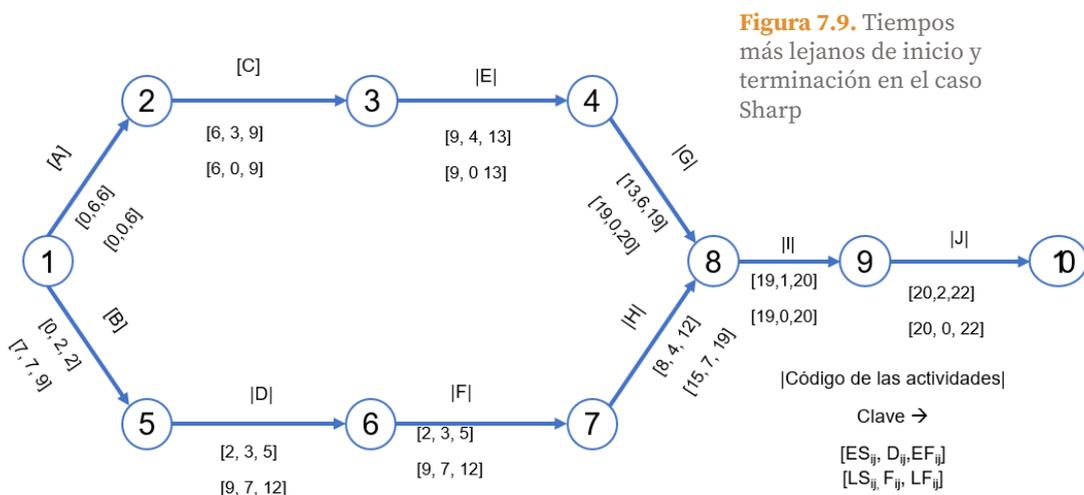
Puede identificarse el tiempo total que se requiere para terminar un proyecto haciendo una revisión hacia delante de la red, pero no se pueden responder preguntas como: ¿Qué tan tarde puede comenzar una actividad específica sin prolongar la duración total del proyecto? ¿Cuánto puede demorarse cada actividad si es posible? Estas preguntas se pueden responder si se hace una revisión hacia atrás en la red.

Para hacer esta revisión hacia atrás se necesitan definir algunos términos como el tiempo más lejano de inicio. El tiempo más lejano de inicio de una actividad  $LS_{ij}$  es el tiempo más lejano o tardado en el que una actividad puede comenzar sin demorar la fecha de terminación del proyecto. El tiempo más lejano de terminación para una actividad  $LF_{ij}$  es el tiempo más lejano de inicio más el tiempo que dura la actividad  $D_{ij}$ . En forma simbólica estas relaciones son:  $LF_{ij} = LS_{ij} + D_{ij}$ ; sin embargo, para la revisión hacia atrás es más útil  $LS_{ij} = LF_{ij} - D_{ij}$ .

Para comenzar los cálculos se empieza por el evento final, en este caso el nodo 10 y se fija el tiempo más lejano de terminación para la última actividad como el tiempo total de duración calculado en la revisión hacia delante, es decir  $LF_{9,10} = 22$ . Puesto que se requieren 2 días para

terminar la actividad J el tiempo más lejano de inicio para la actividad J es igual al tiempo más lejano de terminación menos el tiempo de duración, es decir  $LS_{9,10} = LF_{9,10} - D_{9,10} = 22 - 2 = 20$ . Se procede entonces con la actividad I, el tiempo más lejano de terminación es 20, es decir  $LF_{8,9} = 20$ , el tiempo más lejano de inicio para la actividad J es 19, es decir  $LS_{8,9} = LF_{8,9} = 20 - 1 = 19$ . En el nodo 8, tanto la actividad G como la H ingresan a él, por lo tanto, el tiempo más lejano de terminación para cada uno de ellos es 19, el tiempo más lejano de inicio de la actividad I, que es la única actividad que sale del nodo.

Así se continua con el análisis hacia atrás como se muestra en la figura 7.9, además de los valores de los tiempos más lejanos de inicio y terminación se muestra el tiempo de holgura o flotante (que se define más adelante) como el valor central entre paréntesis.



Un factor que debe observarse con respecto al cálculo de los valores  $LF_{ij}$  para una red es que, si un nodo determinado tiene más de una actividad que sale de él, entonces el tiempo más lejano de terminación para una actividad que entra al nodo es igual al menor valor de

los tiempos más lejanos de inicio para todas las actividades que salen del nodo. Para el caso de la Sharp el único nodo que tiene más de una actividad saliendo de él es el nodo 1, los tiempos más lejanos de inicio para las dos actividades que salen del nodo 1 son A y C con 0 y 7 respectivamente. Esto indica que el tiempo más lejano en que puede comenzar el proyecto general es el tiempo 0.

### Tiempo flotante o de holgura

Después de que se han determinado los límites de tiempo para toda la red, se determina el tiempo de holgura como la longitud de tiempo en la que se puede demorar una actividad sin ocasionar que la duración del proyecto general exceda su tiempo programado de terminación. La cantidad de tiempo de holgura de una actividad se calcula tomando la diferencia entre sus tiempos más lejano de inicio y más próximo de inicio, o entre sus tiempos más lejano y más próximo de terminación, en términos de ecuación se expresa:

$$F_{ij} = LS_{ij} - ES_{ij}$$

$$F_{ij} = LF_{ij} - EF_{ij}$$

De la figura 7.9 se puede ver que el tiempo de holgura de la actividad B es 7 o sea que esa actividad se puede demorar 7 semanas, de la misma manera la actividad D tiene un tiempo flotante de 7 semanas. Como se puede observar las actividades en la ruta crítica tienen un tiempo de holgura o flotante igual a cero, lo que significa que si se quiere terminar el proyecto en 22 semanas se tienen que llevar a cabo esas actividades en los tiempos programados. El análisis de la red también señala que existen 7 semanas de tiempo flotante en esa porción de la red asociadas con el empaque del producto.

Como se decía con anterioridad existe otra notación para hacer los cálculos de la red CPM, a través del ejemplo siguiente se presenta dicha notación.

## Ejemplo 7.5

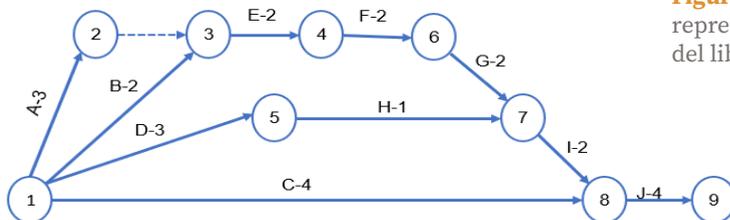
### Edición de libro

Un editor tiene un contrato con un autor, para publicar su libro de texto. Las actividades (simplificadas) relacionadas con la producción del libro se ven a continuación. Formular la red asociada al proyecto.

**Tabla 7.6.** Datos para el proyecto de edición de libro

Código y descripción de la actividad	Predecesores	Duración (semanas)
A. Lectura del manuscrito por el editor	-	3
B. Preparación de páginas muestra por el tipógrafo	-	2
C. Diseño de la portada del libro	-	4
D. Preparación de las figuras del libro	-	3
E. Aprobación por el autor del manuscrito editado y las páginas muestra	A, B	2
F. Tipografía del libro	E	2
G. Revisión por el autor de las páginas tipografiadas	F	2
H. Revisión de las figuras por el autor	D	1
I. Producción de las placas de impresión	G, H	2
J. Producción y encuadernación del libro	C, I	4

La figura 7.10 muestra la red que describe las relaciones de precedencia entre las diversas actividades. Con la actividad ficticia (2, 3) se obtienen nodos finales únicos para las actividades concurrentes A y B. La numeración de los nodos se hace en forma que indique el avance del proyecto.



**Figura 7.10** Red que representa la edición del libro

## Cálculos para la ruta crítica

El resultado final de CPM es la formulación o construcción del programa del proyecto. Para lograr este objetivo en una forma adecuada, se hacen cálculos especiales con los que se obtiene la siguiente información:

1. Duración total necesaria para terminar el proyecto.
2. Clasificación de las actividades del proyecto en críticas y no críticas.

Se dice que una actividad es crítica si no hay margen en la determinación de sus tiempos de inicio y de término. Una actividad no crítica permite alguna holgura en su programación de modo que el tiempo de inicio de la actividad se puede adelantar o retrasar dentro de ciertos límites, sin afectar la fecha de terminación de todo el proyecto.

Para efectuar los cálculos necesarios, se define un evento como un momento en el tiempo en el que se terminan actividades y otras se inician. En términos de redes, un evento corresponde a un nodo. Se define lo siguiente:

$\square_j$  = Tiempo más temprano de ocurrencia del evento  $j$

$\Delta_j$  = Tiempo más tardío de ocurrencia del evento  $j$

$D_{ij}$  = Duración de la actividad  $(i, j)$

Las definiciones de los tiempos más temprano y más tardío del evento se especifican en relación con las fechas de inicio y terminación de todo el proyecto.

Los cálculos de ruta crítica implican dos pasos: el paso hacia adelante determina los tiempos más tempranos o de ocurrencia de los eventos, y el paso hacia atrás calcula sus tiempos más tardíos de ocurrencia.

**Paso hacia adelante (tiempos más tempranos de ocurrencia o tiempos más próximos de ocurrencia,  $\square$ ).** Los cálculos se inician en el nodo 1 y avanzan en forma recursiva hasta el nodo final  $n$ .

**Paso inicial.** Poner  $\square_1 = 0$ , para indicar que el proyecto se inicia cuando el tiempo es 0.

**Paso general  $j$ .** Dado que los nodos  $p, q, \dots, v$ , están enlazados directamente con el nodo  $j$  por las actividades de entrada  $(p, j), (q, j), \dots, (v, j)$  y que los tiempos más tempranos de ocurrencia de los eventos (nodos)  $p, q, \dots, v$  ya se han calculado, entonces se calcula el tiempo más temprano de ocurrencia del evento  $j$  como sigue:

$$\square_j = \text{máx} \{ \square_p + D_{pj}, \square_q + D_{qj}, \dots, \square_v + D_{vj} \}$$

El paso hacia adelante se termina cuando se calcula  $\square_n$  en el nodo  $n$ . Por definición  $\square_j$  representa la ruta (duración) más larga al nodo  $j$ .

**Paso hacia atrás (tiempos más tardíos de ocurrencia o tiempos más lejanos de ocurrencia,  $\Delta$ ).** Después de terminar el paso hacia delante, los cálculos del paso hacia atrás comienzan en el nodo  $n$  y terminan en el nodo 1.

**Paso inicial.** Igualar  $\Delta n = n$  para indicar que las ocurrencias más temprano y más tardío del último nodo en el proyecto son iguales.

**Paso general  $j$ .** Dado que los nodos  $p, q, \dots, v$  están enlazados en forma directa con el nodo  $j$  por actividades de salida  $(j, p), (j, q), \dots, (j, v)$ , y que se calcularon los tiempos más tardíos de los nodos  $p, q, \dots, v$ , el tiempo tardío del nodo  $j$  se calcula como sigue:

$$\Delta_j = \min \{ \Delta_p - D_{jp}, \Delta_q - D_{jq}, \dots, \Delta_v - D_{jv} \}$$

El paso hacia atrás se termina cuando se calcula  $\Delta_1$  en el nodo 1.

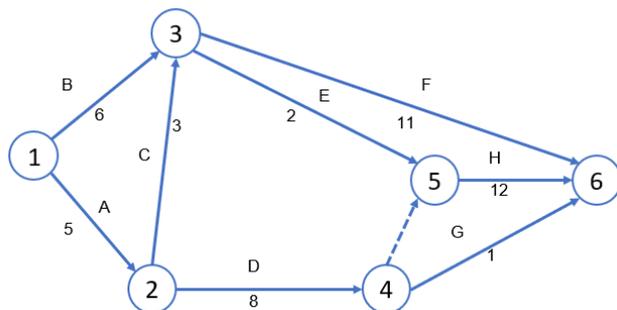
Con base en los cálculos anteriores, una actividad  $(i, j)$  será crítica si satisface tres condiciones:

1.  $\Delta_i = \square_i$
2.  $\Delta_j = \square_j$
3.  $\Delta_j - \Delta_i = \square_j - \square_i = D_{ij}$

Las tres condiciones indican que los tiempos más tempranos y los tiempos más tardíos de ocurrencia de los nodos  $i$  y  $j$  son iguales, y que la duración  $D_{ij}$  se ajusta exactamente al intervalo especificado de tiempo. Una actividad que no satisface las tres condiciones es no crítica. Las actividades críticas de una red deben formar una trayectoria no interrumpida que abarque toda la red, desde el inicio hasta el final.

### Ejemplo 7.6

Determine la ruta crítica para la red del proyecto de la figura 7.11. Todas las duraciones están en días.



**Figura 7.11**  
Red del proyecto

#### Paso hacia adelante

**Nodo 1.** Hacer o definir  $\square_1 = 0$

**Nodo 2.**  $\square_2 = \square_1 + D_{12} = 0 + 5 = 5$

**Nodo 3.**  $\square_3 = \text{máx.} \{ \square_1 + D_{13}, \square_2 + D_{23} \} = \text{máx.} (0 + 6, 5 + 3) = 8$

**Nodo 4.**  $\square_4 = \square_2 + D_{24} = 5 + 8 = 13$

**Nodo 5.**  $\square_5 = \text{máx.} \{ \square_3 + D_{35}, \square_4 + D_{45} \} = \text{máx.} (8 + 2, 13 + 0) = 13$

**Nodo 6.**  $\square_6 = \text{máx.} \{ \square_3 + D_{36}, \square_4 + D_{46}, \square_5 + D_{56} \} =$   
 $= \text{máx.} (8 + 11, 13 + 1, 13 + 12) = 25$

Los cálculos indican que el proyecto se puede terminar en 25 días.

#### Paso hacia atrás

**Nodo 6.** Hacer  $\Delta_6 = \square_6 = 25$

**Nodo 5.**  $\Delta_5 = \Delta_6 - D_{56} = 25 - 12 = 13$

**Nodo 4.**  $\Delta_4 = \text{mín.} \{ \Delta_6 - D_{46}, \Delta_5 - D_{45} \} = \text{mín.} \{ 25 - 1, 13 - 0 \} = 13$

**Nodo 3.**  $\Delta_3 = \text{mín.} \{ \Delta_6 - D_{36}, \Delta_5 - D_{35} \} = \text{mín.} \{ 25 - 11, 13 - 2 \} = 11$

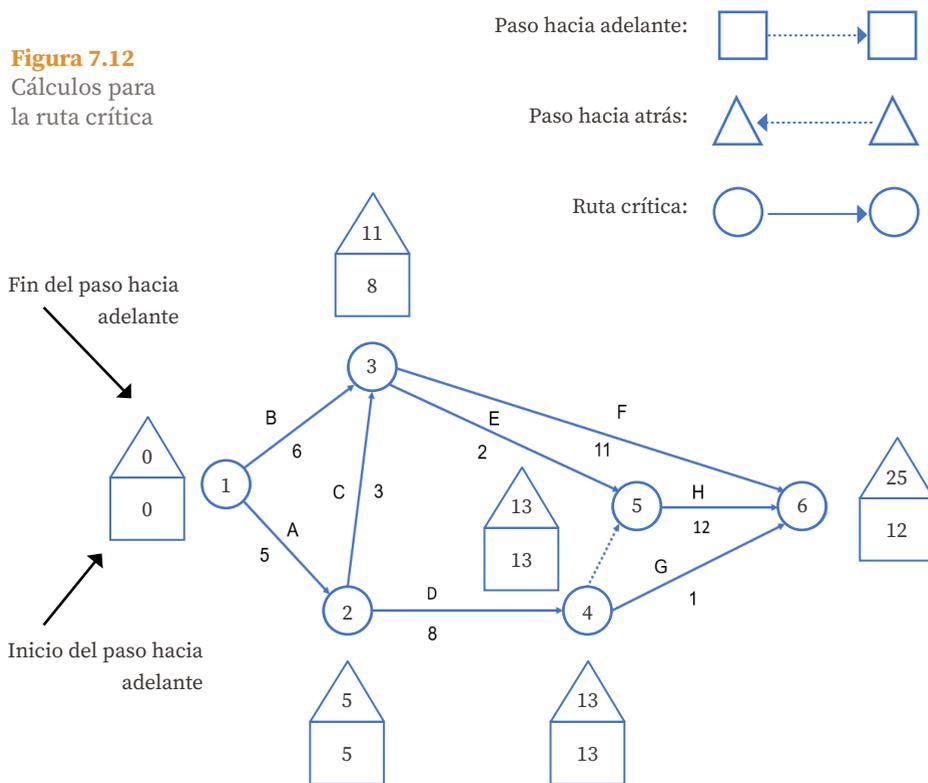
**Nodo 2.**  $\Delta_2 = \text{mín.} \{ \Delta_4 - D_{24}, \Delta_3 - D_{23} \} = \text{mín.} \{ 13 - 8, 11 - 3 \} = 5$

**Nodo 1.**  $\Delta_1 = \text{mín.} \{ \Delta_3 - D_{13}, \Delta_2 - D_{12} \} = \text{mín.} \{ 11 - 6, 5 - 5 \} = 0$

Si los cálculos son correctos, siempre se terminarán con  $\Delta_1 = 0$ .

Los cálculos en los pasos hacia delante y hacia atrás se resumen en la figura 7.12. Las reglas para determinar las actividades indican que la ruta crítica es  $1 \rightarrow 2 \rightarrow 4 \rightarrow 5 \rightarrow 6$  que abarca la red desde el inicio (nodo 1) hasta el nodo final (nodo 6). La suma de las duraciones de las actividades críticas (1,2) (2,4), (4,5), y (5,6) es igual a la duración del proyecto (= 25 días). Observe que la actividad (4, 6) satisface las primeras dos condiciones para que la actividad sea crítica pero no la tercera por lo tanto esa no es una actividad crítica.

**Figura 7.12**  
Cálculos para la ruta crítica



## 7.8 PROBLEMAS CON INCERTIDUMBRE O PERT<sup>3</sup>

Como ya se mencionó en un inicio la Técnica de Revisión y Evaluación de Programas (o Proyectos), denominada PERT por sus siglas en inglés, es un método estadístico para el análisis de las actividades que conforman un determinado proyecto. Esta técnica incorpora la incertidumbre en la programación del proyecto, estableciendo diferentes estimaciones de tiempos para cada actividad, ya que no se conocen con exactitud su duración.

A diferencia del método CPM, este método está más orientado a la gestión del tiempo del proyecto que de los costes. Su aplicación está relacionada con proyectos de infraestructura y de Investigación y Desarrollo.

En la actualidad se utiliza de forma complementaria con el método CPM como un método global denominado Método PERT CPM.

La técnica PERT data de la década de 1950; específicamente en el año 1957 para la Oficina de Proyectos Especiales de la Marina de Estados Unidos, para gestionar el programa de misiles submarinos Polaris.

En 1958 se difundió al público a través de dos publicaciones del Departamento de la Marina de Estados Unidos bajo el título PERT – *Program Evaluation Research Task*. A partir de entonces, se ha utilizado ampliamente en diferentes proyectos, no solo por la industria aeroespacial, sino en diferentes sectores.

---

<sup>3</sup> <https://www.plandemejora.com/pert-tecnica-de-revision-y-evaluacion-de-programas/>

Para aplicar el método PERT seguiremos los siguientes pasos:

**Paso 1: Determinar las actividades y sus relaciones.**

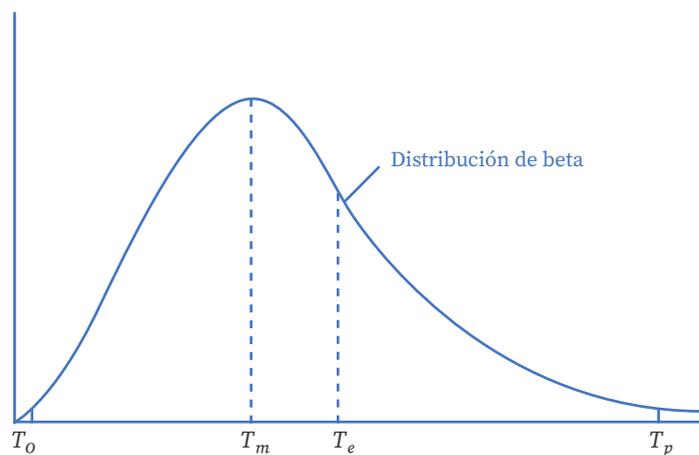
La planificación del proyecto es una fase crucial para culminarlo exitosamente. En ese aspecto, es necesario identificar las actividades que lo componen y las relaciones entre ellas.

**Paso 2: Elaborar el diagrama PERT.**

El diagrama de redes nos permite visualizar en su totalidad las actividades del proyecto y sus relaciones. Para este paso, hemos elaborado un artículo donde abordamos de manera detallada la elaboración del diagrama PERT.

**Paso 3: Definir las estimaciones de tiempos.**

Para el análisis estadístico del método PERT, se considera el tiempo de cada actividad como una variable aleatoria con una distribución de probabilidad beta. Esta distribución se encuentra sesgada a la derecha de la siguiente forma:



**Figura 7.13**  
Distribución beta  
para estimación  
de tiempos

Para realizar el análisis estadístico se requiere que la duración de las actividades se presente en términos de tres estimaciones de tiempo:

***Tiempo optimista ( $T_o$ ):***

Tiempo que tomará una actividad si todo sale como se planeó. Hay solo una pequeña probabilidad, que por lo general se supone de 1 %, de que la actividad tarde menos tiempo.

***Tiempo pesimista ( $T_p$ ):***

Tiempo que tomará una actividad suponiendo condiciones muy desfavorables. Hay solo una pequeña probabilidad, que por lo general se supone de 1 %, de que tarde más.

***Tiempo más probable ( $T_m$ ):***

La estimación más realista del tiempo requerido para terminar la actividad. Viene a ser la moda de la distribución beta.

Estos tiempos deben ser determinados entre el gerente de proyecto y los miembros del equipo que estarán a cargo de las actividades.

**Paso 4: Calcular los tiempos esperados.**

A partir de nuestra distribución beta, convertiremos el proyecto en una red PERT de tiempo constante utilizando los tiempos esperados,  $T_e$ , para cada actividad. A continuación, tenemos la fórmula del tiempo esperado, que está en función del tiempo pesimista y tiempo optimista:

$$T_e = (T_o + 4T_m + T_p) / 6$$

La experiencia muestra que los tiempos reales ( $T_e$ ) de las actividades con frecuencia exceden al tiempo más probable ( $T_m$ ) en las actividades de un proyecto; esto sucede porque las personas tienden a ser excesivamente optimistas en ello.

**Paso 5: Calcular la ruta crítica y el tiempo total del proyecto.**

Con los tiempos esperados calculados en el paso 4, podemos determinar la ruta crítica y el tiempo total del proyecto. Tenemos un artículo detallado donde explicamos cómo calcular la ruta crítica.

**Paso 6: Calcular la varianza del proyecto.**

Aun con el cálculo del tiempo esperado, el problema de la incertidumbre en los tiempos de las actividades no se elimina tan fácilmente. Dado que el tiempo de cada actividad es incierto, entonces el tiempo total de terminación del proyecto también será incierto.

Para resolver este problema, el método PERT utiliza la varianza de las actividades de la ruta crítica para ayudar a determinar la varianza del proyecto global.

La varianza ( $Var_i$ ) de cada actividad ( $i$ ) de la ruta crítica se calcula así:

$$Var_i = (T_p - T_o / 6)^2$$

La varianza del proyecto se calcula así:

$$Var(T) = \sum_{\text{ruta crítica}} Var_i$$

El cálculo de la varianza nos ayudará a determinar la probabilidad de terminar el proyecto en un plazo determinado.

La técnica PERT supone que la duración de una actividad no depende de la duración de ninguna de las otras actividades; es decir son variables aleatorias independientes. Sin embargo, si a una cuadrilla de trabajadores se le asignan dos actividades que pueden realizarse simultáneamente, los tiempos de actividad serán interdependientes. Adicional-

mente, si otras rutas contenidas en la red tienen cantidades mínimas de holgura, una de ellas se podría convertir en la ruta crítica antes de que el proyecto llegue a su fin. En ese caso, también tendríamos que calcular la distribución de probabilidades para esas rutas.

## E

## Ejemplo 7.7

## CPM con tiempo estimado de tres actividades

En la tabla 7.7 siguiente se muestran los tiempos de nueve tareas considerando el tiempo pesimista, el optimista y el tiempo más probable.

**Tabla 7.7** Tiempos de las tareas

Tarea	Predecesores inmediatos	Tiempo optimista	Tiempo más probable	Tiempo pesimista
A	Ninguno	3	6	15
B	Ninguno	2	4	14
C	A	6	12	30
D	A	2	5	8
E	C	5	11	17
F	D	3	6	15
G	B	3	9	27
H	E, F	1	4	7
I	G, H	4	19	28

A partir de estos tiempos se calcula el tiempo esperado como sigue:

**Tabla 7.8** Tiempo esperado

Tarea	Predecesores inmediatos	Tiempo esperado
A	Ninguno	7
B	Ninguno	5.333
C	A	14
D	A	5
E	C	11
F	D	7
G	B	11
H	E, F	4
I	G, H	18

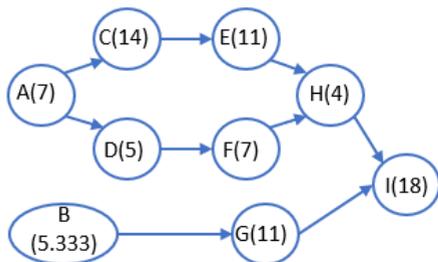
La fórmula asociada es la siguiente:

$$\text{Tiempo esperado} = \frac{\text{Tiempo opt.} + 4(\text{tiempo más probable}) + \text{Tiempo pes.}}{6}$$

Sustituyendo en el problema original se tiene:

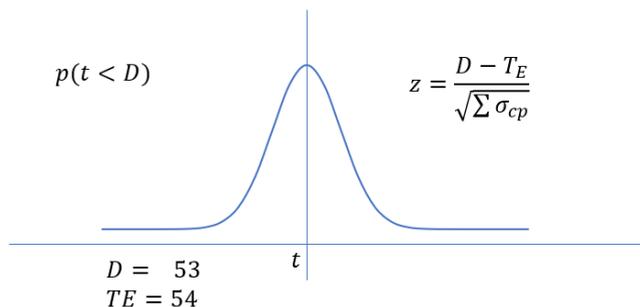
$$T_e = \frac{3 + 4(6) + 15}{6} = ET(A) = 42/6 = 7$$

La red asociada a este problema con la duración se ve en la figura 7.14.



**Figura 7.14**  
Red de la ruta crítica

La siguiente pregunta que se hace es: ¿Cuál es la probabilidad de terminar este proyecto en 53 días? Entonces se ve la varianza como una herramienta para contestar esto:



**Figura 7.15** Varianza en el tiempo de duración del proyecto

Se tiene entonces la siguiente tabla con la fórmula para la varianza en este ejemplo:

$$\text{Varianza de actividades, } \sigma^2 = \left( \frac{\text{Pesim.} - \text{Optim.}}{6} \right)^2$$

En la tabla 7.9 se muestran las tareas con los tiempos pesimista, promedio y optimista, así como la varianza.

**Tabla 7.9** Tiempos y varianza

Tarea	Optimista	Promedio	Pesimista	Varianza
A	3	6	15	4
B	2	4	14	
C	6	12	30	16
D	2	5	8	
E	5	11	17	4
F	3	6	15	
G	3	9	27	
H	1	4	7	1
I	4	19	28	16

(Suma de la varianza a lo largo de la ruta crítica) =  $\sum \sigma^2 = 41$

$$Z = \frac{D - T_E}{\sqrt{\sum \sigma_{cp}^2}} = \frac{53 - 54}{\sqrt{41}} = -.156$$

Una última pregunta es: ¿Cuál es la probabilidad de que el proyecto exceda 56 semanas?

$$Z = \frac{D - T_E}{\sqrt{\sum \sigma_{cp}^2}} = \frac{56 - 54}{\sqrt{41}} = .312$$

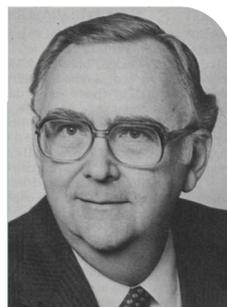
$$P(Z < -.156) = .5 - .1217 = .378, \text{ o } 37.8 \%$$

Podemos decir a manera de conclusión que la ruta crítica nos muestra las actividades principales que determinarán el tiempo total del proyecto; es por ello por lo que los gerentes del proyecto deben estar pendientes del cumplimiento de cada una de ellas. Eso no significa que se deben descuidar las otras actividades; por el contrario, si se retrasan un tiempo mayor a su holgura, el proyecto tendrá problemas para finalizar a tiempo. Y por otro lado la técnica PERT es muy beneficiosa para crear predicciones más eficientes y precisas para un proyecto. Este método puede utilizarse en muchos ámbitos diferentes y es una herramienta fantástica para la planificación de contingencias, la programación de proyectos y los “riesgos del proyecto”. Por lo que es un tema indispensable en la gestión de proyectos.

## 7.9 NOTAS HISTÓRICAS

CPM apareció en 1957 como una herramienta desarrollada por J. E. Kelly de Remington Rand y M.R Walker de DuPont.

*“De hecho, somos afortunados de contar con esta contribución a la literatura de gestión de proyectos de Jim Kelley y Morgan Walker, los autores de uno de los artículos fundamentales y las principales técnicas para la planificación y programación de proyectos”. Como afirman en el artículo, estuvieron en “el lugar correcto en el momento correcto” pero también lograron los objetivos de su proyecto.*



James E. Kelly Jr.



M.R. Walker

*Afirman, además, que es necesario comprender el entorno que existía en DuPont y en Remington Rand Univac en el momento de sus esfuerzos. En este entorno se incluyeron dos hombres que desempeñaron papeles importantes en sus respectivas organizaciones, Granville Read, ingeniero jefe de DuPont y John Mauchly de Univac. Para ayudar a describir este entorno, también tenemos la suerte de contar con un apéndice, "Departamento de Ingeniería de DuPont en*

*la década de 1950: un entorno para nuevas ideas", de John Sayer, Gerente del Grupo de Control de Ingeniería Integrada, un grupo de expertos en DuPont en ese momento. Este fue el grupo del que Morgan Walker era miembro y que alimentó el proyecto que condujo al desarrollo de CPM.” Tomado de Kelley, J. E., Walker, M. R., & Sayer, J. S. (1989). The origins of CPM: a personal history. PM Network, 3(2), 7–22.*

Un proyecto puede tener varias rutas críticas paralelas. Una ruta paralela adicional a través de la red con las duraciones totales menos cortas que la ruta crítica es llamada una sub-ruta crítica.

Originalmente, el método de la ruta crítica consideró solamente dependencias entre los elementos terminales. Un concepto relacionado es la cadena crítica, la cual agrega dependencias de recursos.

Cada recurso depende del manejador en el momento donde la ruta crítica se presente.

A diferencia de la técnica de revisión y evaluación de programas (PERT), el método de la ruta crítica usa tiempos ciertos (reales o determinísticos). Sin embargo, la elaboración de un proyecto con base en redes CPM y PERT es similar.



PERT se desarrolló en la década de 1950 y se usó ampliamente en la administración de proyectos militares de investigación y desarrollo. Su primera aplicación importante fue en el proyecto de los misiles Polaris para la US Navy. De hecho, PERT fue desarrollado específicamente por el Departamento de Defensa de los Estados Unidos de Norteamérica para dar apoyo a la planeación, programación y control de una gran cantidad de trabajos o actividades asociados con el proyecto.

Actualmente PERT se usa en empresas industriales, de la construcción y en general para programación de mantenimiento en aviones, activos fijos, diseño de plantas, así como la planeación y la administración de programas en investigación y desarrollo.

Una de las principales características de PERT, además de su capacidad para identificar programas y planes que se requieren para las actividades, es que puede manejar las incertidumbres que existen en los pronósticos de tiempos para determinar diversas tareas.

Por otra parte, CPM fue desarrollado en forma independiente de PERT, pero está estrechamente relacionado con este y se refiere básicamente a los intercambios entre el costo de un proyecto y su fecha de terminación. Se aboca a la reducción del tiempo necesario para concluir una tarea o actividad, utilizando más trabajadores y/o recursos, lo que en

la mayoría de los casos significa mayores costos. Con CPM, se supone que el tiempo necesario para concluir las diversas actividades del proyecto se conoce con certidumbre al igual que la cantidad de recursos que se utilizan.

Por ello CPM no se ocupa de tiempos inciertos de actividades como en el caso de PERT, sino que se refiere a intercambios entre tiempos y costos. Pese a estas diferencias en la actualidad ha desaparecido en gran medida la distinción entre uso de PERT y CPM. La mayoría de las versiones computarizadas de las técnicas incluyen opciones para manejar incertidumbre en los tiempos de las actividades, así como también análisis de intercambios de tiempos y costos, y gran parte de la literatura actual se refiere a esta técnica como PERT/CPM.

## 7.10 EJERCICIOS PROPUESTOS

1. Se ha establecido que un proyecto tiene la siguiente lista de actividades y los correspondientes tiempos para terminarlas:

Actividad	Tiempo (días)	Predecesores inmediatos
A	1	-
B	4	A
C	3	A
D	7	A
E	6	B
F	2	C, D
G	7	E, F
H	9	D
I	4	G, H

- a) Dibuje el diagrama de la ruta crítica.
- b) Marque los tiempos de inicio y final más próximos.
- c) Marque la ruta crítica.
- d) ¿Qué pasaría si se modificara la actividad F de modo que tomara cuatro días en lugar de dos?

2. Se ha establecido que un proyecto tiene las siguientes actividades y tiempos estimados para terminarlas.

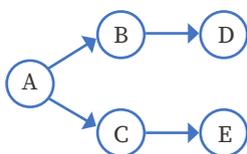
Actividad	Tiempos estimados (semanas)			Precedente inmediato
	a	m	b	
A	1	4	7	-
B	2	6	7	A
C	3	4	6	A, D
D	6	12	14	A
E	3	6	12	D
F	6	8	16	B, C
G	1	5	6	E, F

- a) Calcule el tiempo esperado y la varianza para cada actividad.
  - b) Dibuje el diagrama de la ruta crítica.
  - c) Marque el tiempo de inicio y final más próximo y de inicio y final más lejanos.
  - d) Marque la ruta crítica.
  - e) ¿Qué probabilidad existe de que el proyecto quede concluido en 34 semanas?
3. A continuación se presentan los requerimientos de precedencia, los tiempos normales y los intensivos de las actividades, y los costos normales y los intensivos de un proyecto de construcción:

Actividad	Actividades Precedentes	Tiempo requerido (semanas)		Costo	
		Normal	Intensivo	Normal	Intensivo
A	-	4	2	10 000	11 000
B	A	3	2	6 000	9 000
C	A	2	1	4 000	6 000
D	B	5	3	14 000	18 000
E	B, C	1	1	9 000	9 000
F	C	3	2	7 000	8 000
G	E, F	4	2	13 000	25 000
H	D, E	4	1	11 000	18 000
I	H, G	6	5	20 000	29 000

- a) ¿Cuáles son la ruta crítica y los tiempos estimados para la conclusión?
  - b) Para reducir el proyecto tres semanas, ¿cuáles tareas se deben acortar y cuál sería el costo total final del proyecto?
4. Existe una probabilidad de 82% de que el proyecto siguiente se pueda terminar en  $X$  semanas o menos. ¿Qué valor tiene  $X$ ?

Actividad	Más optimista	Más probable	Más pesimista
A	2	5	11
B	3	3	3
C	1	3	5
D	6	8	10
E	4	7	10



5. Se ha establecido que un proyecto contiene las actividades siguientes, inclusive los 15. tiempos estimados para terminarlo.

Actividad	Tiempos estimados (semanas)			Precedente inmediato
	a	m	b	
A	2	5	8	-
B	1	5	9	-
C	4	6	9	A
D	2	2	2	B
E	1	2	9	A
F	2	4	5	C, D
G	3	8	10	C, E
H	1	2	3	F, G

- Calcule el valor esperado y la varianza para cada actividad.
- Dibuje el diagrama de la ruta crítica. Muestre los tiempos de inicio y final adelantados, y los tiempos de inicio y final retrasados.
- Muestre la ruta crítica.
- ¿Cuál es la probabilidad de que el proyecto quede terminado en 19 semanas?

# Problema de flujo a costo mínimo

## 8.1 INTRODUCCIÓN

En este capítulo se analiza el problema de una red de flujo donde se tienen capacidades en los arcos y se desea enviar flujo a costo mínimo.

El problema del flujo de costo mínimo es el caso general de los problemas de redes vistos en este libro. Igual que el problema de flujo máximo, toma en cuenta un flujo en la red con capacidades en los arcos. Igual que el problema de la ruta más corta, considera un costo (o distancia) del flujo a través de un arco. Igual que el problema de transporte o el de asignación puede manejar varios nodos origen y varios destinos del flujo, de nuevo con costos asociados.<sup>1</sup> Ya hemos considerado en capítulos anteriores dos casos especiales de este problema: el más breve el problema de la trayectoria y el problema del flujo máximo. Nuestra discusión ha sido multifacética:

---

<sup>1</sup> Hillier y Lieberman, “Modelos de optimización en Redes”, c.9, *Introducción a la Investigación de Operaciones*. 9ª. ed. Mc. Graw- Hill, 2010.

1. Hemos visto cómo surgen estos problemas en entornos de aplicación tan diversos como reemplazo de equipos, planificación de proyectos, y programación de producción.
2. Hemos desarrollado una serie de algoritmos y enfoques para resolver estos problemas y su complejidad computacional; problemas de optimización combinatoria tales como el problema de corte mínimo.

Como hemos visto, es fácil entender la naturaleza básica de la ruta más corta y los problemas de flujo máximo y desarrollar algoritmos para resolverlos; sin embargo, diseñar y analizar algoritmos eficientes es una tarea muy desafiante, que requiere considerable ingenio e información sobre las estrategias algorítmicas básicas y sus implementaciones.

### **Algunas aplicaciones del modelo de flujo a costo mínimo<sup>2</sup>**

Una de las aplicaciones más importantes de este problema es en la operación de la red de distribución de una compañía, donde se busca determinar un plan para enviar bienes desde las fuentes (fábricas, centros de distribución, etc.) a las instalaciones intermedias de almacenamiento (según se necesite) y finalmente a los clientes, como se resume en el primer renglón de la tabla 8.1. Como los nodos varían de acuerdo con las aplicaciones en la siguiente tabla 8.1 se presentan aplicaciones comunes del PFCM (problema de flujo a costo mínimo).

---

<sup>2</sup> *Ibidem.*

En el caso de la segunda aplicación, los nodos son fuentes de desechos sólidos y el flujo de materiales a través de la red comienza en las fuentes de desechos sólidos, luego va a las instalaciones para procesar estos materiales de desecho y convertirlos de forma adecuada para el relleno y de ahí enviarlos a los diferentes sitios de relleno. El objetivo es determinar el plan que minimice el costo total, donde el costo se refiere al embarque y al procesamiento.

**Tabla 8.1** Aplicaciones comunes del PFCM

Tipo de aplicación	Nodos fuente	Nodos de transbordo	Nodos demanda
Operación de una red de distribución	Fuentes de bienes	Almacenes intermedios	Clientes
Administración de desechos sólidos	Fuentes de desechos sólidos	Instalaciones de procesamiento	Rellenos
Operación de una red de suministros	Agentes de ventas	Almacenes intermedios	Instalaciones de procesamiento
Coordinación de mezclas de productos en plantas	Plantas	Producción de un artículo específico	Mercado del producto específico
Administración de flujo de efectivo	Fuentes de efectivo en tiempos específicos	Opciones de inversión a corto plazo	Necesidad de efectivo en tiempos específicos

En el tercer renglón de la tabla, el objetivo es encontrar el plan de costo mínimo para obtener suministros de varios proveedores posibles, almacenar los bienes (según se requiera), y luego enviar los suministros a las instalaciones de procesamiento de la compañía, como pueden ser fábricas entre otros. Como la cantidad total que pueden suministrar los proveedores es mayor que las necesidades de la compañía, la red incluye un nodo de demanda ficticio que sin costo recibe toda la capacidad usada de los proveedores.

El cuarto renglón se refiere a la coordinación de mezclas de productos en las plantas, ilustra que los arcos pueden representar algo distinto a un canal de envío de un flujo físico de materiales. Esta aplicación se refiere a una compañía con varias plantas (nodos fuente) que pueden producir los mismos productos, pero con diferentes costos. Cada arco que sale de un nodo fuente representa la producción de uno de los productos posibles de esa planta, donde el arco conduce a un nodo de transbordo que corresponde al producto. En consecuencia, este nodo de transbordo tiene un arco que llega desde cada planta capaz de producir ese producto, y después los arcos que salen de ese nodo van a los respectivos clientes (nodos de demanda). El objetivo es determinar cómo dividir la capacidad de producción de cada planta entre los productos para minimizar el costo total de satisfacer las demandas de los distintos productos.

Finalmente, en el quinto renglón, la administración de flujo de efectivo ejemplifica que diferentes nodos pueden representar un evento que ocurre en distintos momentos. En este caso cada nodo fuente representa un tiempo específico (periodo) en que la compañía dispone de dinero (por inversiones, cuentas por cobrar, venta de acciones, préstamos). El suministro de cada nodo es el dinero que estará disponible. De manera similar, cada nodo destino representa un tiempo específico (periodo) en que la compañía deberá usar su reserva de efectivo. La demanda de cada uno de esos nodos es la cantidad de efectivo que necesitará. El objetivo es maximizar el ingreso de la compañía debido a estas inversiones entre los tiempos en que se dispone de efectivo y se usa. Por lo tanto, los nodos de transbordo representan la opción de una inversión a corto plazo, es decir, la compra de certificados de depósito en un banco por un periodo específico. La red resultante tendrá una serie de flujos que representan un programa de efectivo disponible, invertido y utilizado cuando la inversión madura.

Se verá un algoritmo que es una adecuación del método simplex para redes, aunque existen más algoritmos en la práctica el que más se usa es el simplex en los problemas de redes o se dividen estos si son muy grandes en módulos que resuelven el problema por partes. El problema de flujo a costo mínimo generaliza el modelo de flujo máximo en 4 aspectos:

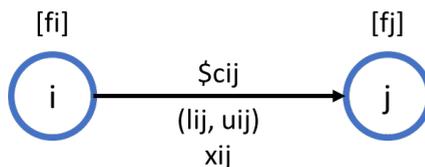
1. Todos los arcos son dirigidos.
2. Un flujo de costo unitario está asociado con cada arco.
3. Los arcos pueden tener capacidades mínimas y máximas.
4. Cualquier nodo en la red puede actuar como nodo fuente (con oferta) o como nodo destino (con demanda).

El objetivo de este modelo es determinar los flujos en los diferentes arcos que minimizan el costo total mientras satisfacen las restricciones de flujo en los arcos y las cantidades de oferta y demanda en los nodos.

Sea la red con restricciones de capacidad  $G = (N, A)$  donde  $N$  es el conjunto de nodos y  $A$  el conjunto de arcos.

- $x_{ij}$  = La cantidad de flujo del nodo  $i$  al nodo  $j$   
 $u_{ij}(l_{ij})$  = Capacidad superior (inferior) del arco  $(i, j)$   
 $c_{ij}$  = Unidad de costo del flujo del nodo  $i$  al nodo  $j$   
 $b_i$  = Flujo neto en el nodo  $i$

Esto se puede ver en la red de la figura siguiente:



**Figura 8.1** Red con restricciones de capacidad en arcos

El valor de  $b_i$ , depende de la naturaleza del nodo  $i$ , donde supone un valor positivo, si  $i$  es un nodo fuente. Negativo si  $i$  es un nodo de demanda, y cero si es un nodo de transbordo.

### Ejemplo 8.1

Grupo Industrial Maseca debe distribuir maíz de tres silos a tres centros de acopio. Las cantidades de oferta en los tres silos son 100, 200 y 50 toneladas y la demanda de las tres granjas es de 150, 80 y 120 toneladas. Maseca usa principalmente ferrocarril para transportar el maíz a las granjas con excepción de tres rutas que usan tráileres. La siguiente red muestra la problemática:

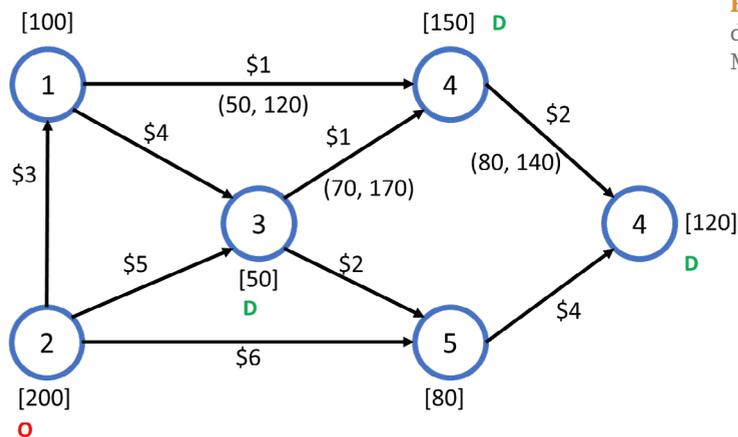


Figura 8.2 Red de distribución de Maseca

Los nodos 1, 2 y 3 representan los silos con sus respectivas ofertas. Los centros están representados por los nodos 4, 5 y 6 cuyas demandas están representadas por [150], [80] y [120]. Las rutas permiten transporte entre los silos. Las rutas de los tráileres están representadas por (1, 4), (3, 4) y (4, 6). Las capacidades en las rutas están dadas en términos de las toneladas a transportar (asociadas a la capacidad de los tráileres); por ejemplo, en la ruta (1, 4) se tiene entre 50 y 120 toneladas.

Las otras rutas que usan tren tienen capacidad ilimitada. Los costos de transportación por tonelada están indicados en los arcos.

La solución óptima será el flujo a costo mínimo en los diferentes arcos que satisfagan las restricciones de flujo en el modelo.

## 8.2 FORMULACIÓN DE PROGRAMACIÓN LINEAL

Dado que los problemas de flujo de costo mínimo son programas lineales, no es sorprendente descubrir que también podemos usar metodologías de programación lineal para resolver problemas de flujo a costo mínimo.

El modelo de la red con capacidad limitada se puede formular como un modelo de programación lineal.

$$\min z = \sum_{(i,j) \in A} \sum c_{ij} x_{ij}$$

Sujeto a

$$\sum_{(i,j) \in A} x_{jk} - \sum_{(i,j) \in A} x_{ij} = b_j \quad j \in N$$

$$l_{ij} \leq x_{ij} \leq u_{ij}$$

La ecuación para el nodo  $j$  mide el flujo neto  $b_j$  en el nodo  $j$  como:

$$(\text{flujo que sale del nodo } j) - (\text{flujo que entra al nodo } j) = b_j$$

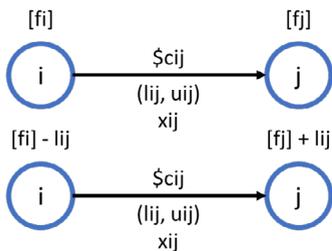
El nodo  $j$  es un nodo fuente si  $b_j > 0$  y es sumidero si  $b_j < 0$ .

Podemos quitar la cota inferior  $l_{ij}$  de las restricciones usando la sustitución:

$$x_{ij} = x'_{ij} + l_{ij}$$

Ahora tratamos con  $x'_{ij}$  como una nueva variable de flujo cuyo límite superior es  $u_{ij} - l_{ij}$ . Adicionalmente, el flujo neto en el nodo  $i$  se convierte  $[f_i] - l_{ij}$  mientras que el nodo  $j$  es  $[f_i] + l_{ij}$ .

Esto lo podemos ver en la figura siguiente:



**Figura 8.3**  
Representación de las cotas en los arcos

Antes de continuar consideramos las siguientes suposiciones:

**Supuesto 8.1.**

*Todos los datos (costo, oferta/demanda y capacidad) son enteros.*

Como se señaló anteriormente, esta suposición no es realmente restrictiva en la práctica porque las computadoras trabajan con números racionales que podemos convertir a números enteros multiplicando por un número adecuadamente grande.

**Supuesto 8.2.**

*La red es dirigida.*

Hemos demostrado en capítulos anteriores que siempre podemos cumplir esta suposición al transformar cualquier red no dirigida en una red dirigida, al reemplazar los arcos apropiadamente.

**Supuesto 8.3.**

Las ofertas/demandas en los nodos satisfacen la condición  $\sum_{i \in N} b(i) = 0$  y el problema de flujo de costo mínimo tiene una solución factible.

Podemos determinar si el problema de flujo de costo mínimo tiene una solución factible resolviendo un problema de flujo máximo de la siguiente manera. Introducir un nodo fuente  $s^*$  y un nodo sumidero  $t^*$ . Para cada nodo  $i$  con  $b(i) > 0$ , agregue un arco "fuente"  $(s^*, i)$  con capacidad  $b(i)$ , y para cada nodo  $i$  con  $b(i) < 0$ , agregue un arco "sumidero"  $(i, t^*)$  con capacidad  $-b(i)$ . Ahora resuelva un problema de flujo máximo de  $s^*$  a  $t^*$ . Si el flujo máximo satura todos los arcos fuente, el problema de flujo de costo mínimo es factible; de lo contrario, no es factible.

**Supuesto 8.4.**

Suponemos que la red  $G$  contiene una trayectoria dirigida no capacitada (es decir, cada arco en la trayectoria tiene una capacidad infinita) entre cada par de nodos.

Imponemos esta condición, si es necesario, agregando arcos artificiales  $(l, j)$  y  $(j, 1)$  para cada  $j \in N$  y se asigna un gran costo y una capacidad infinita a cada uno de estos arcos. Tal arco no aparecería en una solución de costo mínimo a menos que el problema no contenga ninguna solución factible sin arcos artificiales.

**Supuesto 8.5.**

Todos los costos del arco son no negativos.

Esta suposición no impone pérdida de generalidad ya que la transformación de inversión de un arco convierte un problema de flujo de costo mínimo con costos negativos a aquellos con costos no negativos. Esta transformación, sin embargo, requiere que todos los arcos tengan capacidades finitas. Cuando algunos arcos están incapacitados, suponga que la red no contiene un ciclo de costo negativo dirigido de

capacidad infinita. Si la red contiene tales ciclos, el valor óptimo del flujo de costo mínimo es ilimitado. En ausencia de un ciclo negativo con capacidad infinita, podemos hacer que cada arco no capacitado esté capacitado al establecer su capacidad igual a  $B$ , donde  $B$  es la suma de todas las capacidades de los arcos y los suministros de todos los nodos de suministro.

Una vez que hemos establecido estos supuestos, continuamos con el programa lineal para el ejemplo 8.1, antes y después que las cotas inferiores se han sustituido.

$$\text{Min } z = 3x_{21} + 4x_{13} + x_{14} + 5x_{23} + 6x_{25} + x_{34} + 2x_{35} + 2x_{46} + 4x_{56}$$

Sujeto a:

nodo 1	$-x_{21}$	$+x_{13}$	$+x_{14}$							= 100
nodo 2	$x_{21}$			$+x_{23}$	$+x_{25}$					= 200
nodo 3		$-x_{13}$		$-x_{23}$		$+x_{34}$	$+x_{35}$			= 50
nodo 4			$x_{14}$			$+x_{34}$		$-x_{46}$		= 150
nodo 5					$+x_{25}$		$+x_{35}$		$-x_{56}$	= 80
nodo 6								$x_{46}$	$+x_{56}$	= 120

Con cotas inferior y superior:

$$50 \leq x_{14} \leq 120$$

$$70 \leq x_{34} \leq 170$$

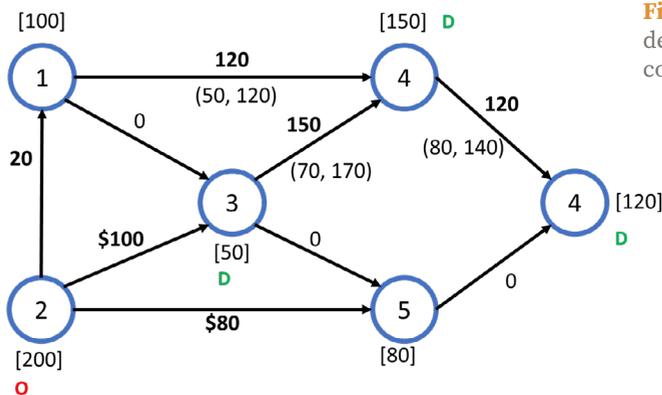
$$80 \leq x_{46} \leq 140$$

La solución usando SOLVER o el software LINDO es:

OBJECTIVE FUNCTION VALUE	
1) 1550.000	
VARIABLE	VALUE
X14	120.000000
X13	0.000000
X21	20.000000
X23	100.000000
X25	80.000000
X34	150.000000
X35	0.000000
X46	120.000000
X56	0.000000

**Figura 8.4**  
Solución con LINDO

Por lo tanto, el plan de envío más económico se muestra en la red siguiente:



**Figura 8.5** Red de distribución a costo mínimo

## 8.3 ALGORITMO SIMPLEX PARA REDES CON CAPACIDADES EN LOS ARCOS

El algoritmo simplex para redes con capacidades en los arcos está basado en los pasos del método simplex regular. El “nuevo” método, sin embargo, explota la estructura especial del modelo de costo mínimo de la red.

Dado  $b_i$ , el flujo neto en el nodo  $i$  se define en el programa lineal como en la sección 8.2, el algoritmo simplex para la red capacitada estipula que la red debe satisfacer la condición

$$\sum_{i=1}^n b_i = 0$$

Esta condición dice que la oferta total en la red debe ser igual a la demanda total. Podemos satisfacer siempre esta condición aumentando un nodo destino, conectado a todos los otros nodos en la red y que tenga en los arcos un costo cero, y capacidades ilimitadas. Note que el balanceo de la red no garantiza que el modelo tendrá una solución factible, por lo que es una condición necesaria pero no suficiente para que haya soluciones factibles.

### Incorporación de la técnica de la cota superior<sup>3</sup>

El primer concepto es incorporar la técnica de la cota superior para manejar con eficiencia las restricciones de capacidad en los arcos. De esta manera, en lugar de tratar estas restricciones como funcionales

---

<sup>3</sup> Hillier y Lieberman, “Modelos de optimización en Redes”, c.9, *Introducción a la Investigación de Operaciones*. 9ª. ed. Mc. Graw- Hill, 2010.

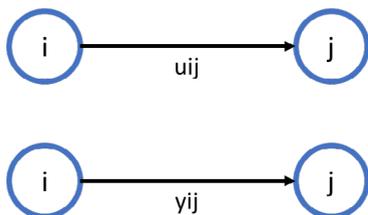
se manejan como restricciones de no negatividad, por lo tanto, solo se toman en cuenta para determinar la variable básica que sale. En particular, a medida que se hace crecer la variable básica entrante desde cero, la variable básica que sale es la primera variable básica que llega a su cota inferior (0) o a su cota superior ( $u_{ij}$ ).

Una variable no básica en su cota superior  $x_{ij} = u_{ij}$ , se reemplaza por  $x_{ij} = u_{ij} - y_{ij}$ , de manera que  $y_{ij} = 0$  se convierte en la variable no básica.

En el contexto de redes  $y_{ij}$  tiene una interpretación en los arcos, ya que siempre que  $y_{ij}$  se convierta en una variable básica con un valor estrictamente positivo pero menor o igual a  $u_{ij}$ , se puede pensar en este valor como un flujo del nodo  $j$  al nodo  $i$ , es decir en la dirección contraria al arco  $(i, j)$ , lo que en realidad cancela la cantidad de flujo previamente asignada ( $x_{ij} = u_{ij}$ ) del nodo  $i$  al nodo  $j$ .

Así cuando  $x_{ij} = u_{ij}$  se reemplaza con  $x_{ij} = u_{ij} - y_{ij}$  también se reemplaza el arco real  $(i, j)$  por el inverso  $(j, i)$  donde este nuevo arco tiene una capacidad  $u_{ij}$  (la cantidad máxima del flujo  $x_{ij} = u_{ij}$  que se puede cancelar) y costo unitario  $-c_{ij}$  (puesto que cada unidad de flujo cancelada ahorra  $c_{ij}$ ).

Para reflejar el flujo  $x_{ij} = u_{ij}$  a través del arco eliminado, se cambia esta cantidad de flujo neto generada del nodo  $i$  al nodo  $j$  al disminuir  $b_i$  en  $u_{ij}$  unidades e incrementar  $b_j$  en  $u_{ij}$  unidades.



**Figura 8.6**

Interpretación del cambio de variable entrante en los arcos

Para ilustrar este proceso consideramos el siguiente ejemplo.

## E

## Ejemplo 8.2

La compañía ECO S.A., fabricará el mismo nuevo producto en dos plantas distintas y después tendrá que enviarlo a dos almacenes de distribución, donde cualquiera de las dos fábricas puede abastecer a cualquiera de los dos almacenes. La red de distribución disponible para el envío de este producto se muestra en la figura 8.7, donde  $F1$  y  $F2$  son las dos fábricas,  $A1$  y  $A2$  son los dos almacenes y  $CD$  es el centro de distribución. Las cantidades que deben enviarse desde  $F1$  y  $F2$  se muestran a la izquierda, y las cantidades que deben recibirse en  $A1$  y  $A2$  se presentan a la derecha. Cada flecha representa un canal factible de envío.  $F1$  puede enviar directamente a  $A1$  y tiene tres rutas posibles ( $F1 \rightarrow CD \rightarrow A2$ ,  $F1 \rightarrow F2 \rightarrow CD \rightarrow A2$  y  $F1 \rightarrow A1 \rightarrow A2$ ) para mandar bienes a  $A2$ . La fábrica  $F2$  tiene solo una ruta a  $A2$  ( $F2 \rightarrow CD \rightarrow A2$ ) y una a  $A1$  ( $F2 \rightarrow CD \rightarrow A2 \rightarrow A1$ ). El costo por unidad enviada a través de cada canal se muestra al lado de la flecha. También, junto a  $F1 \rightarrow F2$  y  $CD \rightarrow A2$  se muestran las cantidades máximas que se pueden enviar por estos canales. Los otros canales tienen suficiente capacidad para manejar todo lo que las fábricas pueden enviar.

La decisión que debe tomarse se refiere a qué cantidades enviar a través de cada canal de distribución. El objetivo es minimizar el costo total de envío. El modelo de programación lineal de este ejemplo es el siguiente:

1

2

3

4

5

6

7

8

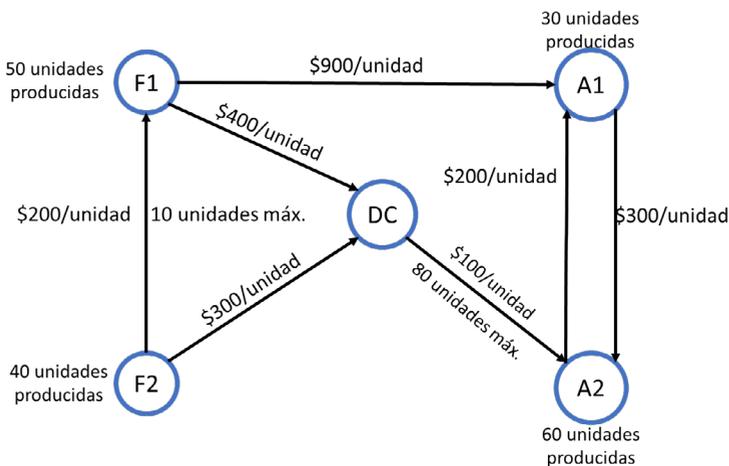
$$\text{Min } z = 2x_{21} + 4x_{13} + 9x_{14} + 3x_{23} + x_{35} + 3x_{45} + 2x_{54}$$

Sujeto a:

	$-x_{21}$	$+x_{13}$	$+x_{14}$					$= 50$
	$x_{21}$			$+x_{23}$	$+x_{25}$			$= 40$
		$-x_{13}$		$-x_{23}$		$+x_{34}$	$+x_{35}$	$= 0$
			$-x_{14}$			$+x_{34}$		$= -30$
					$+x_{25}$		$+x_{35}$	$= -60$

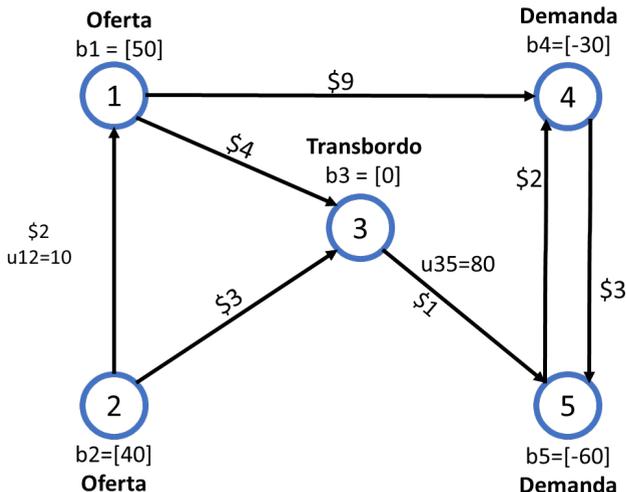
$$x_{12} \leq 10, \quad x_{35} \leq 80, \quad x_{ij} \geq 0$$

Este problema se puede ver en la red 8.7 siguiente:



**Figura 8.7**  
Red de la problemática

En términos de un problema de flujo a costo mínimo reescribimos la red de la siguiente manera:



**Figura 8.8** Red ajustada cuando la técnica de la cota superior conduce a reemplazar  $x_{12} = 10$  por  $x_{12} = 10 - y_{12}$

### Correspondencia entre las soluciones básicas factibles y los árboles de expansión factibles

Como ya hemos visto antes, los árboles de expansión son redes sin circuitos y que en términos de la matriz de incidencia se asocian a una base, lo que conlleva de manera implícita que dicha base corresponde a una solución básica.

En una red con  $n$  nodos, toda solución básica factible tiene  $(n-1)$  variables básicas, y cada variable básica  $x_{ij}$  representa el flujo por el arco  $(i, j)$ . Estos  $(n-1)$  arcos se conocen como arcos básicos. De manera similar, los arcos que corresponden a variables no básicas  $x_{ij} = 0$  o  $y_{ij} = 0$ , se llaman arcos no básicos.

Una propiedad de los arcos básicos es que no forman circuitos, por lo que se puede afirmar que cualquier conjunto de  $n-1$  arcos que no tenga circuitos es un árbol de expansión. En consecuencia, las soluciones básicas factibles se pueden obtener a través de los árboles de expansión. Sin considerar las restricciones de no negatividad ni las

restricciones de capacidad en los arcos para las variables básicas, la solución del árbol de expansión que se obtiene puede o no ser factible respecto a estas restricciones, lo que conduce a la siguiente definición:

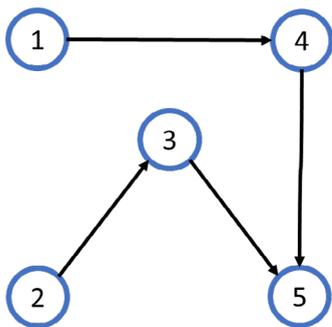
### Definición 8.1

Un árbol de expansión factible es un árbol de expansión cuya solución, a partir de las restricciones de nodos, también satisface todas las demás restricciones,  $0 \leq x_{ij} \leq u_{ij}$  o  $0 \leq y_{ij} \leq u_{ij}$ .

### Teorema fundamental del método simplex para redes

Las soluciones básicas son soluciones del árbol de expansión (y viceversa) y las soluciones básicas factibles son soluciones de árboles de expansión factibles (y viceversa).

Para ilustrar la aplicación del teorema fundamental consideramos la red del ejemplo 8.2 en la figura 8.8, que resulta de sustituir  $x_{12} = 10$  por  $x_{12} = 10 - y_{12}$  de la figura 8.7. Se puede iniciar con un árbol de expansión como el que se muestra en la figura 8.9



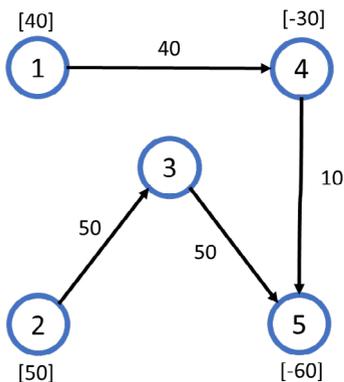
**Figura 8.9** Árbol de expansión para el ejemplo 8.2

Con estos arcos como arcos básicos a continuación se muestra el proceso de encontrar un árbol de expansión factible.

Remitiéndonos al planteamiento de programación lineal, en el lado izquierdo se encuentran las restricciones de los nodos después de sustituir  $x_{12}$  por  $10 - y_{12}$  en donde las variables básicas aparecen en **negritas**. En el lado derecho, de arriba hacia abajo se encuentran los pasos para calcular los valores de las variables.

						$y_{12}=0,$	$x_{13}=0,$	$x_{54}=0$
$-y_{12}$	$+x_{13}$	$+x_{14}$				$= 40$		$x_{14}=40$
$y_{12}$			$+x_{23}$			$= 50$		$x_{23}=50$
	$-x_{13}$		$-x_{23}$	$+x_{35}$		$= 0$	así	$x_{35}=50$
		$-x_{14}$			$+x_{45}$	$-x_{54}$	$= -30$	así $x_{45}=10$
			$-x_{35}$	$-x_{45}$	$+x_{54}$	$= -60$	Redundante	

Ya que los valores de todas estas variables satisfacen las restricciones de no negatividad y la única restricción relevante de capacidad de arco ( $x_{35} \leq 80$ ), el árbol de expansión es factible, por lo que se tiene una solución básica inicial, se usará esta solución para ilustrar el método simplex para redes. En la figura 8.10 se muestra este árbol en donde los números junto a los arcos representan los flujos  $x_{ij}$  en lugar de los costos unitarios  $c_{ij}$ .



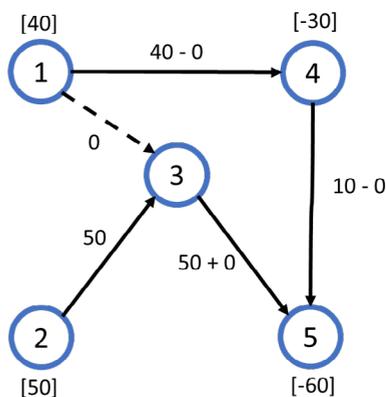
**Figura 8.10** Árbol de expansión factible inicial y su solución

## Selección de la variable no básica entrante

Para comenzar una iteración del método simplex de redes, recordemos que el criterio del simplex estándar para elegir la variable que entra se escoge la variable no básica que, al aumentar su valor, mejore el valor de  $z$  a la tasa más rápida, entonces veamos cómo se hace este proceso sin una tabla simplex.

Consideramos la variable no básica  $x_{13}$  de la solución básica inicial, lo que significa elegir el arco no básico (1, 3). Aumentar  $x_{13}$  desde cero hasta algún valor  $\theta$ , es equivalente a agregar el arco (1, 3) con un flujo  $\theta$  como se muestra en la figura 8.11.

Como se observa en la red al agregar un arco se forma un circuito sin importar la dirección de los arcos, y al agregar flujo en el arco (1, 3) el flujo en los demás arcos del circuito se ve afectado por este valor  $\theta$ . En particular el flujo aumenta por  $\theta$  en los arcos que se recorren en el sentido en el que están y disminuye si los arcos se recorren en sentido contrario.

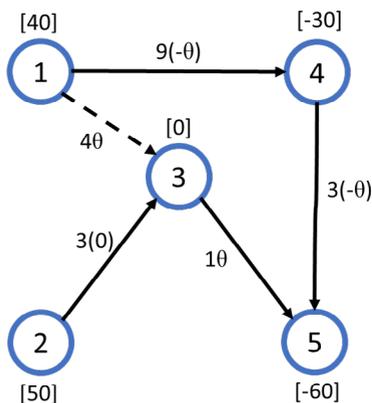


**Figura 8.11** Efecto sobre los flujos al agregar el arco (1, 3) con un flujo  $\theta$

Ahora, ¿cuál es el efecto incremental sobre  $z$  (costo total del flujo) al agregar el flujo  $\theta$  al arco (1, 3)? En la figura 8.12 se muestra la mayor

parte de la respuesta, pues se proporciona el costo unitario multiplicado por el cambio de flujo de cada arco de la figura 8.11, por lo tanto, el incremento global en  $z$  es:

$$\Delta z = c_{13}\theta + c_{35}\theta + c_{45}(-\theta) + c_{14}(-\theta) = 4\theta + \theta - 3\theta - 9\theta = -7\theta$$



**Figura 8.12** Efecto incremental sobre los costos al agregar el arco (1, 3) con flujo  $\theta$

Si se establece  $\theta = 1$  se obtiene la tasa de cambio de  $z$  a medida que  $x_{13}$  se incrementa, es decir:

$$\Delta z = -7 \text{ cuando } \theta = 1.$$

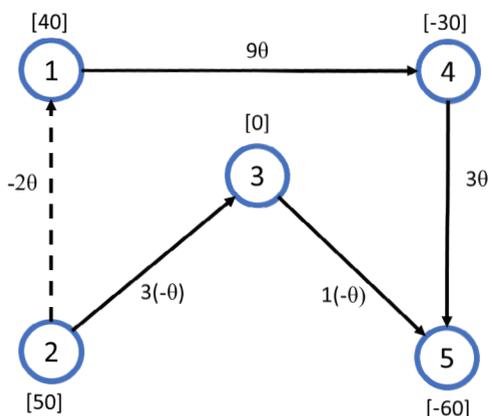
Como el objetivo es minimizar  $z$ , esta tasa grande de disminución de  $z$  al aumentar  $x_{13}$  es muy deseable, por lo tanto,  $x_{13}$  se convierte en el mejor candidato para ser la variable no básica entrante, sin embargo, es necesario realizar el mismo análisis con las otras variables no básicas antes de hacer la selección final, y estas son  $y_{12}$  y  $x_{54}$  que corresponden a los dos arcos no básicos (1, 2) y (5, 4) de la figura 8.8.

En la figura 8.13 se muestra el efecto incremental sobre los costos al agregar el arco (1, 2) con un flujo  $\theta$  al árbol de expansión factible inicial de la figura 8.10. Si se agrega este arco se crea un ciclo no dirigido

$2 \rightarrow 1 \rightarrow 4 \rightarrow 5 \leftarrow 3 \leftarrow 2$ , con lo que el flujo aumenta en  $\theta$  para los arcos  $(1, 4)$  y  $(4, 5)$  y disminuye en los arcos que se recorren en la dirección opuesta  $(3, 5)$  y  $(2, 3)$ . Estos incrementos de flujo se multiplican por los costos  $c_{ij}$  asociados a los arcos, por lo tanto:

$$\begin{aligned} \Delta z &= c_{21}\theta + c_{14}\theta + c_{45}\theta + c_{35}(-\theta) + c_{23}(-\theta) = \\ &= -2\theta + 9\theta + 3\theta + 1(-\theta) + 3(-\theta) = 6\theta \end{aligned}$$

Si  $\theta = 1$  entonces  $\Delta z = 6$



**Figura 8.13** Efecto incremental sobre los costos al agregar el arco  $(2, 1)$  con flujo  $\theta$

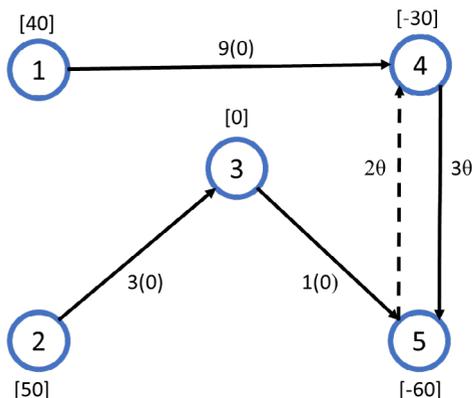
El hecho de que  $z$  aumente en lugar de disminuir cuando  $y_{12}$  (flujo en sentido contrario a través del arco  $(1, 2)$ ) elimina a esta variable como candidata a ser la variable no básica entrante. Recordemos que al aumentar el valor de  $y_{12}$  desde cero significa disminuir  $x_{12}$ , es decir el flujo a través del arco  $(1, 2)$  desde su cota superior que es 10.

En el caso del último arco no básico  $(5, 4)$  se obtiene un resultado similar, al agregar este arco se crea un ciclo no dirigido  $4 \rightarrow 5 \rightarrow 4$  como se muestra en la figura 8.14, en este caso el flujo se aumenta por  $\theta$  en ambos arcos como se muestra a continuación:

$$\Delta z = c_{45}\theta + c_{54}\theta = 2\theta + 3\theta = 5\theta = 5 \text{ cuando } \theta = 1$$

Por lo que  $x_{54}$  se elimina como candidato a variable básica entrante.  
 En resumen:

$$\Delta z = \begin{cases} -7 & \text{si } \Delta x_{13} = 1 \\ 6 & \text{si } \Delta y_{12} = 1 \\ 5 & \text{si } \Delta x_{54} = 1 \end{cases}$$



**Figura 8.14** Efecto incremental sobre los costos al agregar el arco (5, 4) con flujo  $\theta$

En consecuencia, el valor negativo de  $x_{13}$  implica que  $x_{13}$  se convierte en la variable básica entrante en la primera iteración. En caso de que haya más de una variable no básica con un valor negativo de  $\Delta z$ , se elige la de mayor valor negativo, si no hay otras variables no básicas con un valor negativo de  $\Delta z$  la solución básica factible actual es óptima.

### Selección de la variable básica saliente y la siguiente solución básica factible

Una vez que se ha elegido la variable no básica entrante, el siguiente paso es determinar la variable básica saliente y así obtener la próxima solución básica factible.

Ya vimos que  $x_{13}$  es la variable básica entrante, se aumenta el flujo  $\theta$  a través del arco (1, 3) lo más posible, hasta que una de las variables básicas llegue a su cota inferior cero o a su cota superior  $u_{ij}$ . Para aquellos arcos cuyo flujo aumenta junto con  $\theta$ , es decir los arcos (1, 3) y (3, 5) solo es necesario considerar sus cotas superiores que son respectivamente:  $u_{13} = \infty$  y  $u_{35} = 80$ :

$$\begin{aligned} x_{13} = \theta &\leq \infty \\ x_{35} = 50 + \theta &\leq 80, \quad \text{entonces } \theta \leq 30 \end{aligned}$$

En el caso de los arcos cuyo flujo disminuye con  $\theta$ , los arcos (4, 5) y (1, 4) solo se tiene que considerar la cota inferior cero:

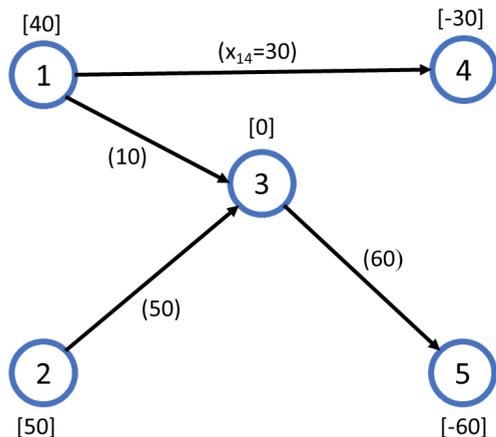
$$\begin{aligned} x_{45} = 10 - \theta &\geq 0, \quad \text{entonces } \theta \leq 10 \\ x_{14} = 40 - \theta &\geq 0, \quad \text{entonces } \theta \leq 40 \end{aligned}$$

Los arcos cuyo flujo no cambia con  $\theta$  porque no forman parte del ciclo, no se afectan y no se analizan.

Para los 5 arcos de la figura 8.11 se concluye que  $x_{45}$  debe ser la variable básica que sale, puesto que alcanza una cota para el valor más pequeño de  $\theta$  que es 10. Con  $\theta = 10$  se obtienen los siguientes valores:

$$\begin{aligned} x_{13} = \theta &= 10 \\ x_{35} = 50 + \theta &= 60 \\ x_{14} = 40 - \theta &= 30 \\ x_{23} &= 50 \end{aligned}$$

Con estos valores se obtiene el nuevo árbol de expansión mínima y nueva solución básica factible como se muestra en la figura 8.15.



**Figura 8.15**  
Segundo árbol de expansión factible

Si la variable básica que sale hubiera alcanzado su cota superior, entonces se habría necesitado ajustar con la técnica de la cota superior como se verá en las siguientes dos iteraciones, como se alcanzó la cota inferior cero, no necesita hacerse nada más.

Iteración 2

A partir del árbol de expansión factible de la figura 8.15 y los costos unitarios  $c_{ij}$  (mostrados en la figura 8.8) se llega a los cálculos para seleccionar la variable no básica entrante que se presentan en la tabla 8.2.

En las columnas de la tabla se muestran los arcos no básicos, los ciclos que se forman al agregarlos y el valor incremental en  $z$  cuando el valor de  $\theta = 1$

**Tabla 8.2** Cálculos para elegir la variable entrante en la iteración 2

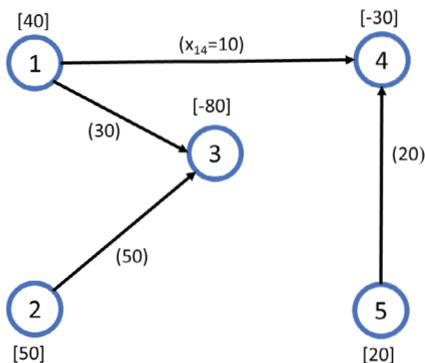
Arco no básico	Ciclo creado	$\Delta z$ cuando $\theta = 1$
(2, 1)	$2 \leftarrow 1 \rightarrow 3 \leftarrow 2$	$-2 + 4 - 3 = -1$
(4, 5)	$4 \rightarrow 5 \leftarrow 3 \leftarrow 1 \rightarrow 4$	$3 - 1 - 4 + 9 = 7$
(5, 4)	$5 \rightarrow 4 \leftarrow 1 \rightarrow 3 \rightarrow 5$	$2 - 9 + 4 + 1 = -2 \leftarrow \text{mínimo}$

La variable no básica entrante es  $x_{54}$ .

Ahora el flujo  $\theta$  en el arco (5, 4) se hace tan grande como sea posible, sin violar las cotas de los arcos sobre los flujos:

$$\begin{array}{ll}
 x_{54} = \theta \leq u_{54} = \infty & \text{entonces } \theta \leq \infty \\
 x_{14} = 30 - \theta \geq 0 & \text{entonces } \theta \leq 30 \\
 x_{13} = 10 + \theta \leq u_{13} = \infty & \text{entonces } \theta \leq \infty \\
 x_{35} = 60 + \theta \leq u_{35} = 80 & \text{entonces } \theta \leq 20 \leftarrow \text{mínimo}
 \end{array}$$

Como  $x_{35}$  impone la menor cota superior (20) sobre  $\theta$ ,  $x_{35}$  se convierte en la variable básica que sale. Se sustituye  $\theta = 20$  en las igualdades anteriores y se obtiene el flujo a través de los arcos básicos de la siguiente solución factible como se muestra en la figura 8.16.

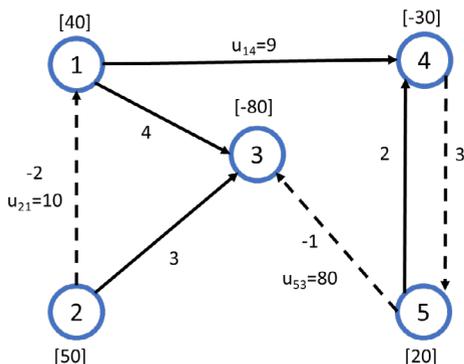


**Figura 8.16**  
Tercer árbol de expansión factible

Es de especial interés en este punto que la variable básica que sale  $x_{35}$  se obtuvo a partir de la variable que alcanzó su cota superior (80). Por lo tanto, mediante la técnica de la cota superior  $x_{35}$  se sustituye por  $80 - y_{35}$ , donde  $y_{35} = 0$  es la nueva variable no básica.

Al mismo tiempo el arco original (3, 5) se sustituye por el arco invertido (5, 3) con  $c_{53} = -1$  y  $u_{53} = 80$ . Los valores de  $b_5$  y  $b_3$  también se ajustan al agregar 80 a  $b_5$  y restar 80 a  $b_3$ , la red ajustada se muestra

en la figura 8.17, donde los arcos no básicos se representan con líneas punteadas, mientras los costos se representan junto a los arcos.



**Figura 8.17** Red ajustada con costos unitarios al terminar la iteración 2

### Iteración 3

Para iniciar esta iteración se usan las figuras 8.16 y 8.17, en la tabla 8.3 se muestran los cálculos que conducen a la elección de  $y_{12}$  el inverso del arco (2, 1) como la variable básica entrante.

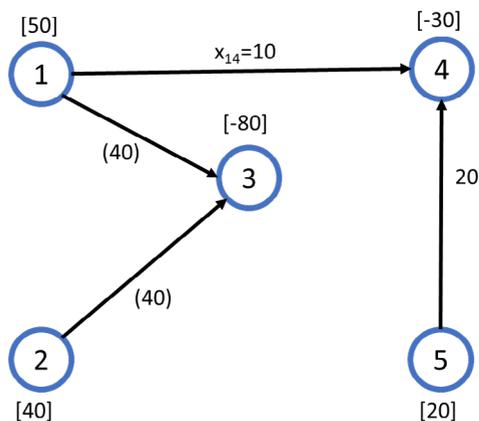
**Tabla 8.3** Cálculos para elegir la variable entrante en la iteración 3

Arco no básico	Ciclo creado	$\Delta z$ cuando $\theta = 1$
(2, 1)	$2 \leftarrow 1 \rightarrow 3 \leftarrow 2$	$-2 + 4 - 3 = -1 \leftarrow$ mínimo
(4, 5)	$4 \rightarrow 5 \rightarrow 4$	$3 + 2 = 5$
(5, 3)	$5 \rightarrow 3 \leftarrow 1 \rightarrow 4 \rightarrow 5$	$-1 - 4 + 9 - 2 = 2$

Ahora se procede a agregar tanto flujo  $\theta$  como sea posible al arco (2, 1) como la variable básica entrante, sin violar las restricciones de los arcos:

$$\begin{aligned}
 y_{12} = \theta &\leq u_{12} = 10 && \text{entonces } \theta \leq 10 \leftarrow \text{mínimo} \\
 x_{13} = 30 + \theta &\leq u_{13} = \infty && \text{entonces } \theta \leq \infty \\
 x_{23} = 50 - \theta &\geq 0 && \text{entonces } \theta \leq 50
 \end{aligned}$$

La variable  $y_{12}$  impone la cota superior más pequeña (10) sobre  $\theta$ , por lo que se convierte en la variable básica que sale. Haciendo  $\theta = 10$  en las ecuaciones anteriores se obtiene la solución básica factible que se muestra en la figura 8.18.

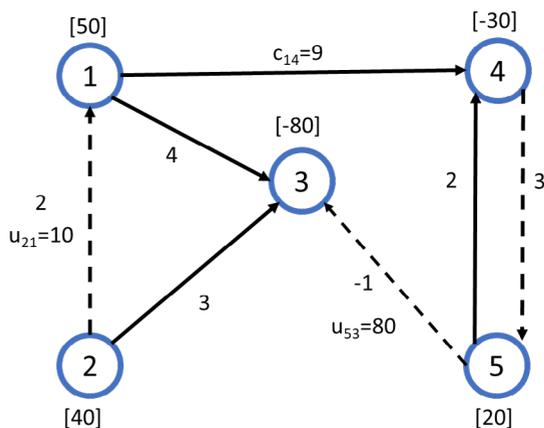


**Figura 8.18**  
Cuarto árbol de expansión factible y final

Igual que en la iteración 2, la variable básica que sale  $y_{12}$  se obtuvo con la variable que alcanzó su cota superior. Existen otros dos puntos de interés especial respecto a esta elección en particular:

1. La variable no básica que entra también se convierte en la variable básica que sale en la misma iteración, en ocasiones este fenómeno ocurre con la técnica de la cota superior, cuando al aumentar el valor de la variable entrante, esta alcanza su cota superior antes que lo haga cualquiera de las variables básicas con su cota.
2. El otro punto de interés es que el arco (2, 1) que ahora es necesario cambiar por su inverso (1, 2) porque la variable básica que sale alcanza una cota superior, ya es el arco inverso. Esto no representa mayor problema ya que el arco inverso del inverso es el original, por lo tanto, el arco (2, 1) con  $c_{21} = -2$  y  $u_{21} = 10$  de la figura 8.17, ahora se sustituye por el arco (1, 2) con  $c_{12} = 2$  y  $u_{12} = 10$  que es el arco de la red original.

Al mismo tiempo, la variable  $y_{12} = 10$  se sustituye por  $10 - x_{12}$ , con  $x_{12} = 0$  como la nueva variable no básica. La red ajustada se muestra en la figura 8.19.



**Figura 8.19** Red ajustada con costos unitarios al terminar la iteración 3

### Prueba de optimalidad

En esta parte el algoritmo usará las figuras 8.18 y 8.19 para encontrar la siguiente variable entrante con los cálculos que se muestran en la tabla 8.4.

**Tabla 8.4** Cálculos para elegir la variable entrante en la iteración 3

Arco no básico	Ciclo creado	$\Delta z$ cuando $\theta = 1$
(1, 2)	$1 \leftarrow 2 \rightarrow 3 \leftarrow 2$	$2 + 3 - 4 = 1$
(4, 5)	$4 \rightarrow 5 \rightarrow 4$	$3 + 2 = 5$
(5, 3)	$5 \rightarrow 3 \leftarrow 1 \rightarrow 4 \rightarrow 5$	$-1 - 4 + 9 - 2 = 2$

Como se puede observar de la tabla 8.4 ninguno de los arcos no básicos da un valor negativo de  $\Delta z$ , por lo que no se logra una mejora del valor de  $z$  por lo que la solución factible actual (figura 8.18) pasa la prueba de optimalidad y el algoritmo se detiene.

Para identificar los flujos a través de los arcos reales y no de los inversos en el caso de esta solución óptima se deben comparar la red original y la red de la figura 8.19.

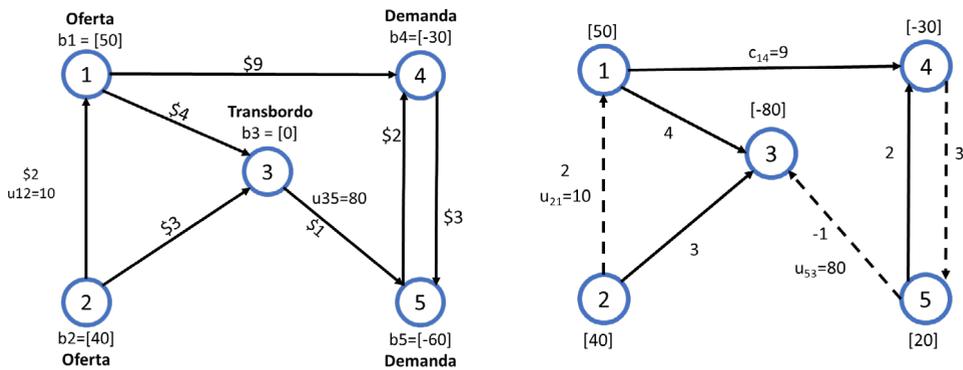


Figura 8.20 Comparación de las redes original y final

Como se puede observar todos los arcos tienen la misma dirección excepto el arco (3, 5), lo que indica que el flujo en este arco está dado por la variable  $y_{35}$ , en consecuencia, se hacen los siguientes cálculos:

$$x_{35} = u_{35} - y_{35} = 80 - y_{35}$$

El arco (5, 3) resulta ser un arco no básico, entonces  $y_{35} = 0$  y  $x_{35} = 80$  es el flujo a través del arco real (3, 5), todos los demás flujos quedan sin modificarse, la solución óptima se muestra en la figura 8.21 siguiente.

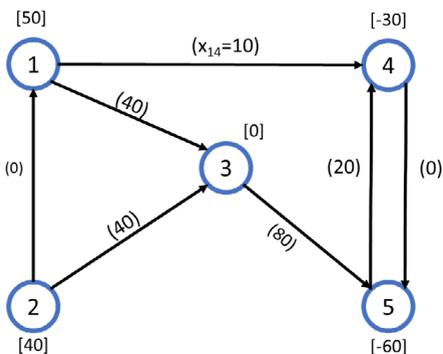


Figura 8.21 Flujo óptimo en la red original

La solución usando el software LINDO con el problema escrito como un modelo de programación lineal se muestra en la figura 8.22

Solución usando LINDO y simplex estándar

OBJECTIVE FUNCTION VALUE		
1) 490.0000		
VARIABLE	VALUE	REDUCED COST
X12	0.000000	1.000000
X13	40.000000	0.000000
X14	10.000000	0.000000
X23	40.000000	0.000000
X35	80.000000	0.000000
X45	0.000000	5.000000
X54	20.000000	0.000000

**Figura 8.22**  
Solución usando método simplex estándar con LINDO

En la siguiente sección se presenta el método simplex basado también en árboles de expansión y el dual simplex.

## 8.4 ALGORITMO SIMPLEX PARA REDES

**Propósito:** Resolver el problema de redes a costo mínimo

### Descripción

**Paso 0** Determine una solución básica factible inicial en los arcos, para la red. Vaya al paso 1.

**Paso 1** Determine un arco (variable) entrante usando la condición de optimalidad del método simplex, si la solución es óptima deténgase, de otra forma vaya al paso 2.

**Paso 2** Determine el arco (variable) saliente usando la condición de factibilidad del método simplex. Cambie la base y vaya al paso 1.

Un nodo  $n$  con flujo neto igual a cero,  $f_1 + f_2 + \dots + f_n = 0$  consiste en  $n - 1$  restricciones independientes. Así, una solución básica asociada debe incluir  $n - 1$  arcos. Recuerde que una solución básica corresponde siempre a un árbol de expansión en la red. Esto significa que el conjunto de arcos que definen una solución básica no puede incluir circuitos.

El arco que entra (paso 1) está determinado al calcular sus coeficientes objetivo  $z_{ij} - c_{ij}$  para todos los arcos no básicos actuales  $(i, j)$ , si se cumple que para toda  $z_{ij} - c_{ij} \leq 0$ , entonces la base actual es óptima. En otro caso, seleccionamos el arco no básico con el más positivo de los  $z_{ij} - c_{ij}$  para entrar a la base.

Los cálculos de  $z_{ij} - c_{ij}$  están basados en el uso de la dualidad exactamente como se hace en el modelo de transporte. Usando el modelo de programación lineal de la sección 8.2 sea  $w_i$  la variable dual asociada con la restricción del nodo  $i$ , entonces el problema dual asociado con las restricciones principales (excluyendo las cotas superiores) está dado por:

$$\max z = \sum_{i=1}^n f_i w_i$$

Sujeto a

$$w_i - w_j \leq c_{ij} \quad (i, j) \in A$$

$$w_i \text{ irrestricta} \quad i = 1, 2, \dots, n$$

De la teoría de la programación lineal tenemos

$$w_i - w_j = c_{ij} \quad \text{para el arco básico } (i, j)$$

Ya que el programa lineal original tiene una restricción redundante por definición, podemos asignar un valor arbitrario a una de las variables duales. Por conveniencia, asumiremos que  $w_1 = 0$ . Podemos entonces resolver las ecuaciones básicas  $w_i - w_j = c_{ij}$  para determinar los valores duales remanentes. Subsecuentemente, podemos calcular  $z_{ij} - c_{ij}$  para las variables no básicas como

$$z_{ij} - c_{ij} = w_i - w_j - c_{ij}$$

El único detalle es como se determina la variable que sale.

E

### Ejemplo 8.3

Una empresa química tiene dos plantas que fabrican una sustancia básica para dos clientes, con capacidades de 660 y 800 toneladas por mes. La capacidad de producción mensual de la planta 1 es de 400 a 800 toneladas, y la de la planta 2 es de 450 a 900 toneladas.

Los costos de producción en las plantas 1 y 2 son \$25 y \$28 por tonelada, respectivamente

La materia prima para las plantas procede de dos proveedores, que tienen contrato para embarcar al menos 500 y 700 toneladas mensua-

1

2

3

4

5

6

7

8

les para las plantas 1 y 2, a costos de \$200 y 210 por tonelada, respectivamente.

La empresa también absorbe el costo del transporte tanto de la materia prima como del compuesto terminado. Los costos de transporte de la materia prima del proveedor 1 a las plantas 1 y 2 son de \$ 10 y \$ 12 por tonelada, los del proveedor 2 son de \$ 9 y 13, respectivamente.

Los costos de transporte de la planta 1 a los clientes 1 y 2 son \$ 3 y \$ 4 por tonelada, respectivamente, y de la planta 2, los costos son \$ 5 y \$ 2, respectivamente. Suponiendo que 1 tonelada de materia prima produce 1 tonelada del compuesto final, formule el problema como modelo de red y resuelva.

La información anterior la podemos resumir en la siguiente tabla:

**Tabla 8.5** Información resumida del ejemplo 8.3

	Cap. de Producción	Costo de prod.	Mat. prima × ton.	Costo de materia prima	Proveedor 1. Costo transporte	Proveedor 2. Costo transporte	Cliente 1 Costo transporte	Cliente 2 Costo transporte
Planta 1	400-800	\$ 25	500	\$ 200	\$ 10	\$ 3	\$ 3	\$ 4
Planta 2	450-900	\$ 28	700	\$ 210	\$ 12	\$ 4	\$ 5	\$ 2

El problema lo podemos ver como un modelo de redes de flujo a costo mínimo con límites de capacidad en los arcos como sigue, en la red también se está incluyendo un flujo factible inicial:

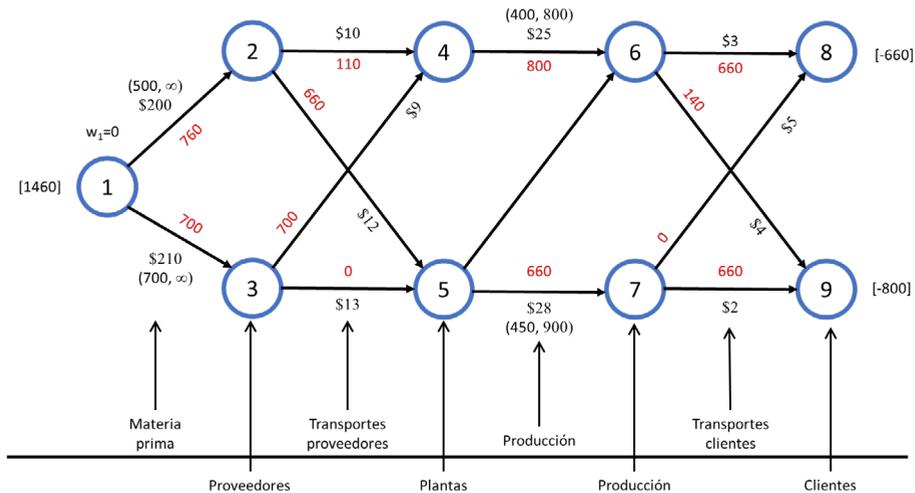


Figura 8.23 Red del ejemplo 8.3

También se puede plantear como un problema de programación lineal de la siguiente manera:

$$\begin{aligned} \text{Min } z = & 200 x_{12} + 210 x_{13} + 10 x_{24} + 12 x_{25} + 9 x_{34} + 13 x_{35} + 25 x_{46} + \\ & + 28 x_{57} + 3 x_{68} + 4 x_{69} + 5 x_{78} + 2 x_{79} \end{aligned}$$

Sujeto a

$x_{12} + x_{13} = 1460$	Nodo 1
$x_{12} - x_{24} - x_{25} = 0$	Nodo 2
$x_{13} - x_{34} - x_{35} = 0$	Nodo 3
$x_{24} + x_{34} - x_{46} = 0$	Nodo 4
$x_{25} + x_{35} - x_{57} = 0$	Nodo 5
$x_{46} - x_{68} - x_{69} = 0$	Nodo 6
$x_{57} - x_{78} - x_{79} = 0$	Nodo 7
$x_{68} + x_{78} = 660$	Nodo 8
$x_{69} + x_{79} = 800$	Nodo 9
$x_{12} \geq 500$	
$x_{13} \geq 700$	
$400 \leq x_{46} \leq 800$	
$450 \leq x_{57} \leq 900$	
$x_{ij} \geq 0$ para toda $i$ y $j$ .	

Resolviendo con el algoritmo simplex para redes se define:

**Paso 0. Determinación de la solución básica inicial**

Se determina un árbol como solución básica inicial, T: (1,3), (3,4), (3,5), (2,5), (4,6), (5,7), (6,8), (7,9) con flujo factible dado en los arcos y ajuste de las cotas inferiores de los arcos (4,6), (5,7), mismo que se puede ver en la siguiente figura 8.24:

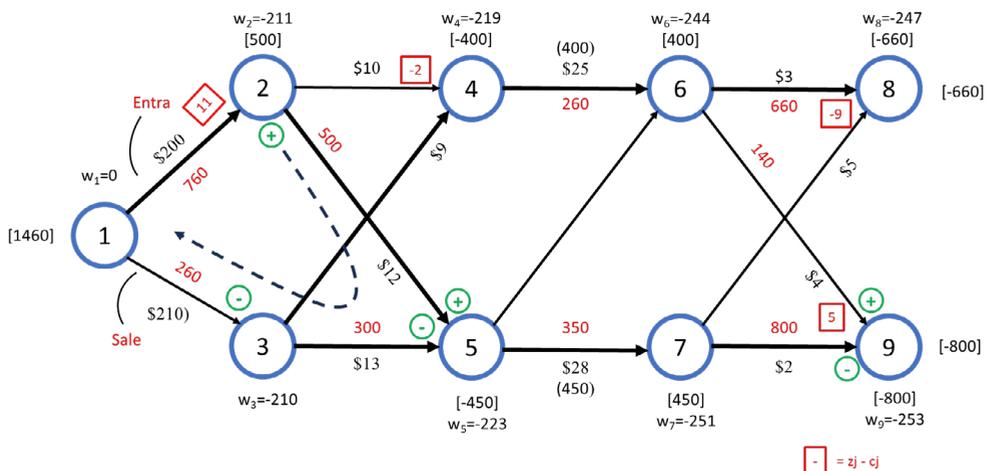


Figura 8.24 Primer árbol de expansión

**Paso 1. Determinación del arco que entra**

$$W_1 = 0$$

Arcos básicos

$$(1,3) = w_1 - w_3 = 210 \Rightarrow 0 - w_3 = 210 \Rightarrow w_3 = -210$$

$$(3,5) = w_3 - w_5 = 13 \Rightarrow -210 - w_5 = 13 \Rightarrow w_5 = -223$$

$$(2,5) = w_2 - w_5 = 2 \Rightarrow w_2 + 223 = 12 \Rightarrow w_2 = -211$$

$$(5,7) = w_5 - w_7 = 28 \Rightarrow -223 - w_7 = 28 \Rightarrow w_7 = -251$$

$$(3,4) = w_3 - w_4 = 9 \Rightarrow -210 - w_4 = 9 \Rightarrow w_4 = -219$$

$$(4,6) = w_4 - w_6 = 25 \Rightarrow -219 - w_6 = 25 \Rightarrow w_6 = -244$$

$$(6,8) = w_6 - w_8 = 3 \Rightarrow -244 - w_8 = 3 \Rightarrow w_8 = -247$$

$$(7,9) = w_7 - w_9 = 2 \Rightarrow -251 - w_9 = 2 \Rightarrow w_9 = -253$$

No básicos

$$(1,2) = w_1 - w_2 - c_{12} = 0 + 211 - 200 = 11 \leftarrow \text{entra y se forma el circuito}$$

$$1 \rightarrow 2 \rightarrow 5 \leftarrow 3 \leftarrow 1$$

$$(2,4) = w_2 - w_4 - c_{24} = -211 + 219 - 10 = -2$$

$$(7,8) = w_7 - w_8 - c_{78} = -251 + 247 - 5 = -9$$

$$(6,9) = w_6 - w_9 - c_{69} = -244 + 253 - 4 = 5$$

**Paso 2. Determinación del arco que sale**

1.- Incremento máx. permitido del arco que entra (1,2) = ∞

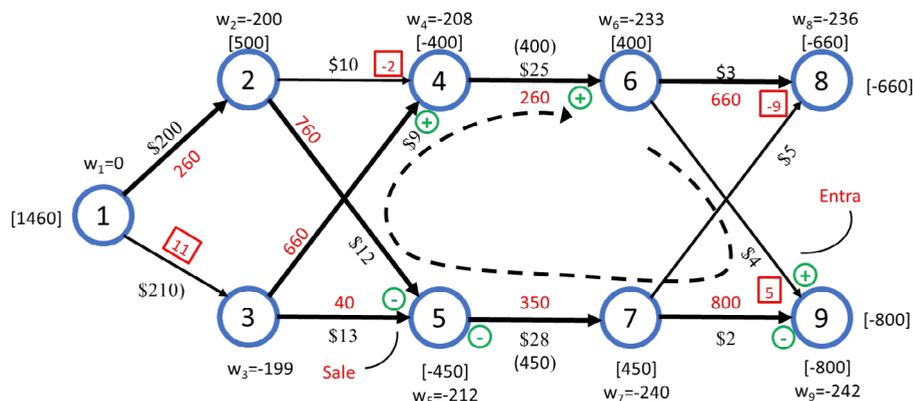
2.- Incremento máx. permitido del arco que entra (2,5) = ∞

3.- Decremento máx. permitido del arco que entra (3,5) = 300

4.- Decremento máx. permitido del arco que entra (1,3) = 260

**Figura 8.25**

Primera iteración



**Paso 1. Determinación del arco que entra**

$$W_1 = 0$$

Se calculan  $w_j$ 's

Arcos básicos

$$(1,2) = w_1 - w_2 = 200 \Rightarrow 0 - w_2 = 200 \Rightarrow w_2 = -200$$

$$(2,5) = w_2 - w_5 = 12 \Rightarrow -200 - w_5 = 12 \Rightarrow w_5 = -212$$

$$(3,5) = w_3 - w_5 = 13 \Rightarrow w_3 + 212 = 13 \Rightarrow w_3 = -199$$

$$(3,4) = w_3 - w_4 = 9 \Rightarrow -199 - w_4 = 9 \Rightarrow w_4 = -208$$

$$(5,7) = w_5 - w_7 = 28 \Rightarrow -212 - w_7 = 28 \Rightarrow w_7 = -240$$

$$(4,6) = w_4 - w_6 = 25 \Rightarrow -208 - w_6 = 25 \Rightarrow w_6 = -233$$

$$(6,8) = w_6 - w_8 = 3 \Rightarrow -233 - w_8 = 3 \Rightarrow w_8 = -236$$

$$(7,9) = w_7 - w_9 = 2 \Rightarrow -240 - w_9 = 2 \Rightarrow w_9 = -242$$

No básicos

$$(1,3) = w_1 - w_3 - c_{13} = 0 + 199 - 210 = -11$$

$$(2,4) = w_2 - w_4 - c_{24} = -200 + 208 - 10 = -2$$

$$(6,9) = w_6 - w_9 - c_{69} = -233 + 242 - 4 = 5$$

$$(7,8) = w_7 - w_8 - c_{78} = -240 + 236 - 5 = -9$$

Entra (6,9) y se forma el circuito  $6 \rightarrow 9 \leftarrow 7 \leftarrow 5 \leftarrow 3 \rightarrow 4 \rightarrow 6$

**Paso 2. Determinación del arco que sale**

1.- Incremento máx. de (6,9) =  $\infty$

2.- Incremento máx. de (3,4) = 660

3.- Decremento máx. (4,6) = 260

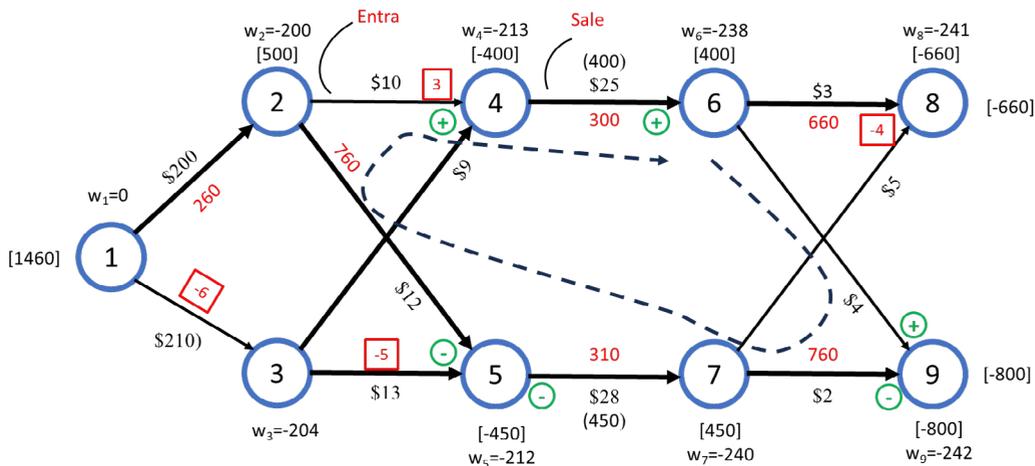
4.- Decremento máx. (7,9) = 800

5.- Decremento máx. (5,7) = 350

6.- Decremento máx. (3,5) = 40  $\leftarrow$  sale (3,5)

**Figura 8.26**

Segunda iteración



**Paso 1. Determinación del arco que entra**

$$W_1 = 0$$

Se calculan las  $w_j$ 's

Se ajusta el flujo

Arcos básicos

- $(1,2) = w_1 - w_2 = 200 \Rightarrow 0 - w_2 = 200 \Rightarrow w_2 = -200$
- $(2,5) = w_2 - w_5 = 12 \Rightarrow -200 - w_5 = 12 \Rightarrow w_5 = -212$
- $(5,7) = w_5 - w_7 = 28 \Rightarrow w_7 = -240$
- $(7,9) = w_7 - w_9 = 2 \Rightarrow w_9 = -242$
- $(3,4) = w_3 - w_4 = 9 \Rightarrow w_3 = 213 = 9 \Rightarrow w_3 = -204$
- $(6,9) \Rightarrow w_6 = -288$
- $(6,8) \Rightarrow w_8 = -241$
- $(4,6) = w_4 - w_6 = 25 \Rightarrow w_4 + 238 = w_4 = -213$
- $(3,4) = w_3 - w_4 = 9 \Rightarrow w_3 + 213 = 9 \Rightarrow w_3 = -204$

No básicos

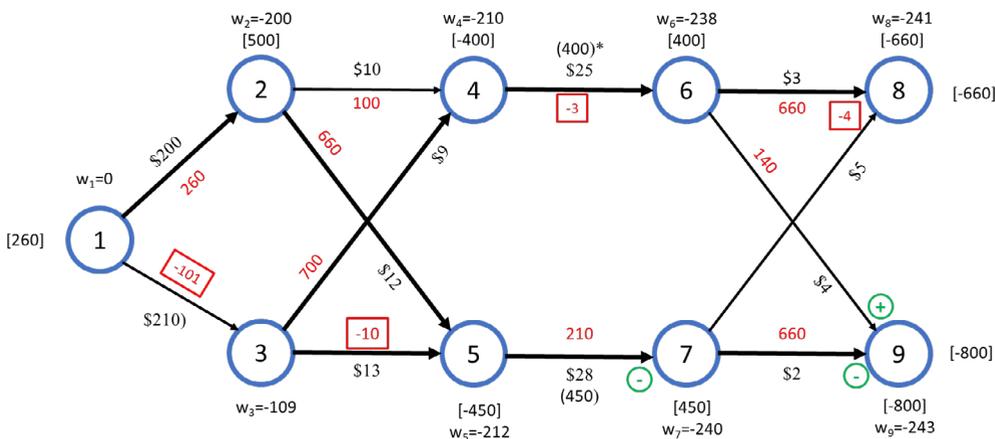
$$\begin{aligned} (1,3) &= w_1 - w_3 - c_{13} = 0 + 204 - 210 = -6 \\ (2,4) &= w_2 - w_4 - c_{24} = -200 + 213 - 10 = 3 \leftarrow \\ (3,5) &= w_3 - w_5 - c_{35} = -204 + 212 - 13 = -5 \\ (7,8) &= w_7 - w_8 - c_{78} = -240 + 241 - 5 = -4 \end{aligned}$$

Entra (2,4) y se forma el circuito  $2 \rightarrow 4 \rightarrow 6 \rightarrow 9 \leftarrow 7 \leftarrow 5 \leftarrow 2$

**Paso 2. Determinación del arco que sale**

- 1.- Incremento máx. de (2,4) =  $\infty$
- 2.- Incremento máx. de (4,6) = 100  $\leftarrow$  sale de su cota superior
- 3.- Incremento máx. de (6,9) =  $\infty$
- 4.- Decremento máx. (7,9) = 760
- 5.- Decremento máx. (5,7) = 310
- 6.- Decremento máx. (2,5) = 760

**Figura 8.27** Tercera iteración



El costo total óptimo es \$ 356, 560 con:

$$x_{12} = 260 + 500 = 760$$

$$x_{56} = 210$$

$$x_{13} = 0 + 700 = 700$$

$$x_{57} = 210 + 450 = 660$$

$$x_{24} = 100$$

$$x_{68} = 660$$

$$x_{25} = 660$$

$$x_{69} = 140$$

$$x_{34} = 700$$

$$x_{78} = 0$$

$$x_{35} = 0 = x_{78}$$

$$x_{79} = 660$$

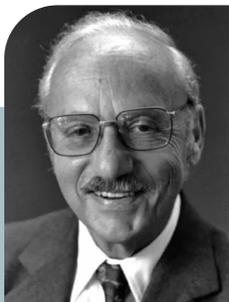
$$x_{46} = 400 + 400 = 800$$

La solución con LINDO es:

OBJECTIVE FUNCTION VALUE		
1) 356560.0		
VARIABLE	VALUE	REDUCED COST
X12	760.000000	0.000000
X13	700.000000	0.000000
X24	100.000000	0.000000
X25	660.000000	0.000000
X34	700.000000	0.000000
X35	0.000000	2.000000
X46	800.000000	0.000000
X57	660.000000	0.000000
X68	660.000000	0.000000
X69	140.000000	0.000000
X78	0.000000	4.000000
X79	660.000000	0.000000

**Figura 8.28**  
Solución con  
simplex estándar  
y LINDO

## 8.5 NOTAS HISTÓRICAS



George Dantzig

Dantzig (1951) desarrolló el algoritmo simplex para redes para el problema de transporte sin restricciones de capacidad, a través de especializar su método simplex de programación lineal. Probó la propiedad de árbol de expansión de la base y la propiedad de variable entera de la solución óptima. Más tarde desarrolló una técnica de cota superior para programación lineal, lo que llevó a una especialización eficiente del método simplex para el problema de flujo a costo mínimo.

El algoritmo simplex para redes ganó su actual popularidad en los inicios de los años setenta, cuando la comunidad de Investigación de Operaciones comenzó a desarrollar algoritmos de pruebas usando índices de árbol eficientes. Johnson (1966) sugirió los primeros índices de árbol, Srinivasan y Thompson (1973), Glover, Karney, Klingman y Napier (1974) implantaron estas ideas y encontraron que el algoritmo simplex para redes era sustancialmente más rápido que los existentes.



Ellis Johnson

Ford y Fulkerson (1957) desarrollaron los algoritmos primal-dual para el problema de transporte con restricciones de capacidad, más tarde generalizaron este método para resolver el problema de flujo a costo mínimo desarrollando el algoritmo de rutas más cortas.

1

2

3

4

5

6

7

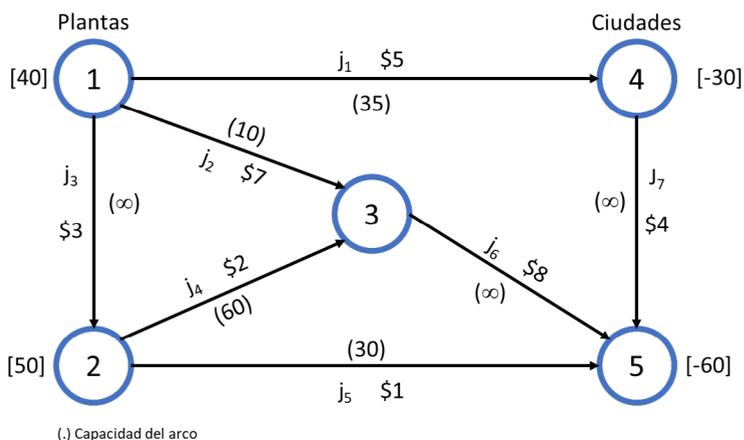
8

El algoritmo de eliminación de circuitos negativos se atribuye a Klein (1967). Tres aplicaciones de este algoritmo corren en tiempo polinomial: la primera debida a Barahona y Tardos (1989), la segunda debida a Goldberg y Tarjan (1988), y la tercera, a Wallacher y Zimmerman (1991).

## 8.6 EJERCICIOS PROPUESTOS

- Una red de agua conecta dos plantas desalinadoras a dos ciudades. La oferta total diaria de las dos plantas es de 40 y 50 millones de galones y las demandas diarias de las ciudades 1 y 2 son 30 y 60 millones de galones. Ambas plantas 1 y 2 tienen rutas directas a cada ciudad. El agua desalinada de las plantas se puede transportar de la planta 2 a una estación especial de bombeo. Adicionalmente, la planta 1 está unida a la planta 2 y la ciudad 1 a la ciudad 2. El modelo está balanceado porque la oferta en los nodos 1 y 2 es igual a la demanda en los nodos 4 y 5.

La figura 1 siguiente muestra la red, donde junto a los nodos están las ofertas y demandas, en los arcos aparecen los costos y entre paréntesis las capacidades máximas de los arcos



2. La compañía Audiomex produce aparatos de sonido portátiles. Sin embargo, la administración ha decidido subcontratar la producción de las bocinas necesarias para dichos aparatos de sonido. Existen tres proveedores. Sus precios por cada embarque de 1 000 bocinas se muestran en la siguiente tabla.

**Tabla 1.** Proveedor y precio por embarque

Proveedor	Precio
1	22,500
2	22,700
3	22,300

Además, cada proveedor cobrará un costo de envío. Cada embarque llegará a uno de los dos almacenes de la compañía. Cada proveedor tiene su propia fórmula para calcular este costo según los kilómetros recorridos hasta el almacén. Estas fórmulas y los datos del kilometraje se muestran a continuación.

**Tabla 2.** Proveedor y su costo por envío

Proveedor	Cargo por envío
1	\$ 300 + 0.40/km
2	\$ 200 + 0.50/km
3	\$ 500 + 0.20/km

**Tabla 3.** Proveedor y los km a los almacenes

Proveedor	Almacén 1	Almacén 2
1	1 600 km	400 km
2	500 km	600 km
3	2 000 km	1 000 km

Cuando una de las dos fábricas requiere un embarque de bocinas para amenizar los bailes, contrata un camión para traerlo de los almacenes. El costo por embarque se presenta en la siguiente tabla, junto con el número de embarques por mes que requiere cada planta.

**Tabla 4.** Costo por embarque de los almacenes a las fábricas

	Costo unitario por envío	
	Fábrica 1	Fábrica 2
Almacén 1	\$ 200	\$ 700
Almacén 2	\$ 400	\$ 500
Demanda mensual	10	6

Cada proveedor puede surtir hasta 10 embarques por mes; pero debido a las limitaciones de transporte, cada uno puede enviar un máximo de solo 6 embarques por mes a cada almacén. De manera similar, cada almacén puede enviar hasta 6 embarques por mes a cada fábrica.

Ahora, la administración desea desarrollar un plan mensual para determinar cuántos embarques (si son necesarios) ordenar a cada proveedor, cuántos de ellos deben ir a cada almacén y cuántos embarques debe enviar cada almacén a cada fábrica. El objetivo es minimizar la suma de los costos de compra (incluyendo los cargos de envío) y los costos de envío desde los almacenes a las fábricas.

- a) Dibuje la red que describa el problema de Audiomex, donde se muestren los nodos de suministro, transbordo y demanda, así como los costos asociados.
- b) Formule este problema como uno de flujo de costo mínimo, con todos los datos necesarios en la red. Además, incluya un nodo de demanda ficticio que reciba (sin costo) la capacidad no utilizada por los proveedores.

AGNARSSON, G., Greenlaw, R., *Graph Theory: Modeling, Applications and Algorithms*, Pearson Prentice Hall, Estados Unidos, 2007.

AHUJA, R.K., Magnanti, T.L., Orlin, J.B., *Network Flows*, Prentice Hall, Estados Unidos, 1993.

BALL, M.O., Magnanti, T.L., Monma, C.L., Nemhauser, G.L., Editors, *Network Models, Volume 7*, Handbooks in Operations Research and Management Science, Informs, Elsevier Science, 1995.

BAZARAA, M.S., Jarvis, J.J., Sherali, H.D., *Programación Lineal y Flujo en Redes*, 2ª. edición corregida, Limusa, México, 2007.

BERTSEKAS, D.P., *Linear Network Optimization, Algorithms and Codes*, Massachusetts Institute of Technology Press, Estados Unidos, 1992.

CAI, X., Sha, D., Wong, C.K., *Time-Varying Network Optimization*, Springer, International Series in Operations Research and Management Science, Estados Unidos, 2007.

GIBBONS, A., *Algorithmic Graph Theory*, Cambridge University Press, Inglaterra, 1985.

HILLIER, F.S., Hillier, M.S., Lieberman, G.J., *Métodos Cuantitativos para Administración, un enfoque de modelos y casos de estudio*, Mc Graw Hill, México, 2002, p.p. 239-283.

1

2

3

4

5

6

7

8

- HILLIER, F.S., Lieberman, G.J., *Introducción a la Investigación de Operaciones*, 8ª. edición, Mc Graw Hill, México, 2006, p.p. 374-439.
- HU, T.C., *Integer Programming and Network Flows*, Addison-Wesley, Estados Unidos, 1970.
- IRI, M., *Network Flow, Transportation and Scheduling, Theory and Algorithms*, volume 57, Academic Press, Estados Unidos, 1969.
- JENSEN, P.A., Barnes, J. W., *Network Flow Programming*, John Wiley and Sons, Estados, 1980.
- JOHNSONBAUGH, R., *Matemáticas Discretas*, 6ª. edición, Pearson Prentice Hall, México, 2005, p.p. 318-469.
- KENNINGTON, J.L., Helgason, R.V., *Algorithms for Network Programming*, John Wiley and Sons, Estados Unidos, 1980.
- LUENBERGER, D.G., *Programación Lineal y no Lineal*, Addison Wesley, México, 1989, p.p. 117-159.
- ROCKAFELLAR, R.T., *Network Flows and Monotropic Optimization*, John Wiley and Sons, Estados Unidos, 1984.
- TAHA, H.A., *Investigación de Operaciones*, 7ª. edición, Pearson Prentice Hall, México, 2004.
- WILSON, R.J., Watkins, J.J., *Graphs, an Introductory Approach*, John Wiley and Sons, Estados Unidos, 1990.
- WINSTON, W.L., *Investigación de Operaciones*, 4ª. edición, Thomson, México, 2005. p.p. 360-474.

1

2

3

4

5

6

7

8



UNIDAD DE APOYO EDITORIAL

*Teoría de redes*

Se publicó de manera digital en noviembre de 2023,  
en el repositorio de la Facultad de Ingeniería, UNAM,  
Ciudad Universitaria, Ciudad de México. C.P. 04510

El diseño y cuidado de la edición estuvo a cargo  
de la Unidad de Apoyo Editorial de la Facultad  
de Ingeniería. Las familias tipográficas utilizadas fueron  
Bree, Soruce Serif Pro y Sienna Math Pro,  
con sus respectivas variantes.