



UNIVERSIDAD NACIONAL
AVANZADA

DIVISION DE ESTUDIOS SUPERIORES
FACULTAD DE INGENIERIA
SECCION DE INVESTIGACION DE OPERACIONES

REDES DE FLUJO CON COSTOS CONVEXOS O CONCAVOS

T E S I S

que presenta

MARIO ALBERTO PARRA UGALDE

para obtener el grado de

MAESTRO EN INGENIERIA

(Investigación de Operaciones)

JURADO

DR. FELIPE OCHOA ROSSO

DR. SERGIO FUENTES MAYA

DR. MIGUEL COBIAN SELA

DR. JORGE DIAZ PADILLA

M en I FRANCISCO JAUFFRED M.

COORDINADOR DE LA SECCION

M en I Gustavo Rocha B.

SECRETARIO ACADEMICO

M en I Sergio Tirado L.

Cd. Universitaria México, D.F., Septiembre de 1982.



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

REFERENCIAS

1. Paul A. Jensen, J. Wesley Barnes. "Network Flow Programming", John Wiley and Sons, 1980.
2. Katta Murty. "Linear And Combinatorial Programming", John - - Wiley and Sons, 1976.
3. Mokhtar S. Bazaraa, John J. Jarvis. "Linear Programming And - Network Flows", John Wiley and Sons, 1976.
4. R. Gardinkel, G. Nemhauser. "Integer Programming", Wiley-Interscience, 1972.
5. W. Millar. "Some General Theorems For Nonlinear Systems Possessing Resistance", Philosophy Mag., 13, 1150-1160 (1951)
6. Jeff L. Kennington, Richard V. Helgason. "Algorithms For Network Programming", Wiley-Interscience, 1980.
7. H. Moskowitz, C.P. Wright, "Operations Research Techniques - For Management", Prentice-Hall, 1979

ABSTRACTO

En este trabajo se describe, con sentido práctico, una técnica de optimización relativa a la programación de redes de flujo aplicable a redes con funciones de costo no lineales separables. Además se proporcionan los algoritmos detallados y programas para computadora que implementan los procedimientos de resolución, los correspondientes manuales de usuario y ejemplos representativos resueltos con los programas.

El Capítulo 1 define los conceptos básicos sobre modelos de redes, la notación consistente empleada, cierta terminología necesaria y el modelo matemático del cual se deducen algunos resultados teóricos importantes sobre la teoría de la programación Lineal.

En el Capítulo 2 se consideran redes de flujo con costo mínimo en las que el costo del flujo en sus arcos es una función convexa. El método de resolución empleado consiste en una aproximación por segmentación implícita, en la cual, el flujo en los arcos es cambiado iterativamente en una cantidad preestablecida.

El Capítulo 3 estudia el problema de redes de flujo con costo mínimo cuando el costo del flujo en los arcos es una función cóncava, la cual se idealiza como una función de costo con cargo fijo. El algoritmo de resolución que se describe es un procedimiento de ramificación y acotamiento ("branch and bound"). En ambos capítulos, 2 y 3, se han reservado las dos últimas secciones para describir la organización y formato de datos requeridos por los programas y resolver algunos ejemplos sencillos fundamentalmente con el deseo de ilustrar eficazmente el uso de los programas.

Además se incluyen dos apéndices extensos e importantes, en los cuales se ofrece la descripción de los algoritmos de computadora para los modelos de redes. Tales algoritmos se presentan en cuatro niveles de abstracción: un riguroso diagrama de flujo (la forma del diagrama de flujo es particularmente compacta y facilita la lectura y comprensión), una descripción verbal (clara y concisa) paralela, un ejemplo con ilustraciones, y una codificación del lenguaje FORTRAN IV la cual se presenta en la última sección de cada apéndice.

Los temas y capítulos se han escrito de tal manera que ellos sean independientes, tanto como sea posible, de otros temas y capítulos. Esto se ha hecho con el ánimo de darle flexibilidad al reporte y facilitar la lectura. Creo que el nivel de discusión permitirá un estudio adecuado a aquellos que deseen aprender la técnica y los trucos para utilizarla y también a aquellos que deseen estudiar los algoritmos rigurosamente.

Finalmente, creo importante mencionar que la mayor parte del material de este trabajo, incluyendo los algoritmos, se adquirió de la referencia 1.

CONTENIDO

1	CONCEPTOS Y RESULTADOS	1
1.1	MODELOS DE REDES DE FLUJO	1
1.2	RELACION ENTRE LOS PROBLEMAS DE PROGRAMACION DE REDES DE FLUJO	3
1.3	PARAMETROS DE HOLGURA EN NODOS	4
1.4	NOTACION EN REDES	5
1.5	TRANSFORMACIONES DE PARAMETROS UTILES	10
1.6	MODELO MATEMATICO	11
1.7	TERMINOLOGIA SOBRE REDES	15
1.8	REDES EXPANDIDA Y MARGINAL	20
2	REDES DE FLUJO CON COSTOS CONVEXOS	23
2.1	FUNCIONES DE COSTO CONVEXAS	23
2.2	REDES DE FLUJO CON FLUJO FISICO	26
2.3	FUNCIONES DE COSTO QUE DEPENDEN DE VARIABLES ALEATORIAS	34
2.4	APROXIMACION POR SEGMENTACION LINEAL	37
2.5	APROXIMACION POR SEGMENTACION IMPLICITA	41
2.6	ORGANIZACION Y FORMATO DE DATOS PARA EL PROGRAMA NTCNVX	46
2.7	APLICACIONES RESUELTAS CON EL PROGRAMA NTCNVX	50
3	REDES DE FLUJO CON COSTOS CONCAVOS	75
3.1	APLICACIONES GENERALES	76
3.2	NOTACION	81
3.3	ENUMERACION EXHAUSTIVA	82
3.4	ENUMERACION IMPLICITA	87
3.5	ALGORITMO DE ENUMERACION IMPLICITA	89
3.6	DETERMINACION DE UNA COTA INFERIOR	92
3.7	ORGANIZACION Y FORMATO DE DATOS PARA EL PROGRAMA NTCNCV	101
3.8	APLICACIONES RESUELTAS CON EL PROGRAMA NTCNCV	105

APENDICE A

ALGORITMO DE RESOLUCION PARA REDES DE FLUJO CON COSTOS CONVEXOS	131
A.1 ALGORITMOS PARA MANIPULAR REDES	131
A.2 ALGORITMO NTCNVX	150
A.3 CODIFICACION FORTRAN IV DEL PROGRAMA NTCNVX	157

APENDICE B

ALGORITMO DE RESOLUCION PARA REDES DE FLUJO CON COSTOS CONCAVOS	173
B.1 ALGORITMO DE DESVIACIONES PARA RESOLVER REDES DE FLUJO CON COSTOS LINEALES	173
B.2 ALGORITMO NTCNCV	198
B.3 CODIFICACION FORTRAN IV DEL PROGRAMA NTCNCV	208

CAPITULO I

CONCEPTOS Y RESULTADOS

1.1 MODELOS DE REDES DE FLUJO

Como se ilustra en la Figura 1.1, una red es una colección de nodos y arcos. Esta representación es útil para modelar o formular una gran variedad de situaciones físicas y conceptuales. Por ejemplo, sistemas de carreteras para el tráfico de vehículos a menudo son formulados como modelos de redes. En éstos, los nodos representan a los centros de tráfico y los arcos son las vías que unen pares de centros. En forma similar, sistemas de ferrocarriles, sistemas de aerolíneas y, en general, todos los problemas de planeación del transporte se pueden representar apropiadamente como redes. Otras situaciones, tales como sistemas de inventarios, sistemas de producción-distribución, sistemas de comunicación, sistemas de ríos y sistemas de tubería, han sido representadas como redes. A los modelos de tales situaciones se les llama *modelos de redes de flujo*.

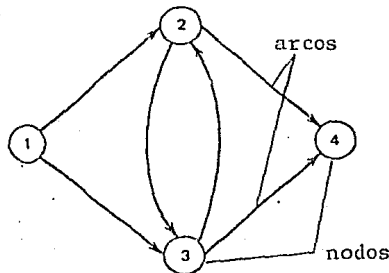
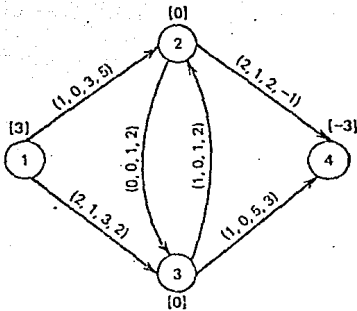


Figura 1.1

Red

Se considerará modelos de redes de flujo en los que el flujo en cada arco se maneja entre cierto rango, y el objetivo es seleccionar flujos en los arcos que optimicen alguna medida de efectividad. Para ilustrar esto, suponga que la red de la Figura 1.2 define un modelo de redes de flujo. Cada arco en la red lleva un flujo dirigido según la flecha y tiene tres parámetros asignados: una cota inferior, que es la mínima cantidad -- que puede fluir en el arco; una cota superior, que es la máxima cantidad de flujo que puede llevar el arco; y un costo para cada unidad de flujo que pase a través del arco.



Definición de términos
 [flujo externo fijo]
 (flujo, cota inferior,
 cota superior, costo)

Figura 1,2

Red de Flujo con Solución

Si el parámetro de cota inferior no está presente en el arco se asume que su cota inferior es cero y si está presente, y por alguna razón se debe reducir a cero, no hay problema, en la sección 1.5 se proporciona una transformación sencilla para remover cotas inferiores.

Con cada nodo se asocia un parámetro que representa la cantidad de flujo que entra o sale de la red por el nodo. A este parámetro se le llama flujo externo. Un flujo externo positivo indica que el flujo entra a la red y el nodo en cuestión, se dice que es de "depósito". Un flujo externo negativo indica que el flujo sale de la red siendo el nodo de "demanda". Si el flujo externo es igual a cero, no hay recursos disponibles ni demandados en el nodo, en este caso, se dice que el nodo es de "transbordo" (o intermedio). En cualquier caso, hay conserva--

ción del flujo, es decir, el flujo que entra a un nodo por los arcos más el flujo externo debe ser igual al flujo que sale.

Claramente, los flujos en los arcos son las variables de decisión en el problema de optimización, el cual, consiste en seleccionar flujos en los arcos que satisfaciendo las condiciones restrictivas establecidas, minimicen el costo total del flujo. El lector podría verificar que los flujos mostrados en la Figura 1.2 son óptimos para los parámetros dados.

1.2 RELACION ENTRE LOS PROBLEMAS DE PROGRAMACION DE REDES DE FLUJO.

El problema de optimizar alguna función objetivo sujeta a restricciones es denominado *programación matemática*. Debido a que todos los problemas considerados aquí se definen como una red que lleva flujo, se usa el término *programación de redes de flujo*. En la Figura 1.2 se ilustra un ejemplo específico de un problema de flujo a costo mínimo, lineal y sin ganancias. Se dice que el flujo es sin ganancia cuando el flujo se conserva al pasar a través del arco y, con ganancia, cuando el flujo se incrementa o disminuye por algún factor al pasar por el arco. A dicho factor se le llama ganancia. Ahora bien, en la Figura 1.3 se muestra la relación entre los diferentes problemas de programación de redes de flujo. El punto central de esta figura es el problema de flujo a costo mínimo, lineal y sin ganancia. A la izquierda se listan los problemas menos generales en el sentido de que son especializaciones del problema central. Los problemas listados a la derecha son más generales o bien, los problemas básicos son un caso particular de éstos. También se muestra el problema general de programación lineal para indicar su relación con los problemas de programación de redes. Se han identificado algoritmos que resuelven cada uno de los problemas de la Figura 1.3, encontrándose que los algoritmos para los problemas menos generales son más eficientes, en tiempo de proceso y espacio (memoria), que para los problemas más generales. Resulta que los algoritmos definidos para los problemas más generales pueden resolver los problemas en su forma menos general,

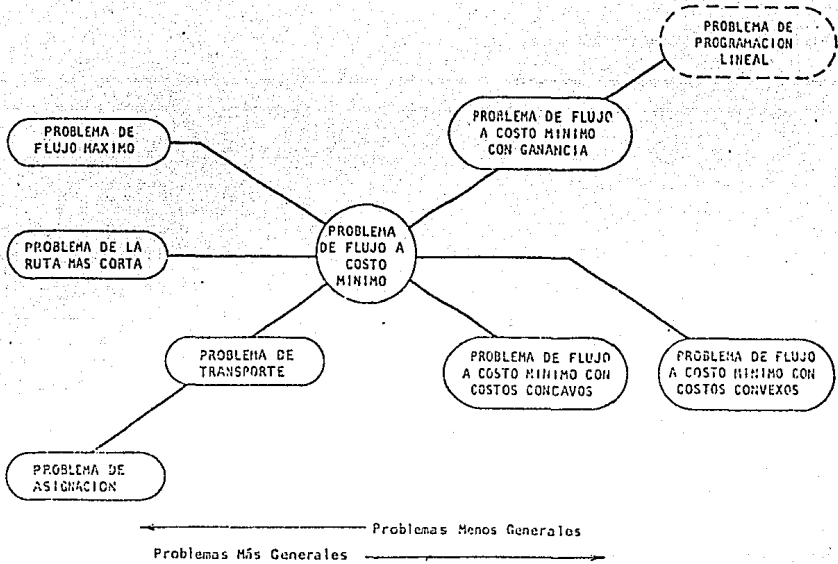


Figura 1.3

Relación entre los Problemas de Programación de Redes de Flujo

sin embargo, el inverso rara vez es cierto.

1.3 PARAMETROS DE HOLGURA EN NODOS

Ahora se incluyen dos parámetros adicionales en los nodos: el flujo externo de holgura y el costo unitario del mismo. Además del flujo externo fijo, es posible definir una cantidad - máxima de flujo adicional que puede entrar (si es positiva) o salir (si es negativa) del nodo: el flujo externo de holgura. Si esto es permitido se debe pagar un costo por cada unidad de ese flujo de holgura. Por ejemplo, en la Figura 1.4 al menos dos unidades de flujo entran a la red en el nodo 1. Si el costo está en pesos, hasta dos unidades más pueden entrar en ese nodo a un costo de \$1 por unidad de flujo. Una unidad de flujo debe salir del nodo 2 y hasta 2 unidades de flujo a un costo de menos \$2 por unidad (un costo negativo equivale a un beneficio)

pueden entrar. La interpretación para los nodos 3 y 4 es similar.

[flujo externo fijo, máximo
flujo externo de holgura,
costo de holgura]

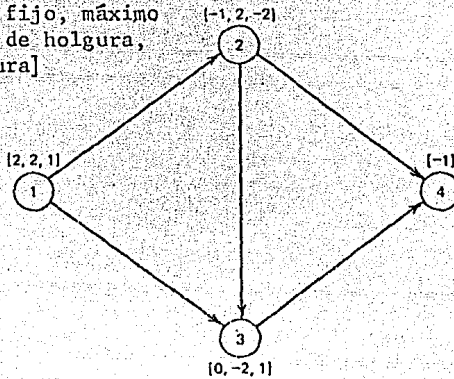


Figura 1.4

Red con Parámetros de Holgura en Nodos

1.4 NOTACION EN REDES

1.4.1. Estructura

La estructura de un modelo de redes está definida por no dos y arcos. Un nodo i es un elemento de la lista de nodos - $N = [1, 2, \dots, i, \dots, n]$, y un arco queda definido por un par ordena do de nodos (i, j) o como un elemento k de la lista de arcos - $M = [1, 2, \dots, k, \dots, m]$. El arco k o $k(i, j)$ se origina en el nodo i , nodo origen, y termina en el nodo j , nodo terminal o destino. Para definir todas las conexiones presentes en la red, se iden tifican las listas origen y destino

$$O = [o_1, o_2, \dots, o_m]$$

$$T = [t_1, t_2, \dots, t_m]$$

donde o_k y t_k son los nodos origen y destino, respectivamente, del arco k . La colección de nodos y arcos es una red dirigida (o sim plemente red) y se identifica por $D = [N, M]$, donde los valores

de n y m y los conjuntos O y T son suficientes para definir todos los arcos de la red.

Además de estas listas, se definen las listas de arcos - que se originan en el nodo i , $MO_i = [k | o_k = i]$, y terminan en el nodo i , $MT_i = [k | t_k = i]$.

1.4.2. Flujo y Costo

Flujo

El flujo en el arco usualmente representa alguna cantidad física tal como el flujo de un fluido, flujo de personas, artículos, materiales o flujo de dinero. Una característica principal del flujo en la red es que se conserva en cada nodo.

Costo

Un costo se asocia con el flujo en el arco. El costo en el arco es una función del flujo, $h_k(f_k)$. Nuestro objetivo será minimizar el costo de la red, el cual, es la suma de los costos de todos los arcos.

$$H(f) = \sum_{k=1}^m h_k(f_k)$$

Se identifican tres clases de funciones de costo: lineal, convexa, y cóncava. La función de costo lineal es

$$h_k(f_k) = h_k f_k$$

donde h_k es el parámetro de arco que proporciona el costo por unidad de flujo. En consecuencia, el costo lineal de la red es

$$H(f) = \sum_{k=1}^m h_k f_k = fh$$

Las Figuras 1.5a y 1.5b muestran varias funciones de costo convexas y cóncavas representativas.

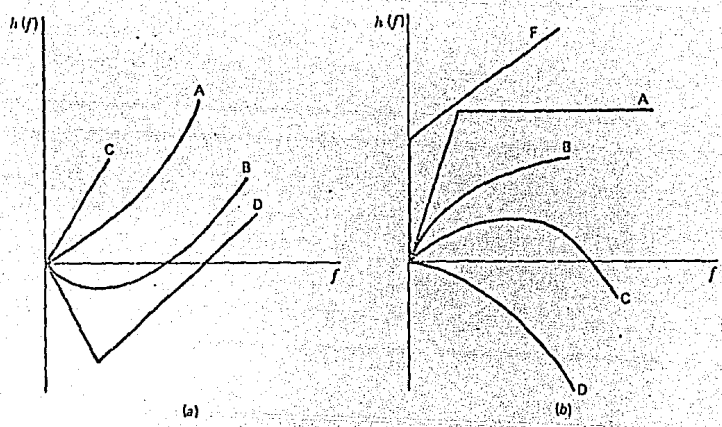


Figura 1.5
 Costos no Lineales Representativos
 (a) Costos Convexos (b) Costos Concavos

1.4.3. Representación Gráfica Del Modelo De Redes.

En la representación gráfica los nodos se simbolizan por círculos con el índice de nodo adentro. Los nodos se enumeran consecutivamente comenzando con el número uno y sus parámetros se muestran adyacentes a él entre paréntesis cuadrados. Un arco se dibuja como un segmento de línea dirigido que conecta sus nodos origen y terminal, y tanto su índice como sus parámetros se muestran adyacentes al arco, estos últimos, entre paréntesis. Los arcos también se enumeran consecutivamente, sin embargo, en este caso el lector deberá conocer la manera de efectuar la enumeración implementada en los algoritmos de resolución que se describen en los capítulos que siguen. Para identificar los parámetros, en cada representación gráfica se proporciona la definición de los mismos. En la Figura 1.6a se ilustra una red con los parámetros de nodo y arco.

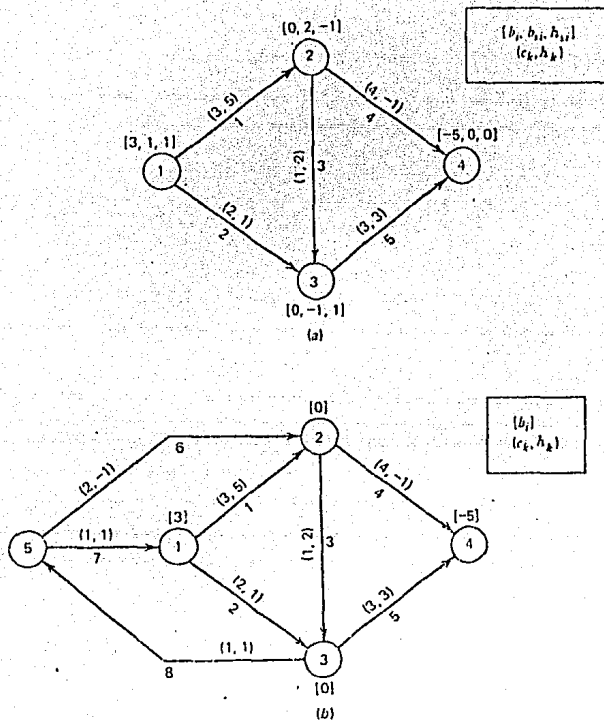


Figura 1.6

Representación de los Flujos Externos de Holgura
 (a) Con Parámetros de Nodo (b) Con un Nodo de Holgura

Una representación alternativa de los flujos de holgura-
 usa arcos en sustitución de los parámetros de holgura de los no
 dos. En esta representación, se crea un nodo de holgura que no
 requiere conservación de flujo y se le designa con la enumera-
 ción más alta en la red, es decir, será el nodo n . Para un flu-
 jo de holgura f_{sl} positivo (negativo), se construye un arco des-
 de (hacia) el nodo de holgura hacia (desde) el nodo i . La capacidad del ar-

co es $|b_{si}|$ y el costo es h_{si} . El algoritmo del Apéndice A usa precisamente esta representación, la cual es creada por el mismo algoritmo.

1.4.4. Conservación del Flujo.

Un flujo factible $f(f=[f_1, f_2, \dots, f_m])$ es aquel que satisface la conservación de flujo en todos los nodos excluyendo al nodo de holgura. De acuerdo con la definición de nodo y arcos de holgura todos los flujos en la red, excepto los flujos externos fijos en los nodos, están en arcos. Por consiguiente, la conservación de flujo en cada nodo implica:

flujo total que sale del nodo - flujo total que entra = flujo externo fijo en el nodo.

Matemáticamente, la restricción de la conservación de flujo o ecuación de Kirchoff para un nodo i cualquiera es

$$\sum_{k \in NO_i} f_k - \sum_{k \in MT_i} f_k = b_i \quad \left. \begin{array}{l} \text{si } b_i > 0 \text{ el nodo } i \text{ es de depósito} \\ \text{si } b_i = 0 \text{ el nodo } i \text{ es de transbordo} \\ \text{si } b_i < 0 \text{ el nodo } i \text{ es de demanda} \end{array} \right\}$$

Para la red de la Figura 1.6a las restricciones de conservación de flujo son

nodo

$$\begin{array}{rcl} 1 : & f_1 + f_2 & - f_7 = 3 \\ 2 : & -f_1 & + f_3 + f_4 - f_6 = 0 \\ 3 : & -f_2 - f_3 & + f_5 + f_8 = 0 \\ 4 : & & - f_4 - f_5 = -5 \end{array}$$

Observe que las columnas que no representan arcos de holgura -- (1,2,3,4 y 5) tienen exactamente dos elementos diferentes de cero. Si la columna representa al arco $k(i,j)$, aparece un coeficiente +1 en la fila i y un -1 en la fila j . Distintamente, los arcos de holgura solo tienen un elemento diferente de cero debido a -

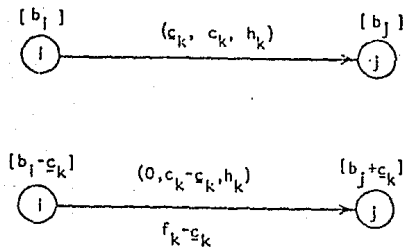
que la restricción de conservación de flujo no se escribe para el nodo de holgura. A la matriz de los coeficientes asociados con las ecuaciones de conservación de flujo se le denomina *matriz de incidencia nodos-arcos*. Si esta se denota por A y por b , el vector columna de los flujos externos fijos, las ecuaciones de conservación de flujo se pueden escribir matricialmente como $Af=b$.

1.5 TRANSFORMACIONES DE PARAMETROS UTILES.

1.5.1. Transformación De Capacidades Mínimas A Cero.

En algunas ocasiones puede ser conveniente o necesario transformar un problema de redes de flujo con cotas inferiores diferentes de cero a un problema equivalente con $c_k=0$ para $k=1,2,\dots,m$. Esta transformación permite eliminar la capacidad mínima de la lista de parámetros en los arcos, lo cual, puede significar un chorro importante de memoria principal en el computador cuando se resuelven redes con gran cantidad de arcos. Para lograrla se procede a cambiar

1. c_k por cero
2. c_k por $c'_k = c_k - c_k$
3. f_k por $f'_k = f_k - c_k$
4. b_i por $b'_i = b_i - c_k$
5. b_j por $b'_j = b_j + c_k$



El procedimiento se aplica arco por arco para todos aquellos con $c_k \neq 0$. Cada transformación usa los parámetros actualizados como resultado de la anterior.

1.5.2. Transformación De Los Parámetros De Nodo

Los parámetros de nodos pueden convertirse fácilmente en parámetros de arco reemplazando el nodo en cuestión por dos no-

dos y un arco, como se observa en la Figura 1.7.

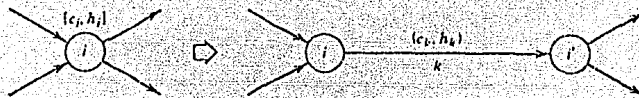


Figura 1.7

Conversión de Parámetros de Nodo a Arco

1.5.3. Transformación de un Arco No Dirigido.

Un arco no dirigido $(i;j)$ con capacidad mínima igual a cero y capacidad máxima c_{ij} se puede manejar reemplazándolo por un par de arcos opuestos (i,j) y (j,i) , ambos con capacidad mínima igual a cero y capacidad máxima igual a c_{ij}



Figura 1.8

Transformación de un Arco no Dirigido

Otra manera de manejar este caso es asignando una dirección arbitraria a la línea y tratarla como un arco con $c_k = -c$ y $c_k = c$. Luego, si el flujo en el arco es positivo, este va en la dirección supuesta; mientras que si el flujo es negativo, fluye en la dirección contraria a la supuesta con el signo cambiado, es decir, positivo. Por supuesto, lo anterior procede si el algoritmo de resolución usado permite flujos negativos.

1.6 MODELO MATEMATICO

La representación gráfica junto con los parámetros de nodo y arco son suficientes para definir un modelo de redes de --flujo. Las características del nodo de holgura, la función ob-jetivo y el concepto familiar de la conservación del flujo es-tán implícitos en la definición.

Ahora se describe la representación matemática del modelo, en la cual, se especifican explícitamente todas las características del mismo. La representación gráfica se usará para presentar los ejemplos, mientras que la matemática es importante para deducir los resultados teóricos necesarios empleados en el algoritmo de resolución. En términos de los parámetros y variables ya definidas, el problema de redes de flujo con costo mínimo lineal se puede expresar como un problema de programación lineal (PPL),

$$\begin{aligned} \text{Min. } & \sum_{k=1}^m h_k f_k \quad (\text{o equivalentemente Max. } \sum_{k=1}^m -h_k f_k) \\ \text{s.a. } & \sum_{k \in MO_i} f_k - \sum_{k \in MT_i} f_k = b_i \quad i=1, \dots, n-1 \\ & f_k \geq 0 \\ & \quad \quad \quad k=1, \dots, m \\ & f_k \leq c_k \end{aligned}$$

donde n es el nodo de holgura.

En notación matricial el modelo anterior es

$$\text{Min. } hf \tag{1}$$

$$\text{s.a. } Af = b \tag{2}$$

$$f \geq 0 \tag{3}$$

$$f \leq c \tag{4}$$

Llegados a este punto, se puede ya resumir algunos resultados teóricos importantes sobre la teoría de la programación lineal. Estos son bien conocidos y se presentan aquí sin prueba; el lector puede encontrar una justificación rigurosa de ellos en la numerosa bibliografía existente al respecto.

Considere la partición de la matriz A de manera que $A = [B, R]$. Cualquier matriz B formada por $n-1$ columnas linealmente independientes de A es denominada una base y las variables correspondientes a estas son las variables básicas, las cuales se-

representan con el vector f_B . El conjunto de variables no básicas asociadas con R es f_R . Ahora bien, las restricciones del PPL se pueden escribir como

$$[B, R] \begin{bmatrix} f_B \\ f_R \end{bmatrix} = b \quad (5)$$

y resolviendo para f_B se tiene

$$f_B = B^{-1} [b - Rf_R] \quad (6)$$

Para encontrar una solución básica, primero se asigna a cada variable no básica el valor de cero (cota inferior) o c_k (cota superior); luego se usa la ecuación (6) para obtenerla.

El dual del PPL es

$$\begin{aligned} \text{Min} \quad & \sum_{i=1}^{n-1} \pi_i b_i + \sum_{k=1}^m \delta_k c_k \\ & \pi_i - \pi_j + \delta_k \geq -h_k \quad k = 1, 2, \dots, m \\ & \quad \quad \quad i = o(k), j = t(k) \\ & \pi_i \text{ libre} \quad i=1, \dots, n-1 \\ & \delta_k \geq 0 \end{aligned}$$

En notación matricial

$$\text{Min.} \quad \pi b + \delta c \quad (7)$$

$$\text{sa.} \quad \pi A + \delta I \geq -h \quad (8)$$

$$\pi \text{ libre} \quad (9)$$

$$\delta \geq 0 \quad (10)$$

Las condiciones necesarias y suficientes, denominadas de *Kuhn y Tucker*, para que las variables del PPL correspondan a la solución óptima, han sido especializadas para el problema de redes de flujo con costo mínimo en los siguientes tres teoremas.

TEOREMA 1

Dada una solución $f = [f_B, f_R]^t$ al problema primal y una-

solución $[\pi, \delta]$ al problema dual. Entonces las soluciones son óptimas si y solo si:

1. f es factible para el problema primal; es decir, se satisfacen (2), (3) y (4).
2. $[\pi, \delta]$ es factible para el dual; es decir, se satisfacen (8) y (10).
3. Se satisface la holgura complementaria; es decir,
 - (a) $f_k(i, j) > 0$, $\pi_i - \pi_j + \delta_k = -h_k$
 - (b) $f_k < c_k$, $\delta_k = 0$
 - (c) $\pi_i - \pi_j + \delta_k > -h_k$, $f_k(i, j) = 0$
 - (d) $d_k > 0$, $f_k(i, j) = c_k$

TEOREMA 2

Si $[\pi, \delta]$ es una solución óptima del problema dual, entonces, --
 $\delta_k = \max [0, \pi_j - \pi_i - h_k]$.

Supuesto el Teorema 2 la optimalidad del problema se puede presentar como sigue.

TEOREMA 3.

Dada una solución f para el problema primal y una solución parcial π para el problema dual, las soluciones son óptimas para sus respectivos problema si y solo si se satisfacen -- las siguientes condiciones:

1. factibilidad primal.
2. $\delta_k = \text{Max}[0, \pi_j - \pi_i - h_k]$ (factibilidad dual restringida)
3. Holgura complementaria. Para cada arco $k(i, j)$
 - (a) $\pi_j - \pi_i = h_k$ para $0 < f_k < c_k$
 - (b) $f_k = 0$ para $\pi_j - \pi_i > h_k$
 - (c) $f_k = c_k$ para $\pi_j - \pi_i < h_k$

Como una ayuda visual la tercera condición se ilustra -- en un diagrama de holgura complementaria para el arco $k(i, j)$.

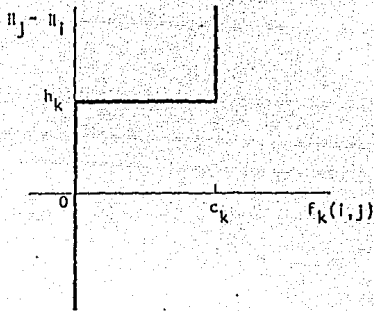


Figura 1.9
Diagrama de Holgura Complementaria

1.7 TERMINOLOGIA SOBRE REDES

Antes de seguir adelante con la exposición de este trabajo es preciso introducir ciertos términos sobre la teoría de redes. Además se pretende unificar términos, pues, es común encontrar en la literatura al respecto diferentes definiciones para el mismo término.

1.7.1. Gráfica, Multigráfica y Gráfica Dirigida.

Una gráfica, denotada por $G=[N,M]$, consiste de un conjunto finito de nodos N y un conjunto M formado por pares desordenados de nodos de N . Cada elemento de M es una línea o arista y se representa por $(i;j)$ donde i y j son elementos de N . Una "vuelta" ("loop") se forma con una línea que une a un nodo consigo mismo. En una gráfica no se permiten vueltas ni líneas múltiples, es decir, dos o más líneas que unen al mismo par de nodos. En cambio, en una multigráfica sí se permiten líneas múltiples pero no vueltas. Finalmente, una gráfica es dirigida si todas sus líneas son arcos o pares ordenados de nodos (i,j) . En la Figura 1.10 se ilustran ejemplos de estos términos.

1.7.2. Cadena y Circuito.

Una cadena del nodo i_1 al nodo i_r en G , es una sucesión de nodos y arcos distintos, $i_1, k_1, i_2, k_2, \dots, k_{r-1}, i_r$, donde k_s es el arco (i_s, i_{s+1}) o la línea $(i_s; i_{s+1})$ con su orientación seleccionada de i_s a i_{s+1} , la cual se trata como el arco (i_s, i_{s+1}) para $s=1, \dots, r-1$. De esta manera, cada arco en la cadena tiene la misma dirección. Un circuito es una cadena con dos o más arcos de i_1 a i_1 . Ver Figura 1.11.

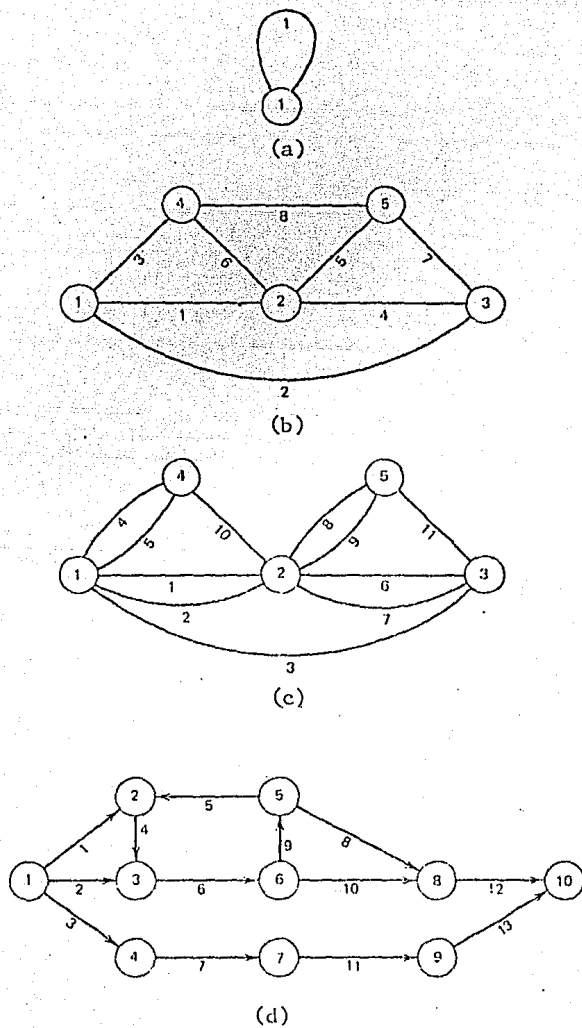


Figura 1.10

Ejemplos de (a) Vuelta ("Loop") (b) Gráfica

(c) Multigráfica (d) Gráfica Dirigida

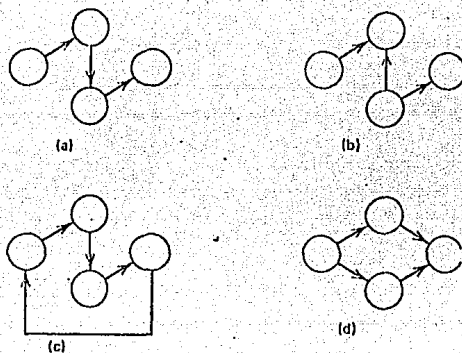


Figura 1.11
Gráficas Dirigidas (a) Cadena (b) Trayectoria
(c) Circuito (d) Ciclo

1.7.3. Trayectorias y Ciclos.

Si en la definición de cadena se permite que cada arco tenga la forma (i_s, i_{s+1}) o bien (i_{s+1}, i_s) para $s=1, \dots, r-1$, se obtiene una trayectoria del nodo i_1 al nodo i_r . Una trayectoria de i_1 a i_1 que contiene al menos dos arcos es un ciclo y se dice que contiene a i_1 . Ver Figura 1.11. De las definiciones anteriores se deduce que toda cadena es una trayectoria y que todo circuito es un ciclo, pero no recíprocamente.

Gráfica Conectada.

G es una gráfica conectada si existe una trayectoria entre cada par de nodos. En caso contrario G es desconectada.

1.7.4. Red y Subred Dirigidas

Una red dirigida es una multigráfica dirigida $D=[N, M]$ o-

bien, es una gráfica en que se permiten líneas múltiples.

Una subred $D_s = (N_s, M_s)$ de D es una red dirigida con $N_s \subset N$ y $M_s \subset M$.

1.7.5. Árboles y Bosques

Un árbol es una gráfica conectada y sin ciclos, mientras que un bosque es una gráfica y sin ciclos. Un árbol dirigido $D_T = (N_T, M_T)$, es una subred que define una trayectoria única entre algún nodo específico, llamado nodo raíz, y cualquier otro nodo en N_T . Una subred de expansión de D es una subred con $N_s = N$, por consiguiente, un árbol de expansión (o generador) de D tiene $N_T = N$. Un bosque de expansión de D es una subred de expansión dirigida, desconectada y sin ciclos.

Características de los árboles.

1. El número de arcos en un árbol es $n-1$ donde n es el número de nodos.
2. En un árbol de expansión dirigido solo el nodo raíz no es terminal de algún arco.
3. Cualquier número de arcos se pueden originar en un nodo del árbol dirigido.
4. Entre cualesquier par de nodos hay una única trayectoria.
5. Las $n-1$ columnas de la matriz A , que forman la base para el problema de flujo a costo mínimo, corresponden a los arcos en M_T de un árbol de expansión.
6. Los arcos que no se incluyen en M_T reciben el nombre de aristas. Si una arista es agregada a un árbol, se forma un único ciclo y la red resultante ya no es un árbol.
7. Si existe una trayectoria dirigida o cadena del nodo i al nodo j entonces, el nodo i precede al j y el nodo j sucede al i . Si militarmente, el arco k precede al arco k' si existe una trayectoria dirigida que los contiene y k aparece primero.

La Figura 1.12 ilustra cuatro árboles obtenidos de la Figura 1.6b. En (b) se tiene un árbol de expansión y (c) y (d) -

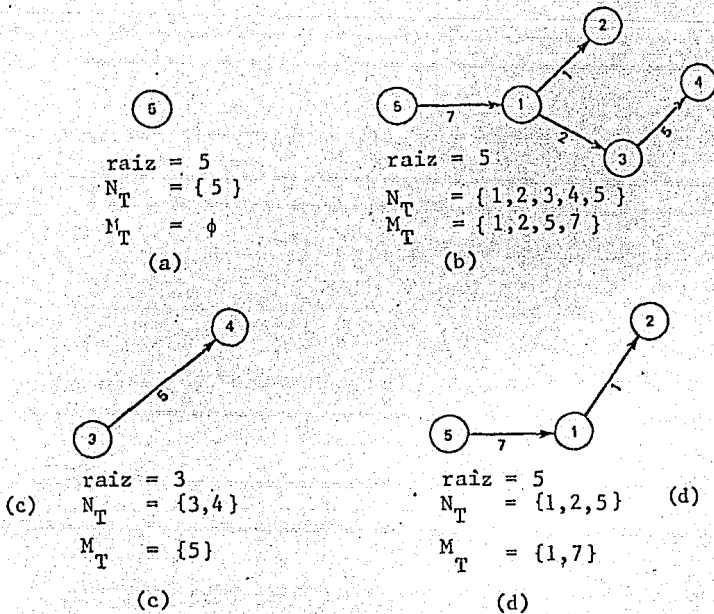


Figura 1.12
Ejemplos de Redes Arboles

tomadas juntas constituyen un bosque de expansión.

Apuntador hacia atrás.

Un árbol dirigido puede definirse mediante los apuntadores hacia atrás de los nodos que lo forman. El apuntador hacia atrás de un nodo i cualquiera es el índice del único arco que termina en el nodo i . Dada la característica 2., solo el nodo-raíz del árbol no tiene apuntador hacia atrás y esto se indica asignándole el valor cero. A continuación se proporcionan los apuntadores hacia atrás del árbol de la Figura 1.12b

Nodo	1	2	3	4	5
Apuntador hacia atrás	7	1	2	5	0

1.7.6. Red Circulatoria

Una red es circulatoria si no contiene flujos externos, es decir, si todos sus nodos son de transbordo. Conviene establecer que cualquier red con flujos externos puede convertirse -

en una red circulatoria equivalente. Esto se logra agregando - dos nodos: un nodo fuente $s=n+1$ y un nodo sumidero $t=n+2$, y arcos artificiales para representar los flujos externos fijos y - de holgura. El nodo s se conecta con cada nodo de depósito, - mientras que cada nodo de demanda se conecta con el nodo t . Finalmente se genera un "arco de retorno" del nodo t al s . Debido a que el algoritmo del Apéndice B hace uso de esta red modelo, se ha reservado el apartado B.1.1 para desarrollar completamente dicha conversión y la manera como el algoritmo enumera - los arcos.

1.8 REDES EXPANDIDA Y MARGINAL

1.8.1 Red Expandida

Llegados a este punto, es claro que los modelos de redes considerados aquí se basan en gráficas dirigidas. Es conveniente, desde el punto de vista de obtener soluciones óptimas a los problemas de redes de flujo, definir una red expandida, la cual se obtiene directamente de la red original.

Primero es necesario definir la existencia de un arco re flejado, $-k$, para $k \in M$ tal que el arco $-k$ conecta los mismos nodos que k (arco hacia adelante) pero tiene dirección opuesta.

$$o(-k) = t(k)$$

$$t(-k) = o(k)$$

La red expandida, $D_E = [N, N_E]$, tiene el mismo conjunto de nodos - que D y su conjunto de arcos contiene, además de los arcos de D , todos los arcos reflejados; esto es, si $M = [1, 2, \dots, M]$ entonces - $M_E = [1, 2, \dots, m, -1, -2, \dots, -m]$. La Figura 1.13 muestra una red dirigida y su red expandida asociada.

Nótese que si la red original es conectada, entonces - existe una trayectoria dirigida entre cada par de nodos en la - red expandida. Es claro que el arco reflejado de un arco reflejado es su asociado arco hacia adelante.

Para una red sin ganancia, si h_k es el costo para el arco hacia adelante k , entonces $-h_k$ es el costo para el arco re- flejado $-k$.

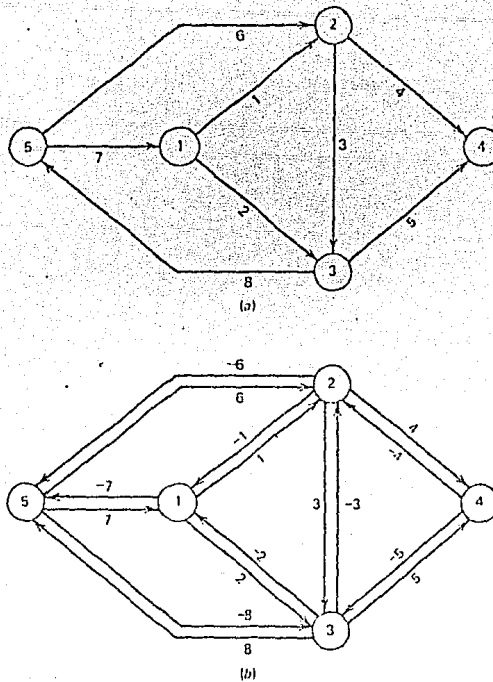


Figura 1.13

Ejemplo de Red Expandida (a) Red Dirigida
(b) Red Expandida Asociada

1.8.2 Red Marginal

La red marginal, $D^* = [N, M^*]$, tiene el mismo conjunto de nodos que D y D_E . El conjunto de arcos de D^* consiste de un subconjunto de M_E que incluye solo arcos admisibles. Un arco hacia adelante es admisible si el flujo no ha alcanzado su capacidad; esto es, si es posible incrementar el flujo en la red original. Un arco reflejado es admisible si el flujo en el correspondiente arco hacia adelante no es igual a cero (cota inferior); esto es, si es posible disminuir el flujo en el arco asociado de la red original.

La definición usual de la *función de admisibilidad* que indica la admisibilidad de los arcos de la red expandida es

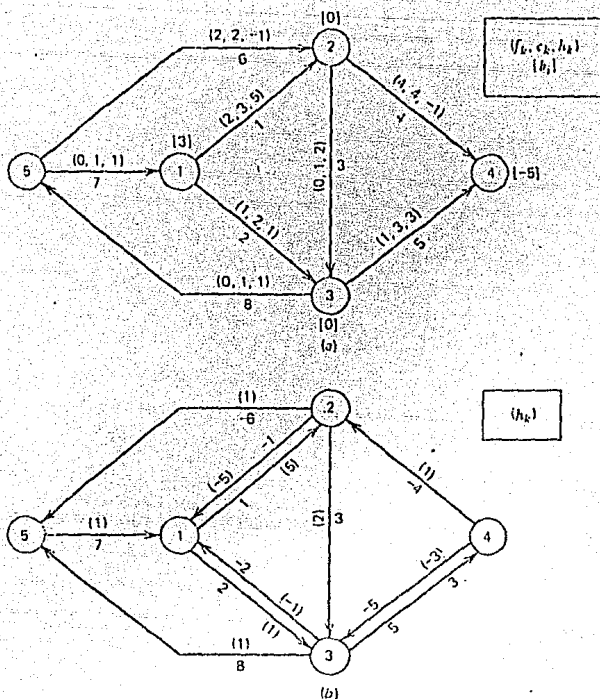


Figura 1.14
Ejemplo de Red Marginal (a) Red con Flujos Asignados
(b) Red Marginal Correspondiente

$$A_d(k) = 1 \quad \begin{cases} \text{si } k > 0 \text{ y } f_k < c_k \\ \text{si } k < 0 \text{ y } \underline{f}_k > 0 \end{cases} \quad \text{arco admisible}$$

$$A_d(k) = 0 \quad \begin{cases} \text{si } k > 0 \text{ y } f_k \geq c_k \\ \text{si } k < 0 \text{ y } \underline{f}_k \leq 0 \end{cases} \quad \text{arco inadmissible}$$

La Figura 1.14b muestra la red marginal asociada con la Figura 1.14a como resultado de aplicar la anterior función de admisibilidad.

CAPITULO 2

REDES DE FLUJO CON COSTOS CONVEXOS

En este capítulo se considera el caso en que el costo de flujo en el arco es una función convexa del flujo. En muchos casos, por no tener la herramienta adecuada, se ignoran los costos convexos y se modelan como costos lineales, obteniéndose resultados significativamente diferentes de aquellos que se obtendrían al considerarse los costos convexos. Entonces, la introducción de este tipo de costos permite obtener soluciones exactas y nos abre el campo de aplicación, pudiéndose ahora resolver otra clase de problemas prácticos. Además se proporciona un procedimiento de solución cuya eficiencia relativa con otros procedimientos se incrementa conforme aumenta el número de arcos con costo convexo.

2.1. FUNCIONES DE COSTO CONVEXAS

Se recuerda que en el capítulo previo no se permitía que el flujo tomará valores negativos. En este capítulo si se permite que el flujo tome valores negativos, por tanto, la nueva condición de factibilidad es

$$c_k \leq f_k \leq c_k$$

donde c_k puede ser negativo. La dirección de los arcos siempre se indica con una flecha aunque sólo sea como referencia. Así, si $f_k(i,j) > 0$, el flujo va del nodo i al nodo j ; mientras que si $f_k(i,j) < 0$, el flujo va del nodo j al nodo i con signo positivo.

En la Figura 2.1 se muestran tres funciones convexas diferentes, en donde $h(f)$ representa la función de costo del arco.

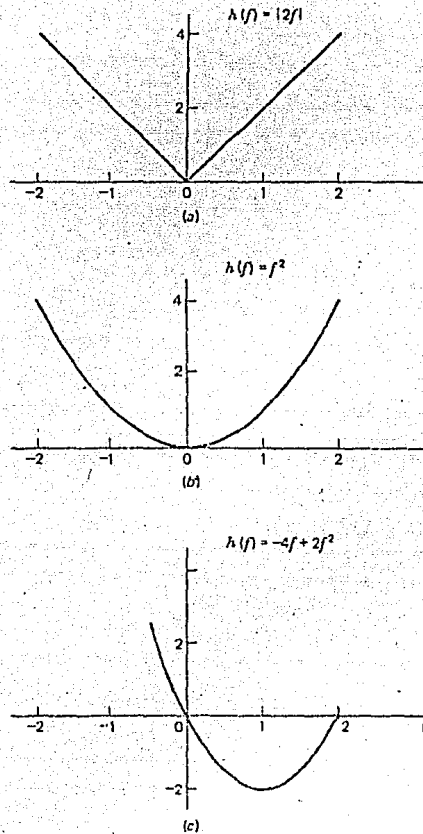


Figura 2.1

Ejemplos de Funciones de Costo Convexas

Una función es convexa si al trazar una línea recta entre dos puntos de la función, dicha línea queda comprendida en la función o por encima de la función como se ilustra en la Figura 2.2. Matemáticamente, esto se puede escribir como sigue

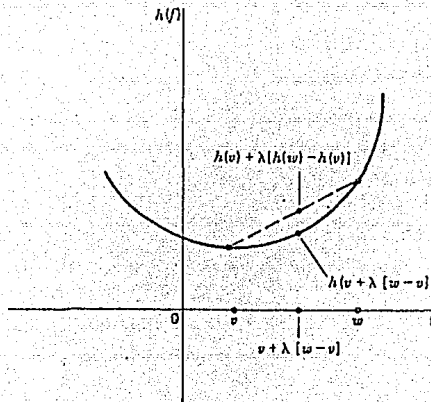


Figura 2.2
Una Función Convexa

$h(f)$ es convexa si y solo si

$$\lambda h(v) + (1-\lambda)h(w) \geq h(\lambda v + (1-\lambda)w) \quad (1)$$
para todo λ , $0 \leq \lambda \leq 1$

donde v y w son valores de flujo tal que $w > v$. Esta definición es muy útil visualmente, sin embargo, raramente es usada en la determinación analítica de convexidad. La función convexa que aquí se considera debe ser continua. No es necesario que $h(f)$ sea diferenciable en cada punto, pero deberá tener la primera derivada $[h'(f)]$ bien definida en todos los puntos de un número finito de puntos. Bajo estas condiciones, una definición alternativa de convexidad dice que la derivada de h debe ser monótona creciente cuando f crece, esto es

$h(f)$ es convexa si y solo si es continua y

$$h'(w) \geq h'(v) \quad \text{para todo } w > v \text{ en que está} \quad (2)$$
definida la primera derivada

Esta definición no necesita que h' sea continua pero, si sucede que h' es continua y definida en todos los puntos, es posible usar la segunda derivada como prueba de convexidad.

$h(f)$ es convexa si y solo si $h''(f) \geq 0$ para todo f que tiene definida la segunda derivada, (3)

EJEMPLO: Considere la función de costo cuadrática $h(f) = f^2$. Entonces

$$h'(f) = 2f \quad \text{y} \quad h''(f) = 2$$

Puesto que h' es creciente y continúa las condiciones (2) y (3) indican convexidad.

2.2 REDES DE FLUJO CON FLUJO FISICO

La mayoría de las disciplinas de la Ingeniería tratan con problemas que envuelven flujo de alguna cantidad física en una red. En electricidad se considera el flujo de corriente eléctrica por cables. En el análisis de tuberías se trata con el flujo de fluidos en tubos. En sistemas térmicos se trata con el flujo de calor a través de conductos.

Para cada sistema físico de estos se han derivado técnicas computacionales que resuelven el problema de redes de flujo. Por otra parte, hay una equivalencia entre esas técnicas y la técnica de solución para los problemas de programación de redes de flujo. En particular, cada uno de los problemas mencionados se puede representar y resolver como un problema de redes de flujo con costos convexos en los arcos. Por tanto, se podrán resolver con la técnica que se describe más adelante.

Para ilustrar la equivalencia, considere el problema de encontrar las corrientes en una red de corriente directa, CD. Es importante mencionar que no es posible usar esta técnica en redes que contengan inductancia o capacitancia y corrientes alternas. Los problemas de flujo en tuberías y flujo de calor son análogos al problema de la red CD.

Considere el ejemplo de la Figura 2.3 donde se supone que las resistencias son lineales (el caso no lineal se considera después). Un resistor R tiene la característica de que si la corriente I pasa a través de él, entonces aparece un voltaje

V entre sus terminales, tal que

$$V = IR$$

Las cantidades V, I y R se miden en voltios, amperes y ohmios, respectivamente. La batería mostrada proporciona un voltaje constante entre sus terminales, o sea, es independiente de la corriente que la atraviesa.

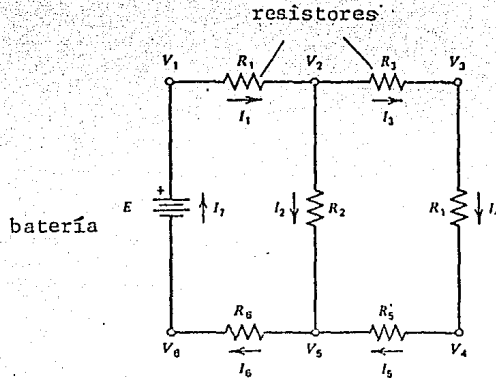


Figura 2.3

Una Red Resistiva de Corriente Directa (CD)

Esta red se resuelve para las corrientes en los resistores y los voltajes en los nodos. La solución se puede obtener haciendo uso de las Leyes De Kirchoff.

1. Las sumas de las corrientes que entran y salen de un nodo deben ser iguales. Esta condición es equivalente a la conservación de flujo.
2. El voltaje presente en cada resistor debe ser igual a la diferencia de potencial entre los nodos que terminan el resistor.

En el diagrama de la Figura 2.3 V_i representa el potencial en el nodo i y I_k representa la corriente en el resistor k . Las direcciones asignadas a las corrientes son arbitrarias y se usan únicamente para referencia. Note que cada I_k puede ser positiva o negativa y no está acotada en magnitud.

Para el problema ejemplo, las Leyes de Kirchoff se escriben como sigue.

$$\begin{array}{rcl}
 I_7 - I_1 & & = 0 \\
 I_1 - I_2 - I_3 & & = 0 \\
 I_3 - I_4 & & = 0 \\
 I_2 - I_4 - I_5 & & = 0 \\
 I_5 - I_6 & & = 0 \\
 -I_7 & & I_6 = 0
 \end{array}$$

Estas ecuaciones son similares a las ecuaciones de conservación de flujo para el problema de programación de redes. Una de estas ecuaciones es redundante. Ahora se escribe la Ley de Kirchoff para los voltajes.

$$\begin{array}{rcl}
 -V_1 & + V_6 & = -E \\
 V_1 - V_2 & & = I_1 R_1 \\
 V_2 & - V_5 & = I_2 R_2 \\
 V_2 - V_3 & & = I_3 R_3 \\
 V_3 - V_4 & & = I_4 R_4 \\
 V_4 - V_5 & & = I_5 R_5 \\
 V_5 - V_6 & & = I_6 R_6
 \end{array}$$

Puesto que todos los potenciales aparecen como diferencias en estas ecuaciones se puede, arbitrariamente, hacer cero uno de los potenciales; digamos V_6 . Entonces se tienen 7 corrientes y 5 voltajes como incógnitas y 12 ecuaciones lineales. Este problema de resolver un conjunto de ecuaciones lineales definidas por un circuito eléctrico CD es bien conocido y se puede resolver usando el procedimiento estandar para resolver ecuaciones lineales.

EJEMPLO

Sea $E=10$, $R_1=1$, $R_2=3$, $R_3=1$, $R_4=2$, $R_5=1$ y $R_6=2$. Resolviendo el conjunto de ecuaciones se tiene

$$I_1 = I_6 = I_7 = 2.12$$

$$I_3 = I_4 = I_5 = 0.91, \quad I_2 = 1.21$$

$$V_1 = 10.0, \quad V_2 = 7.88, \quad V_3 = 6.97$$

$$V_4 = 5.15, \quad V_5 = 4.24, \quad V_6 = 0.00$$

Hay casos en que los resistores son no lineales y no es posible usar el procedimiento para resolver ecuaciones lineales. Estos casos, en que los resistores no lineales se representan por una relación no lineal entre el voltaje y la corriente, son más difíciles de resolver; pero esto se puede hacer con éxito disponiendo de las técnicas modificadas de Newton-Raphson.

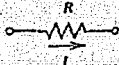
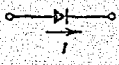
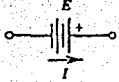
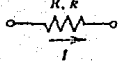
Seguidamente se describe un método alternativo para resolver las corrientes de una red CD, el cual representa el problema como un problema de redes con costo mínimo convexo. El método hace uso del concepto *contenido* de un elemento resistivo. El contenido de un elemento resistivo de dos terminales en un circuito eléctrico CD, depende de las características de su voltaje.

Considere un elemento k como el de la Figura 2.4, en el que arbitrariamente se ha establecido una dirección positiva para la corriente I_k que va del nodo i al nodo j . Cuando la corriente va del nodo j al i toma un valor negativo. El voltaje en el elemento se define como una función de I_k y se escribe

$$v_k(I) = V_i - V_j$$

A esta función $v_k(I)$ se le llama *voltaje característico*. La forma de la misma depende de la clase particular del elemento bajo consideración. Varios elementos se listan en el Cuadro 2.1. Note que el voltaje característico puede tener diferentes definiciones según la corriente sea positiva o negativa.

Cuadro 2.1 VOLTAJE CARACTERISTICO PARA VARIOS ELEMENTOS

Elemento	Símbolo	Voltaje característico
Resistor lineal		$v(I) = IR$
Diodo		$I \geq 0 \quad v(I) = IR_1$ $I < 0 \quad v(I) = IR_2$ $R_2 \gg R_1$
Batería		$v(I) = -E$
Resistor no lineal		$I \geq 0 \quad v(I) = RI^k$ $I < 0 \quad v(I) = R(-I)^k$

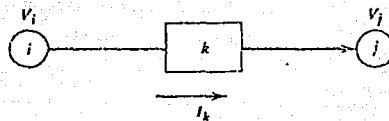


Figura 2.4

Un Elemento Resistivo

El contenido de un elemento para una corriente I_k se define como el área debajo del voltaje característico entre 0 y $-I_k$. Esta se muestra en la Figura 2.5a

$$G_k(I_k) = \int_0^{-I_k} v(I) dI \quad (4)$$

Si la corriente es negativa, el contenido del elemento es el --área indicada en la Figura 2.5b y la fórmula adecuada es

$$G_k(I_k) = -\int_{I_k}^0 v(I) dI \quad (5)$$

Al aplicar las fórmulas (4) y (5) debe tenerse cuidado con $v(I)$, puesto que su forma puede depender del signo de la corriente. En el Cuadro 2.2 se listan las funciones contenido para los elementos considerados en el Cuadro 2.1

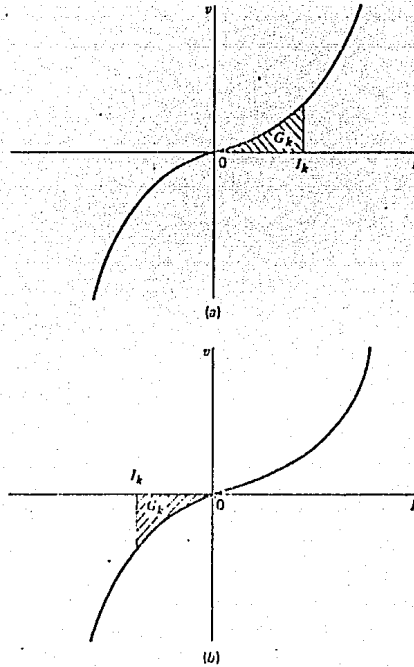


Figura 2.5

El Contenido de un Elemento (a) para corriente positiva (b) para corriente negativa

Cuadro 2.2 FUNCION CONTENIDO PARA LOS ELEMENTOS DEL CUADRO 2.1

Elemento	Función Contenido
Resistor lineal	$G(I) = I^2 R / 2$
Diodo	$I \geq 0 : G(I) = I^2 R_1 / 2$
	$I < 0 : G(I) = I^2 R_2 / 2$
Batería	$G(I) = - EI$
Resistor no Lineal	$I \geq 0 : G(I) = R I^{k+1} / (k+1)$
	$I < 0 : G(I) = R (-I)^{k+1} / (k+1)$

Ahora bien, falta definir el contenido de una red, el cual es simplemente la suma de los contenidos de sus elementos.

$$G_{\text{red}} = \sum_{k=1}^m G_k$$

donde M es el conjunto de elementos en la red. Se ha demostrado [Referencia 5.] que la solución para las corrientes (las cuales deben satisfacer la conservación de flujo en cada nodo) de una red eléctrica es un punto estacionario de la función contenido de la red. Para ilustrar esto, considere la batería simple y el resistor lineal de la Figura 2.6. Del Cuadro 2.2 los contenidos de los elementos son

$$G_B = -10I_B \quad \text{y} \quad G_R = 2I_R^2$$

y por consiguiente el contenido de la red es

$$G_{\text{red}} = -10I_B + 2I_R^2 \quad (6)$$

La conservación de flujo requiere que

$$I_B = I_R = I \quad (7)$$

Sustituyendo (7) en (6) e igualando la derivada con cero se obtiene el punto estacionario.

$$\frac{dG_{\text{red}}}{dI} = -10 + 4I = 0 \quad \text{o} \quad I = 2.5 \text{ amperes}$$

Esta es la corriente que fluye en el circuito de la Figura 2.6.

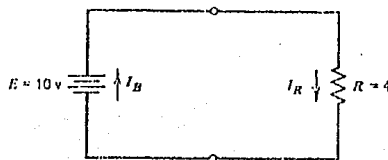


Figura 2.6
Ejemplo de Red Eléctrica

Cuando la función $v(I)$ es monótona creciente, como todos los casos del Cuadro 2.1, la función contenido definida en la ecuación (1) es una función convexa. En consecuencia, una red eléctrica que solo contiene elementos de este tipo debe tener una función contenido convexa. Esto garantiza que un punto estacionario también sea un mínimo global. De esta manera, concluimos que el procedimiento para encontrar el flujo a costo mínimo en redes con funciones de costo convexo en los arcos también se puede usar para encontrar el contenido mínimo de los flujos, y por tanto, resolver problemas de redes de flujo físicas.

El enfoque de la programación de redes de flujo puede ser muy práctico para resolver redes de flujo físicas. La información que debe proporcionar el analista es simple. Sólo se requiere la descripción de la red física y los parámetros de sus componentes; no es necesario definir las ecuaciones de Kirchoff. Es cierto que los componentes no lineales se manejan con más dificultad que los lineales, pero la diferencia no es importante. Las limitaciones en los diodos y corrientes se modelan fácilmente con los parámetros de capacidad mínima y máxima de los arcos. Las fuentes de corriente y los sumideros son modelados como flujos externos en los nodos. Finalmente, el flujo en el arco se relaciona con la corriente y el potencial de nodo (π_i) con el voltaje en el nodo.

Un problema análogo es el de determinar el flujo de fluido en una red de tubos. En este caso, f_k es la razón de flujo en el tubo k y π_i es la altura en el nodo i . La pérdida de altura en el tubo $k(i,j)$ es dada por la fórmula de Hazen-Williams

$$\pi_i - \pi_j = K f_k^x$$

donde K es una constante que depende de la longitud, ancho y material del tubo y x es un exponente que toma los valores 1.85 o 2.0 dependiendo de la aproximación usada. En este caso, el contenido del tubo k es

$$G_k = \begin{cases} \frac{K f^{x+1}}{x+1} & \text{para } f \geq 0 \\ \frac{K(-f)^{x+1}}{x+1} & \text{para } f < 0 \end{cases}$$

Este contenido individual se convierte en un término de la función objetivo en la programación de redes de flujo con costo -- convexo.

Para sistemas termales, la resistencia termal a la transferencia de calor a través de un cuerpo sólido es

$$R = L/KA$$

donde L es la longitud de la trayectoria de transmisión de calor, A es el área de la sección transversal, y K es el coeficiente de conductividad termal. En este caso, f_k corresponde a la razón de flujo de calor (en unidades de calor por unidad de tiempo) y π_i representa la temperatura en grados del cuerpo i.

2.3 FUNCIONES DE COSTO QUE DEPENDEN DE VARIABLES ALEATORIAS

Todos los problemas considerados previamente son determinísticos; esto es, no consideran directamente variables aleatorias. En esta sección se presentan ciertos problemas prácticos que involucran riesgo y que poseen equivalentes determinísticos que tienen funciones de costo convexas.

Considere un arco con función de costo $h(f, x)$ que depende del flujo y de alguna variable aleatoria (v.a.) x . Si x tiene una función densidad de probabilidad $g(x)$, el costo esperado en función de flujo es

$$\bar{h}(f) = E h(f, x) = \int_{-\infty}^{\infty} h(f, x) g(x) dx$$

Una condición suficiente para asegurar la convexidad de $\bar{h}(f)$ es que su primera derivada sea continua y su segunda derivada sea positiva en todos los puntos; esto es

$$\frac{d^2 \bar{h}}{df^2} = \int_{-\infty}^{\infty} \frac{\partial^2 h}{\partial f^2} g(x) dx \geq 0$$

Puesto que $g(x)$ es una función densidad y es positiva en todos los puntos, la condición de convexidad se satisface si $h(f, x)$ es una función convexa de f para todos los valores de x donde $g(x)$ es estrictamente positiva. Una condición suficiente más general no requiere que la segunda derivada exista en todos los puntos, sólo requiere que la derivada parcial de h con respecto a f sea una función monótona creciente de f para todo x .

EJEMPLO

Considere un arco cuyo flujo representa el número de trabajadores disponibles en un almacén. Sea x una v.a. que representa el trabajo requerido en términos del número de trabajadores y $D(x-f)^2$ una penalización considerada debido a la diferencia entre el número de trabajadores disponibles y requeridos. Si $g(x)$ es una función densidad de probabilidad con media u y variancia σ^2 , el valor esperado del costo en el arco k es

$$\begin{aligned}\bar{h}(f) &= \int_{-\infty}^{\infty} D(x-f)^2 g(x) dx = \int_{-\infty}^{\infty} D[x-u-(f-u)]^2 g(x) dx \\ &= D \left[\int_{-\infty}^{\infty} (x-u)^2 g(x) dx - 2(f-u) \int_{-\infty}^{\infty} (x-u)g(x) dx + (f-u)^2 \int_{-\infty}^{\infty} g(x) dx \right] \\ &= D \left[\sigma^2 + (f-u)^2 \right]\end{aligned}$$

Esta función incluye un término constante que depende de la variancia y una función cuadrática del flujo que es convexa. Note que la función de costo esperado no depende de la forma de la función densidad.

Otra forma que frecuentemente aparece en la práctica es

$$h(f, x) = \begin{cases} h_1(f, x) & \text{si } x \leq f \\ h_2(f, x) & \text{si } x > f \end{cases} \quad (9)$$

El costo esperado requiere que la integral se divida en dos partes

$$\bar{h}(f) = \int_{-\infty}^f h_1(f, x)g(x)dx + \int_f^{\infty} h_2(f, x)g(x)dx \quad (10)$$

Suponiendo que la primera y segunda derivadas existen en todos los puntos, se tiene que

$$\frac{d^2 \bar{h}(f)}{df^2} = \int_{-\infty}^f \frac{\partial^2 h_1}{\partial f^2} g(x) dx + \int_f^{\infty} \frac{\partial^2 h_2}{\partial f^2} g(x) dx + g(f) \left[\frac{\partial h_1}{\partial f} - \frac{\partial h_2}{\partial f} \right]_{x=f}$$

Puesto que $g(x)$ es siempre positiva, una condición suficiente para que $\bar{h}(f)$ sea convexa es

1. h_1 y h_2 son convexas para todo x y f
2. $\partial h_1 / \partial f \geq \partial h_2 / \partial f$ en $x = f$

EJEMPLO

Considere la situación en que se debe asignar un flujo a cierto arco para satisfacer una demanda x . Si el flujo proporcionado es mayor que la demanda no se incurre en ningún costo pero, si el flujo es menor que la misma, se incurre en una penalización de $-D(x-f)$. La cantidad positiva D es un castigo por cada unidad de demanda no satisfecha. De esta manera

$$h(f, x) = \begin{cases} 0 & \text{si } x \leq f \\ D(x-f) & \text{si } x \geq f \end{cases}$$

$$h_1(f, x) = 0 \quad h_2(f, x) = D(x-f)$$

$$\frac{\partial h_1}{\partial f} = 0 \quad \frac{\partial h_2}{\partial f} = -D$$

$$\frac{\partial^2 h_1}{\partial f^2} = 0 \quad \frac{\partial^2 h_2}{\partial f^2} = 0$$

Note que h_1 y h_2 son convexas (lineales) para todo x . La condición (2) también se satisface puesto que $0 > -D$; por tanto, esta función es convexa. La función de costo esperado es

$$\bar{h}(f) = \int_f^{\infty} D(x-f)g(x)dx$$

Suponga que la v.a. tiene una función densidad uniforme

$$g(x) = \begin{cases} 1/3 & \text{si } 0 \leq x \leq 3 \\ 0 & \text{en otro caso} \end{cases}$$

Claramente

$$\bar{h}(f) = \int_f^3 \frac{D(x-f)}{3} dx \quad 0 \leq f \leq 3$$

$$\bar{h}(f) = -\frac{3D}{2} - Df + \frac{Df^2}{6}$$

Evaluando la segunda derivada ($\frac{\partial^2 D}{\partial f^2} > 0$) se obtiene que la función es convexa.

2.4 APROXIMACION POR SEGMENTACION LINEAL

Una manera de resolver el problema convexo separable es haciendo una aproximación por segmentación lineal en la función de costo convexa de cada arco. Para cada segmento se define -- una variable nueva, es decir, un arco con una función de costo lineal y se resuelve el problema lineal resultante. Esto se -- ilustra en el ejemplo de la Figura 2.8 en donde la función de costo original es

$$h(f) = f^2 ; \quad -2 \leq f \leq 2$$

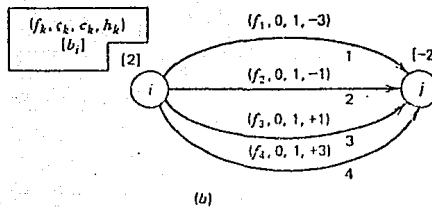
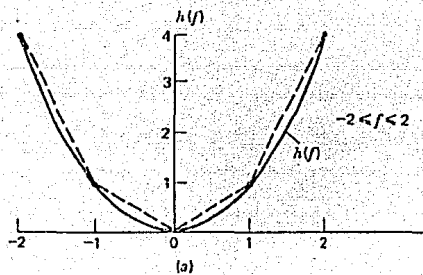


Figura 2.8

Aproximación por Segmentación Lineal para una Función Convexa

Esta función puede ser aproximada dividiendo el rango de f en cuatro segmentos de longitud uno cada uno, y definiendo -- las nuevas variables f_1 , f_2 , f_3 y f_4 para representar los segmentos. La función de costo aproximada es

$$h_p(f) = -3f_1 - f_2 + f_3 + 3f_4 + 4$$

$$f = f_1 + f_2 + f_3 + f_4 - 2$$

Observe que esta función es apropiada solo si se agrega la condición de que las variables se incrementen en orden, esto es

si $f_2 > 0$ entonces $f_1 = 1$

si $f_3 > 0$ entonces $f_2 = 1$

si $f_3 > 0$ entonces $f_3 = 1$

Esta condición se satisface automáticamente para el problema de costo mínimo, puesto que como $h(f)$ es convexa $h_1 < h_2 < h_3 < h_4$. De esta manera, siempre es óptimo incrementar f_1 hasta su máximo - antes de que f_2 se incremente desde cero. Similarmente f_2 se incrementa hasta su máximo antes de que f_3 cambie desde cero, etc. Como el objetivo es minimizar costos, si dos arcos son paralelos el que tenga costo más bajo se usa y satura primero.

Para el caso en que $h(f)$ es cóncava $h_1 > h_2 > h_3 > h_4$, y las condiciones anteriores no se satisfacen automáticamente. Esto con lleva una significativa dificultad en el cálculo y se considera en el próximo capítulo.

La estructura de la red para nuestro ejemplo, aparece en la Figura 2.8b. Observe que un simple arco se ha sustituido -- por cuatro arcos y que se han agregado flujos externos en los nodos como resultado del proceso de conversión de capacidades mínimas a cero.

Se recuerda de la sección 1.5 que el cambio en los flujos externos de los nodos que definen un arco son

$$\Delta b_i = -c$$

$$\Delta b_j = c$$

El término constante en h_p ($h(c)$) se ha eliminado. Esto no afecta los flujos óptimos en la estructura de la Figura 2.8b, - pero si afecta el costo total de la solución óptima. La ventaja de esta transformación es que el problema se puede resolver con la técnica disponible para arcos con costo lineal. La desventaja es que el problema se expande en el número de arcos -- por un factor igual al número de segmentos usados en la aproximación.

La exactitud de la aproximación está bajo el control del modelador. Más segmentos en la aproximación produce una aproximación más exacta pero también un problema (red) más grande.

En términos generales, considere la función convexa en el arco $k(i,j)$

$$h(f), \quad c \leq f \leq c$$

Defina los puntos de quiebra a_1, a_2, \dots, a_p en el eje f , donde $c = a_0$ y $c = a_p$. Sea $h(a_1)$ el costo en el punto de quiebra a_1 . Entonces los parámetros en el l -ésimo arco son

$$\text{capacidad mínima : } c_l = 0$$

$$\text{capacidad : } d_l = a_l - a_{l-1} \quad l=1, \dots, p$$

$$\text{costo (pendiente): } h_l = \frac{h(a_l) - h(a_{l-1})}{a_l - a_{l-1}}$$

La Figura 2.9 recoge otro caso de aproximación por segmentación lineal de una función convexa.

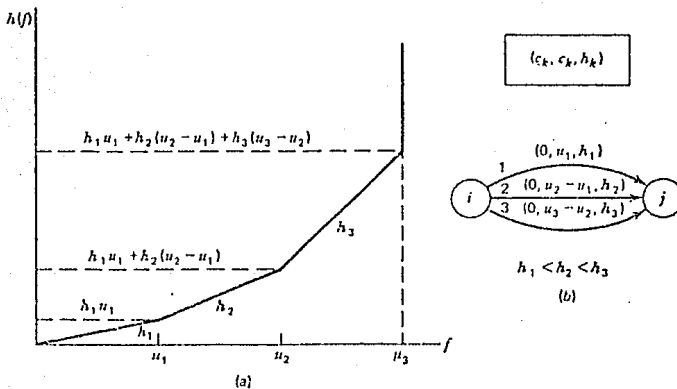


Figura 2.9

Otro Ejemplo de Aproximación por Segmentación Lineal para una Función Convexa.

2.5 APROXIMACION POR SEGMENTACION IMPLICITA

El método de la sección anterior tiene dos desventajas principales.

1. La expansión de la red con la expansión asociada en tiempo de cálculo y espacio (memoria del computador).
2. La inexactitud de la solución inherente en la aproximación.

En esta sección se describe un procedimiento que cubre ambas limitaciones. El mismo no es tan eficiente (tiempo de cálculo) como los procedimientos que se disponen para resolver el problema de flujo a costo mínimo con costos lineales; sin embargo, esto se puede justificar arguyendo que se está resolviendo un problema más difícil.

Como ejemplo considere la Figura 2.10 en la que cada arco tiene una función de costo cuadrática y desde luego convexa

$$h_k = u_k f_k^2$$

Antes de comenzar el procedimiento de segmentación implícita es necesario definir un flujo factible F que satisfaga la conservación de flujo en cada nodo. Defina una cantidad de flujo Δ . El procedimiento consiste esencialmente en efectuar cambios de flujo en múltiplos de Δ de manera que el flujo resultante tenga un costo menor. Para efectuar el cambio de flujo se construye una red de costo marginal que describe los efectos producidos al cambiar el flujo por la cantidad Δ . Esta red marginal incluye un arco hacia adelante y otro reflejado por cada arco en la red original y los costos en ella se definen como sigue

Para arcos hacia adelante

$$h_k = \frac{h_k(f_k + \Delta) - h_k(f_k)}{\Delta} \quad \text{si } f_k + \Delta \leq c_k \quad (8)$$

$$h_k = R \quad \text{si } f_k + \Delta > c_k$$

Para arcos reflejados:

$$\begin{aligned} \underline{h}_k &= \frac{h_k(f_k - \Delta) - h_k(f_k)}{\Delta} && \text{si } f_k - \Delta \geq c_k \\ \underline{h}_k &= R && \text{si } f_k - \Delta < c_k \end{aligned} \quad (9)$$

donde R es un número suficientemente grande.

El costo en el arco hacia adelante es el cambio en el costo unitario causado al incrementarse f_k por Δ , mientras que el costo en el arco reflejado es el cambio en el costo unitario debido a un decremento en f_k por la cantidad Δ . Puesto que $h_k(f)$ es una función convexa.

$$h_k(f_k) - h_k(f_k - \Delta) \leq h_k(f_k + \Delta) - h_k(f_k)$$

Por consiguiente, de las definiciones (8) y (9)

$$h_k + \underline{h}_k \geq 0$$

Cuando el cambio de flujo en un arco pudiera violar su capacidad máxima o mínima, se asigna a h_k o \underline{h}_k un valor muy grande. Esto se hace precisamente para prevenir el cambio de flujo en el arco.

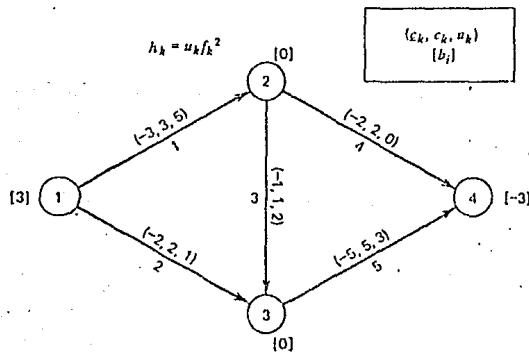


Figura 2.10

Red Ejemplo con Funciones de Costo Cuadráticas

La figura 2.11a muestra una asignación de flujos iniciales para la red de la Figura 2.10 y la Figura 2.11b muestra la correspondiente red de costo marginal para $\Delta=1$. En la última solo se han dibujado los arcos que no tienen asignado un costo. Por ejemplo $f(1)+\Delta=3+1>c_1$, luego, $h_1=R$ y el arco 1 no se dibuja. Lo mismo para h_3 y h_4 .

Un ciclo de arcos en la red marginal implica un ciclo en la red original sobre el cual, es posible cambiar el flujo sin afectar los flujos externos de nodo. Si la red marginal presenta algún ciclo con costo total negativo y se cambian los flujos de los arcos que forman el ciclo, se logrará una disminución en el costo. Por tanto, si no hay ciclos negativos, la solución con el flujo actual es óptima para el valor actual de Δ . Si existe un ciclo negativo, entonces se puede obtener una solución menos costosa cambiando el flujo en el ciclo.

Volviendo al ejemplo, en la Figura 2.11b se detecta el ciclo $M_c = \{-1, 2, -3\}$ con un costo total $-25 + 1 - 2 = -26$. Cambiando el flujo en este ciclo en 1 unidad se obtiene la nueva solución de la Figura 2.11c y su correspondiente red marginal, Figura 2.11d. Esta red marginal tiene un ciclo negativo, $M_c = \{-1, 3, -3\}$, con costo $-15+3+2=-10$ y al cambiar el flujo en este ciclo se obtienen las Figuras 2.11e y 2.11f. La Figura 2.11f no tiene ciclos negativos; por tanto, los flujos de la Figura 2.11e son óptimos para $\Delta=1$.

Cuando se encuentra la solución óptima para un Δ dado se procede a determinar si Δ es suficientemente pequeño. Esto está relacionado con la exactitud deseada en la solución, pues puede hacerse tan pequeño como se quiera. Si se desea una solución entera, el óptimo se obtiene cuando $\Delta=1$. Para obtener un cambio de flujo menor, se divide el actual Δ por 2, se vuelve a calcular los costos de los arcos en la red marginal y se continúa el proceso de buscar ciclos negativos.

En la Figura 2.11g se presenta la red marginal para $\Delta=.5$ obtenida con los flujos de la Figura 2.11e. Como no tiene ciclos negativos tales flujos también son óptimos para este valor. Similarmente se obtiene la Figura 2.11h para $\Delta=0.25$ y de nuevo-

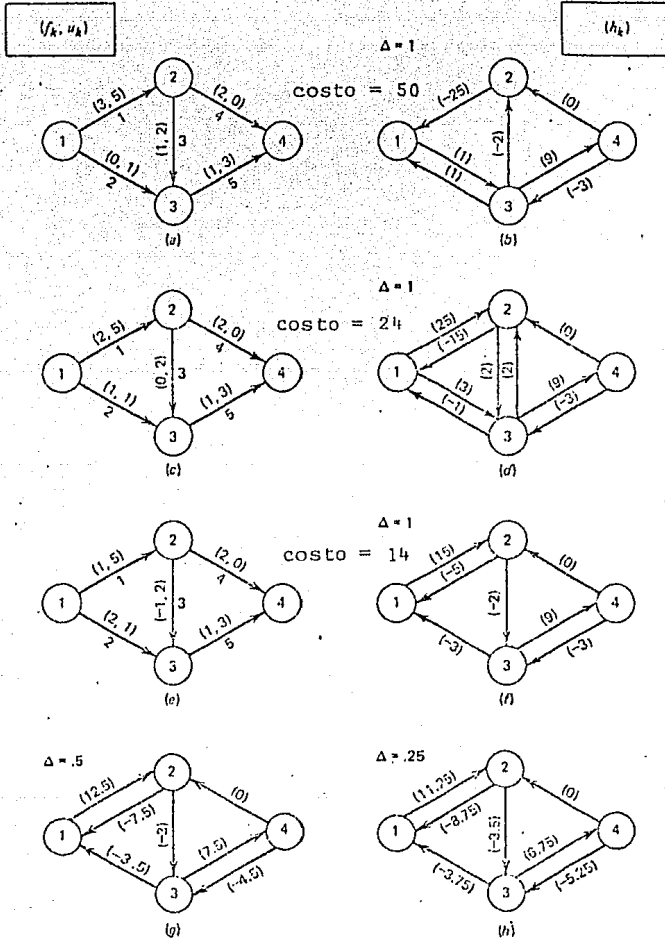


Figura 2.11
Ejemplo de Aplicación de la Aproximación por Segmentación Implícita

se observa que no presenta ciclos negativos. Por tanto los flujos de la Figura 2.11e vuelven a ser óptimos.

Este procedimiento se ha implementado en el algoritmo denominado CONVEX, el cual se describe, al igual que todas las subrutinas necesarias para su ejecución, con un diagrama de flujo en el Apéndice A. Así mismo, se proporciona ahí el algoritmo global NTCNVX (y su correspondiente codificación FORTRAN IV) que lee los datos para el modelo de red, llama a CONVEX e imprime los resultados.

Antes se mencionó que para usar el algoritmo se debe tener a mano un flujo factible inicial. Además de este requisito es necesario proporcionar un árbol de expansión a través del apuntador hacia atrás de los nodos (en la sección 1.7.5 se dió el concepto de apuntador hacia atrás).

Dado que las funciones convexas pueden ser diferentes en cada arco de una red y particulares de cada problema, se hace necesario que el analista o usuario proporcione dos algoritmos (subrutinas en FORTRAN IV): uno que calcule los costos de la red marginal conforme a las definiciones (8) y (9) y otro que calcule cada término de la función objetivo.

2.6 ORGANIZACION Y FORMATO DE DATOS PARA EL PROGRAMA NTCNVX

Cuando el analista haya formulado su problema de optimización como un modelo de redes, es decir, ha enumerado los nodos consecutivamente y determinado los parámetros de nodo y arcos, puede proceder a codificar los datos. En el Cuadro 2.3 se describe totalmente la organización y formato de los datos necesarios para hacer uso del programa NTCNVX.

Cuadro 2.3 CODIFICACION DE DATOS PARA EL PROGRAMA NTCNVX

TARJETA TIPO	COLUMNA	FORMATO	VARIABLE	DESCRIPCION
1	1- 5	I5	N	Número de nodos en la red sin considerar el nodo de holgura.
DATOS DE NODO				
2	1- 5	I5	I	Número asignado al nodo
	6-15	F10.0	BF	Flujo externo fijo del nodo I
	16-25	F10.0	BS	Cota superior del flujo de holgura en el nodo I
	26-35	F10.0	CS	Costo unitario del flujo de holgura en el nodo I
COMENTARIO : Se utilizan una tarjeta de este tipo por cada nodo de la red. Para terminar la lectura de los parámetros de nodo se agrega una tarjeta en blanco al final.				
DATOS DE ARCO				
3	1- 5	I5	I	Nodo origen del arco.
	6-10	I5	J	Nodo terminal del arco.
	11-20	F10.0	LOWER	Cota inferior del arco.
	21-30	F10.0	UPPER	Cota superior del arco.
	31-40	F10.0	COST	Costo unitario del arco o-

Cuadro 2.3 (pag. 2/4)

TARJETA TIPO	COLUMNA	FORMATO	VARIABLE	DESCRIPCION
				término constante (multi- plicador) de la función -- convexa.

COMENTARIO : Se utiliza una tarjeta de este tipo por cada arco de la red. Para indicar el final de la lectura de los datos de arcos se agrega una tarjeta en blanco al final.

FLUJO FACTIBLE INICIAL

4	1-10	F10.0	F(1)	Flujo inicial en el arco 1
	11-20	F10.0	F(2)	Flujo inicial en el arco 2
	---	---	---	-----
	71-80	F10.0	F(8)	Flujo inicial en el arco 8

COMENTARIO : Se utilizan tantas tarjetas de este tipo como sean necesarias para especificar el flujo en cada arco de la red. La condición sobre el flujo factible inicial F es que satisfaga la conservación del flujo en cada nodo. F se debe determinar para la red equivalente que usa un nodo y arcos de holgura para representar los flujos externos de holgura en la red original. Dada la anterior manera de codificar los flujos factibles iniciales, se hace necesario que el analista conozca el método implementado en el programa para enumerar los arcos. Tal método es conceptualmente muy sencillo y se explica en la Sección A.1.1.

ARBOL DE EXPANSION DIRIGIDO

5	1- 5	I5	PB(1)	Apuntador hacia atrás del- nodo 1 en el árbol de ex- pansión.
	6-10	I5	PB(2)	Apuntador hacia atrás del-

Cuadro 2.3 (pag. 3/4)

TARJETA	COLUMNA	FORMATO	VARIABLE	DESCRIPCION
				nodo 2 en el árbol de expansión.
76-80		I5	PB(16)	Apuntador hacia atrás del nodo 16 en el árbol de expansión.

COMENTARIO : Se utilizan tantas tarjetas de este tipo como sean necesarias para especificar el apuntador hacia atrás de cada nodo del árbol, incluyendo el nodo de holgura en caso de ser utilizado.

6	1- 5	I5	SN	Nodo raíz del árbol de expansión mencionado antes.
	6- 15	F10.0	DELMIN	Cambio de flujo mínimo permitido (Δ_{\min}).
	16- 25	F10.0	DELMAX	Cambio de flujo máximo permitido (Δ_{\max}).

OPCIONES DE REPORTE

7	1	I1	ISOLUC	Si ISOLUC $\neq 0$ no se ejecuta ningún cálculo. Se imprime la representación de la red, los flujos factibles iniciales y los apuntadores hacia atrás. Esto permite al analista chequear los datos y confirmar que el flujo inicial correspondiente con el arco. Si ISOLUC = 0, también se imprimen los datos.
	2	I2	IPRINT	0 : Se imprime la solución óptima para cada valor de Δ .

TARJETA	COLUMNA	FORMATO	VARIABLE	DESCRIPCION
TIPO				

2 : Además se imprime cada solución factible como resultado de un cambio de flujo en la red, -- los costos marginales correspondientes y los arcos del ciclo negativo si este existe.

1 : Además se imprime la solución óptima para cada valor de Δ .

Cuadro 2.3 (pag. 4/4)

NOTAS

1. El valor de las variables enteras (FORMATO I) debe ajustarse a la derecha en el campo asignado.
Ejemplo. Si la variable N tiene formato I5 y debe asumir el valor 15 se codifica 15
2. El valor de las variables reales (FORMATO F) se puede colocar en cualquier posición adentro del campo asignado. Si no se codifica el punto, y puesto que todos los formatos incluidos tienen la forma FYY.0, el valor debe ajustarse a la derecha.
Ejemplos. Si el formato es F8.0 entonces 15.2 se puede codificar como 15.2 , 15.2 , ..., 15.2; 75 se puede codificar como 75. , 75. , ..., 75

2.7 APLICACIONES RESUELTAS CON EL PROGRAMA NTCNVX

En esta sección se presentan algunos ejemplos de redes de flujo con costo convexo y se resuelven con el programa NTCNVX. - Con esto se pretende fundamentalmente que el analista se familiarice con el uso del programa, que como se verá , es fácil. En cada aplicación se proporciona la descripción del problema, la red modelo - para resolverlo, la red transformada construida por el programa, la codificación FORTRAN IV de las subrutinas HFUN y COSTOF, la - codificación de los datos y los resultados de la corrida.

2.7.1 Aplicación 1

En la sección 2.5 se resolvió a mano la red ejemplo de la Figura 2.10; ahora se resuelve usando el programa. En este caso, al no haber ningún flujo externo de holgura, el nodo de holgura queda desconectado y, en consecuencia, la red transformada queda igual a la original. En el siguiente cuadro se da la codificación de los datos y en las páginas que siguen los resultados de la corrida según la opción de reportes IPRINT = 2. Las subrutinas HFUN y COSTOF para este problema se proporcionan en la sección 2.8.

Cuadro 2.4

HOJA Nº 1/1

PROGRAMA HTCNVX

CODIFICACION DE DATOS PARA EL PROBLEMA DE LA SECCION 2.7.1

PROGRAMADOR Ing. Parra

	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80
1	4															
1	3															
2	0															
3	0															
4	-3															
2	7	-2			2					0						
2	3	-1			1					2						
1	2	-3			3					5						
1	3	-2			2					1						
3	7	-5			5					3						
3		0			1					2						
0	1		2		4											
1		2	5		1											
02																

RED DE FLUJO CON COSTO CONVEXO

REPRESENTACION DE LA RED TRANSFORMADA

NUMERO DE NODOS = 5 NUMERO DE ARCOS = 5

NODO I	FLUJO EXTERNO FIJO B(I)	POTENCIAL INICIAL PI(I)	APUNTAADOR HACIA ATRAS PB(I)
1	3.00	0.00	0
2	0.00	0.00	1
3	0.00	0.00	2
4	-3.00	0.00	4
5	0.00	0.00	0

ARCO K	N.ORIGEN O(K)	N.TERMINAL T(K)	COTA INF. CL(K)	COTA SUP. C(K)	COSTO H(K)	FLUJO FACTIBLE INICIAL F(K)
1	1	2	-3.00	3.00	5.00	3.00
2	1	3	-2.00	2.00	1.00	0.00
3	2	3	-1.00	1.00	2.00	1.00
4	2	4	-2.00	2.00	0.00	2.00
5	3	4	-5.00	5.00	3.00	1.00

NODO RAIZ DEL ARBOL INICIAL 1 CAMBIO DE FLUJO MINIMO = 0.250
CAMBIO DE FLUJO MAXIMO = 1.000

OPCIONES DE REPORTES : ISOLUC = 0 IPRINT = 2

VALOR DEL CAMBIO DE FLUJO 1.00

ITERACION 1

ARCO K	FLUJO F(K)	COSTO MARGINAL POR INCREMENTO DE FLUJO EN EL ARCO K	COSTO MARGINAL POR DECREMENTO DE FLUJO EN EL ARCO -K
1	3.00	9999.00	-25.00
2	0.00	1.00	1.00
3	1.00	9999.00	-2.00
4	2.00	9999.00	0.00
5	1.00	9.00	-3.00

ARCO DEL CICLO K O -K	FLUJO DESPUES DEL CAMBIO F(K)
-1	2.00
-3	0.00
2	1.00

SOLUCION FACTIBLE (CAMBIO DE FLUJO = 1.0000)

NODO I	POTENCIAL PI(I)	APUNTAOR HACIA ATRAS PB(I)
1	0.00	0
2	5.00	-3
3	3.00	2
4	12.00	5
5	0.00	0

ARCO K	N.ORIGEN O(K)	N.TERMINAL T(K)	COTA INF. CL(K)	COTA SUP. C(K)	COSTO H(K)	FLUJO F(K)	COSTO DEL FLUJO
1	1	2	-3.00	3.00	5.00	2.00	20.00
2	1	3	-2.00	2.00	1.00	1.00	1.00
3	2	3	-1.00	1.00	2.00	0.00	0.00
4	2	4	-2.00	2.00	0.00	2.00	0.00
5	3	4	-5.00	5.00	3.00	1.00	3.00

COSTO TOTAL DEL FLUJO 24.00

ITERACION 2

ARCO K	FLUJO F(K)	COSTO MARGINAL POR INCREMENTO DE FLUJO EN EL ARCO K	COSTO MARGINAL POR DECREMENTO DE FLUJO EN EL ARCO -K
1	2.00	25.00	-15.00
2	1.00	3.00	-1.00
3	0.00	2.00	2.00
4	2.00	9999.00	0.00
5	1.00	9.00	-3.00

ARCO DEL CICLO K O -K	FLUJO DESPUES DEL CAMBIO F(K)
-1	1.00
-3	-1.00
2	2.00

SOLUCION FACTIBLE (CAMBIO DE FLUJO = 1.0000)

NODO I	POTENCIAL PI(I)	APUNTAOR HACIA ATRAS PB(I)
1	0.00	0
2	19998.00	-3
3	9999.00	2
4	10008.00	5
5	0.00	0

ARCO K	N.ORIGEN O(K)	N.TERMINAL T(K)	COTA INF. CL(K)	COTA SUP. C(K)	COSTO H(K)	FLUJO F(K)	COSTO DEL FLUJO
1	1	2	-3.00	3.00	5.00	1.00	5.00
2	1	3	-2.00	2.00	1.00	2.00	4.00
3	2	3	-1.00	1.00	2.00	-1.00	2.00
4	2	4	-2.00	2.00	0.00	2.00	0.00
5	3	4	-5.00	5.00	3.00	1.00	3.00

COSTO TOTAL DEL FLUJO 14.00

SOLUCION OPTIMA PARA EL VALOR DEL CAMBIO DE FLUJO 1.00

NODO I	POTENCIAL PI(I)	APUNTADOR HACIA ATRAS PB(I)
1	0.00	0
2	15.00	1
3	13.00	3
4	22.00	5
5	0.00	0

ARCO K	N.ORIGEN O(K)	N.TERMINAL T(K)	COTA INF. CL(K)	COTA SUP. C(K)	COSTO H(K)	FLUJO F(K)	COSTO DEL FLUJO
1	1	2	-3.00	3.00	5.00	1.00	5.00
2	1	3	-2.00	2.00	1.00	2.00	4.00
3	2	3	-1.00	1.00	2.00	-1.00	2.00
4	2	4	-2.00	2.00	0.00	2.00	0.00
5	3	4	-5.00	5.00	3.00	1.00	3.00

COSTO TOTAL DEL FLUJO 14.00

VALOR DEL CAMBIO DE FLUJO 0.50

SOLUCION OPTIMA PARA EL VALOR DEL CAMBIO DE FLUJO 0.50

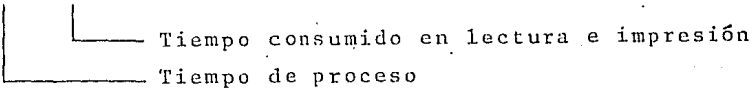
LA HISHA QUE PARA EL CAMBIO DE FLUJO 1.00

VALOR DEL CAMBIO DE FLUJO 0.25

SOLUCION OPTIMA PARA EL VALOR DEL CAMBIO DE FLUJO 0.25

LA HISHA QUE PARA EL CAMBIO DE FLUJO 0.50

PT=1.5 IO=1.1



2.7.2 Aplicación 2

La pérdida de presión en un tubo es dada por la ecuación de Hazen-Williams

$$h = uQ^{1.852}$$

donde h es la pérdida de presión, Q es el flujo en el tubo, y u es una constante que depende del diámetro, longitud, y características del tubo. En términos de redes de flujo la ecuación para el tubo $k(i,j)$ se puede escribir como

$$\Pi_i - \Pi_j = u_k f_k^{1.852}$$

Aquí Π_i se interpreta como la altura en el nodo i y f_k es el flujo en el tubo. La Figura 2.12a muestra una red de tubos con los valores de u_k adyacentes y los flujos que entran y salen de la red. Se desea determinar la solución para los flujos en los tubos y las alturas en los nodos (uniones) si la altura en el nodo 3 debe ser de 45 metros. La cota superior de cada tubo es de $2 \text{ m}^3/\text{seg}$.

Para fijar la altura (variable dual) en un nodo cualquiera s , se agrega a la red un nodo $n' = n+1$ y un arco artificial (n',s) con función de costo lineal y parámetro de costo (u) igual a la altura deseada. El nodo adicional tendrá como parámetros $[b, b_s, h_s] = [0, R, 0]$, donde R es un número grande. Esto se ilustra en la Figura 2.12b. Obsérvese que los parámetros del nodo adicional equivalen a un arco en la red transformada con origen en el nodo de holgura y destino en el artificial y de parámetros $(c, c, h) = (0, R, 0)$.

Ahora bien, si el árbol de expansión inicial de la red transformada tiene raíz en el nodo de holgura, entonces, de la condición 3 del Teorema 3 del capítulo 1 y referenciando a la Figura 2.12c se tiene que

$$\Pi_7 - \Pi_6 = -h_8 \quad \text{para} \quad c_8 < f_8 < c_8$$

y puesto que $\Pi_7 = h_8 = 0$, $\Pi_6 = 0$. Por la misma condición

$$\Pi_6 - \Pi_3 = -h_7$$

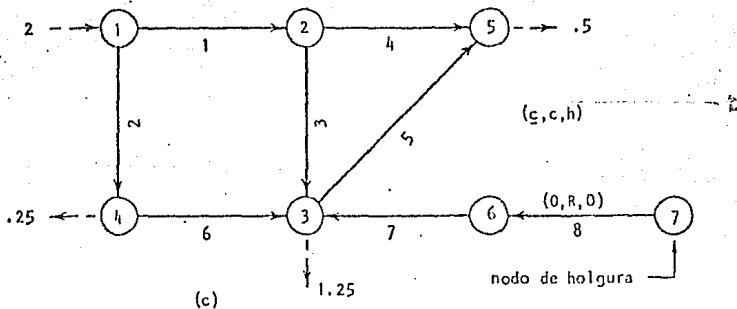
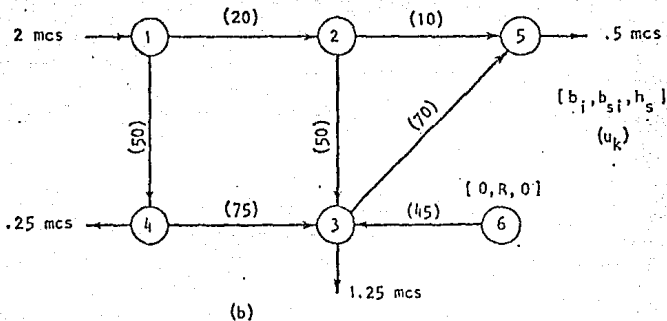
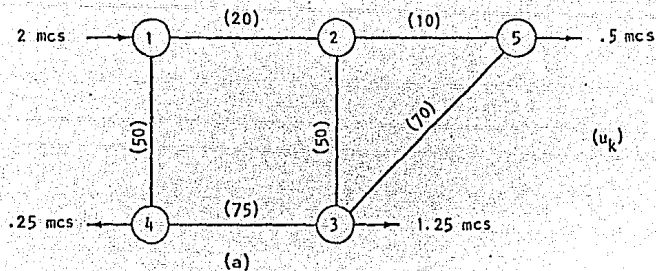


Figura 2.12

(a) Red Física (b) Red Modelo para Fijar en 45 m la Altura del nodo 3. Las Direcciones se han Asignado Arbitrariamente (c) Red Transformada (con nodo y un arco de holgura) Creada por el Programa con Enumeración Creciente por nodo Origen

por tanto, $\pi_3 = h_7 = 45$.

Recordando de la sección 2.3 el término en la función objetivo para un tubo k que lleva fluido es

$$G_k = \frac{u_k |f_k|^{2.852}}{2.852}$$

Por consiguiente, las subrutinas HFUN y COSTOF deben incluir esta fórmula para todos los arcos exceptuando el arco 7 que tiene una función de costo lineal. Tales subrutinas se muestran en la Figura 2.14. En la Figura 2.13 se ilustran el árbol de expansión inicial apropiado y la solución factible inicial para la red transformada. Seguidamente se proporciona la codificación de los datos y los resultados de la corrida ($\Delta_{\min} = 0.05$ y $\Delta_{\max} = 0.4$) con la opción de reportes IPRINT = 1.

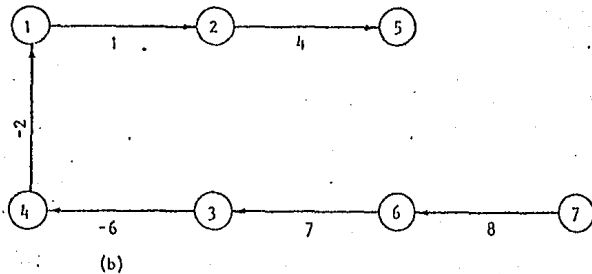
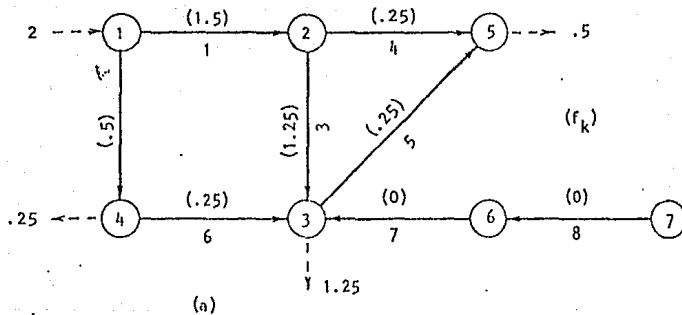


Figura 3.13
 (a) Solución Factible Inicial (b) Arbol de Expansión con Raíz en el Nodo 7


```

10      SUBROUTINE HFUN(K,HPLUS,HMIN,DELTA)
20 %
30      COMMON /L/H(100)/D/F(100)/K/C(100)/N/CL(100)
40 %
50 %      *****
60 %      *
70 %      *      CODIFICACION FORTRAN IV DEL ALGORITMO HFUN      *
80 %      *
90 %      *      PROPOSITO :
100 %     *      CALCULAR LOS COSTOS MARGINALES EN LOS ARCOS *
110 %     *      HACIA ADELANTE Y REFLEJADOS PARA EL PROBLEMA DE LA SEC- *
120 %     *      CION 2.7.2 .
130 %     *
140 %     *****
150 %
160     IF ((F(K) +DELTA) .LE. C(K)) GO TO 10
170     HPLUS = 9999.
180     GO TO 2
190 %
200     10 IF( K .NE. 7 )GO TO 1
210     HPLUS = H(K)
220     GO TO 2
230 %
240     1 HPLUS = H(K)/2.852/DELTA*(ABS(F(K)+DELTA)**2.852
250     *      - ABS(F(K))**2.852)
260 %
270     2 IF ( (F(K) - DELTA) .GE. CL(K)) GO TO 30
280     HMIN = 9999.
290     GO TO 999
300 %
310     30 IF( K .NE. 7 )GO TO 3
320     HMIN = -H(K)
330     GO TO 999
340 %
350     3 HMIN = H(K)/2.852/DELTA*(ABS(F(K)-DELTA)**2.852
360     *      - ABS(F(K))**2.852)
370 %
380     999 CONTINUE
390     RETURN
400     END
410
420
430
440
450
460     SUBROUTINE COSTOF(.K,COSTOK )
470 %
480     COMMON /L/H(100)/D/F(100)/K/C(100)/N/CL(100)
490 %
500 %     *****
510 %     *
520 %     *      CODIFICACION FORTRAN IV DEL ALGORITMO COSTOF      *
530 %     *
540 %     *      PROPOSITO :
550 %     *      CALCULAR EL TERMINO DE LA FUNCION OBJETIVO *
560 %     *      CORRESPONDIENTE AL ARCO K PARA EL PROBLEMA DE SECCION *
570 %     *      2.7.2 .
580 %     *
590 %     *****
600 %
610     IF( K.NE.7 )GO TO 1
620     COSTOK = H(K)*F(K)
630     RETURN
640 %
650     1 COSTOK = H(K)/2.052*ABS(F(K))**2.852
660 %
670     RETURN
680     END

```

Figura 2.14

Cuadro 2.5

PROGRAMA NTCNVX

CODIFICACION DE DATOS PARA EL PROBLEMA DE LA SECCION 2.7.2

HOJA Nº VI

PROGRAMADOR Inq. Parra

0	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80
6																
1	2															
2	0															
3	-1	2	5													
4	0	2	5													
5	0	5	0													
6	0			5		0										
11	2	-2			2		2	0								
11	5	-2			2		5	0								
2	3	-2			2		5	0								
2	5	-2			2		1	0								
4	3	-2			2		7	5								
3	5	-2			2		7	0								
6	3	-2			2		4	5								
11	5		0	5		1	2	5		0	2	5		0		0
-2		11		17		-16		4		8		0				
7	10	5		1	4											
0	11															

RED DE FLUJO CON COSTO CONVEXO

REPRESENTACION DE LA RED TRANSFORMADA

NUMERO DE NODOS = 7 NUMERO DE ARCOS = 8

NODO I	FLUJO EXTERNO FIJO B(I)	POTENCIAL INICIAL PI(I)	APUNTADOR HACIA ATRAS PB(I)
1	2.00	0.00	-2
2	0.00	0.00	1
3	-1.25	0.00	7
4	-0.25	0.00	-6
5	-0.50	0.00	4
6	0.00	0.00	8
7	0.00	0.00	0

ARCO K	N.ORIGEN O(K)	N.TERMINAL T(K)	COTA INF. CL(K)	COTA SUP. C(K)	COSTO H(K)	FLUJO FACTIBLE INICIAL F(K)
1	1	2	-2.00	2.00	20.00	1.50
2	1	4	-2.00	2.00	50.00	0.50
3	2	3	-2.00	2.00	50.00	1.25
4	2	5	-2.00	2.00	10.00	0.25
5	3	5	-2.00	2.00	70.00	0.25
6	4	3	-2.00	2.00	75.00	0.25
7	6	3	-2.00	2.00	45.00	0.00
8	7	6	0.00	3.00	0.00	0.00

NODO RAIZ DEL ARBOL INICIAL = 1 CAMBIO DE FLUJO MINIMO = 0.050
CAMBIO DE FLUJO MAXIMO = 0.400

OPCIONES DE REPORTES : ISOLUC = 0 IPRINT = 1

SOLUCION OPTIMA PARA EL VALOR DEL CAMBIO DE FLUJO = 0.40

NODO I	POTENCIAL PI(I)	APUNTADOR HACIA ATRAS PB(I)
1	0.63	-2
2	33.35	1
3	45.00	7
4	27.02	-6
5	40.86	4
6	0.00	8
7	0.00	0

ARCO K	N.ORIGEN O(K)	N.TERMINAL T(K)	COTA INF. CL(K)	COTA SUP. C(K)	COSTO H(K)	FLUJO F(K)	COSTO DEL FLUJO
1	1	2	-2.00	2.00	20.00	1.10	9.20
2	1	4	-2.00	2.00	50.00	0.90	12.98
3	2	3	-2.00	2.00	50.00	0.45	1.00
4	2	5	-2.00	2.00	10.00	0.65	1.03
5	3	5	-2.00	2.00	70.00	-0.15	0.11
6	4	3	-2.00	2.00	75.00	0.65	7.70
7	6	3	-2.00	2.00	45.00	0.00	0.00
8	7	6	0.00	3.00	0.00	0.00	0.00

COSTO TOTAL DEL FLUJO = 32.82

SOLUCION OPTIMA PARA EL VALOR DEL CAMBIO DE FLUJO 0.20

NODO I	POTENCIAL PI(I)	APUNTAADOR HACIA ATRAS PB(I)
1	5.42	-1
2	33.51	-4
3	45.00	7
4	34.04	-6
5	39.40	5
6	0.00	8
7	0.00	0

ARCO K	N.ORIGEN O(K)	N.TERMINAL T(K)	COTA INF. CL(K)	COTA SUP. C(K)	COSTO H(K)	FLUJO F(K)	COSTO DEL FLUJO
1	1	2	-2.00	2.00	20.00	1.30	14.82
2	1	4	-2.00	2.00	50.00	0.70	6.34
3	2	3	-2.00	2.00	50.00	0.45	1.80
4	2	5	-2.00	2.00	10.00	0.85	2.21
5	3	5	-2.00	2.00	70.00	-0.35	1.23
6	4	3	-2.00	2.00	75.00	0.45	2.70
7	6	3	-2.00	2.00	45.00	0.00	0.00
8	7	6	0.00	3.00	0.00	0.00	0.00

COSTO TOTAL DEL FLUJO 29.09

SOLUCION OPTIMA PARA EL VALOR DEL CAMBIO DE FLUJO 0.10

NODO I	POTENCIAL PI(I)	APUNTAADOR HACIA ATRAS PB(I)
1	0.87	-1
2	31.11	-3
3	45.00	7
4	30.25	2
5	37.73	4
6	0.00	8
7	0.00	0

ARCO K	N.ORIGEN O(K)	N.TERMINAL T(K)	COTA INF. CL(K)	COTA SUP. C(K)	COSTO H(K)	FLUJO F(K)	COSTO DEL FLUJO
1	1	2	-2.00	2.00	20.00	1.30	14.82
2	1	4	-2.00	2.00	50.00	0.70	6.34
3	2	3	-2.00	2.00	50.00	0.55	3.19
4	2	5	-2.00	2.00	10.00	0.75	1.54
5	3	5	-2.00	2.00	70.00	-0.25	0.47
6	4	3	-2.00	2.00	75.00	0.45	2.70
7	6	3	-2.00	2.00	45.00	0.00	0.00
8	7	6	0.00	3.00	0.00	0.00	0.00

COSTO TOTAL DEL FLUJO 29.06

SOLUCION OPTIMA PARA EL VALOR DEL CAMBIO DE FLUJO 0.05

NODO I	POTENCIAL P _i (I)	APUNTAJOR HACIA ATRAS P _B (I)
1	0.97	-1
2	32.34	-4
3	45.00	7
4	28.54	2
5	38.58	5
6	0.00	8
7	0.00	0

ARCO K	N. ORIGEN O(K)	N. TERMINAL T(K)	COTA INF. CL(K)	COTA SUP. C(K)	COSTO H(K)	FLUJO F(K)	COSTO DEL FLUJO
1	1	2	-2.00	2.00	20.00	1.30	14.82
2	1	4	-2.00	2.00	50.00	0.70	6.34
3	2	3	-2.00	2.00	50.00	0.50	2.43
4	2	5	-2.00	2.00	10.00	0.80	1.86
5	3	5	-2.00	2.00	70.00	-0.30	0.79
6	4	3	-2.00	2.00	75.00	0.45	2.70
7	6	3	-2.00	2.00	45.00	0.00	0.00
8	7	6	0.00	3.00	0.00	0.00	0.00

COSTO TOTAL DEL FLUJO 28.93

PT=1.9 IO=1.0

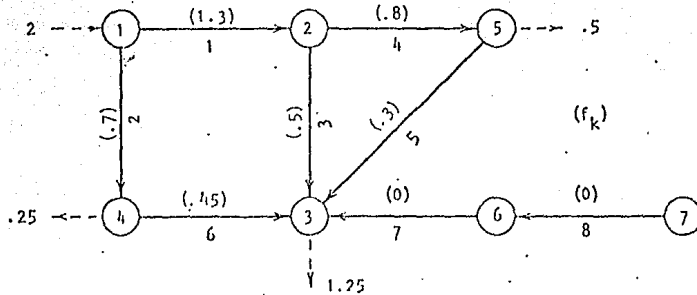


Figura 2.15
 Red con el Flujo Optimo. Nótese que se ha Cambiado la Dirección y Signo del Flujo del arco 5(3,5)

2.7.3 Aplicación 3

Una compañía tiene cuatro plantas de potencia ubicadas en los sitios A, B, C y D a lo largo de un río. Actualmente las plantas se enfrían con el agua del río. Cada planta desecha suficiente calor como para incrementar la temperatura del agua en 10° . El gobierno desea controlar la cantidad de calor descargada en el río por las cuatro plantas de manera que en ningún punto la temperatura del agua exceda 60° . En su fuente, es decir aguas arriba de la planta A, la temperatura es de 50° . El río, por sí mismo, proporciona un enfriamiento con agua helada de 3° de A a B, 5° de B a C, 4° de C a D, y 2° de C a la desembocadura. La temperatura del agua nunca baja de 50° . Otras alternativas para enfriar las plantas tienen un costo por grado de enfriamiento como se muestra adelante, en donde X_i es la cantidad de enfriamiento (en grados) no proporcionado por el río.

Planta	Costo del enfriamiento
A	$10X_A^2$
B	$15X_B$
C	$20X_C$
D	$15X_D^2$

Las Figuras 2.16a y 2.16b muestran, respectivamente, la red modelo para este problema y la red transformada creada por el programa con enumeración creciente por nodo origen. En este caso el flujo representa la temperatura en grados del agua o la cantidad de enfriamiento proporcionado por las otras alternativas. Los nodos 2, 4, 6 y 8 representan a las plantas A, B, C y D, respectivamente. El arco (2,10) representa la alternativa de enfriamiento en la planta A no proporcionado por el río. Este arco tiene una función de costo cuadrática y como cota superior un número grande, $R=100$. La interpretación de los arcos (4,11), (6,12)

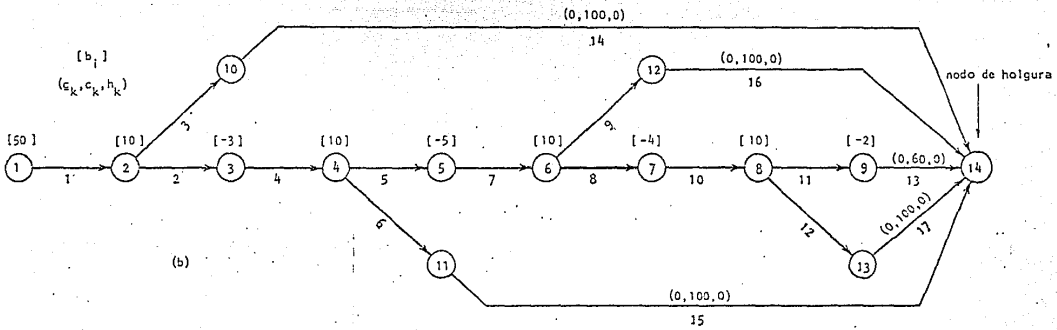
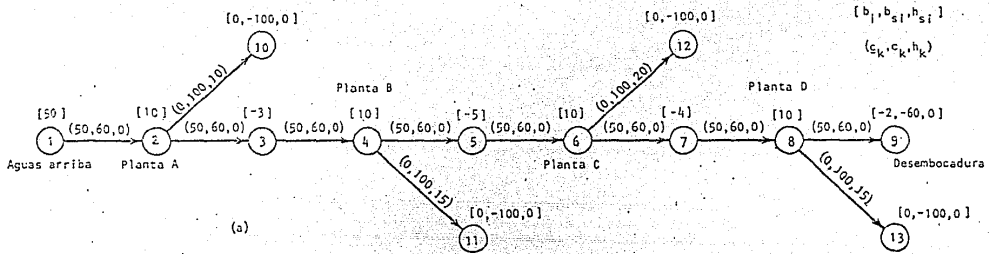


Figura 2.16
 (a) Red Modelo que Representa al Problema (b) Red, con Nodo y Arcos de Holgura, Generada por el Programa. La enumeración es Creciente por nodo Origen (sólo se muestran los parámetros de los arcos de holgura)

y (8,13) es similar. Los arcos (4,11) y (6,12) tienen función de costo lineal y el (8,13) cuadrática. Los nodos 1,3,5,7 y 9 son artificiales y representan el enfriamiento que por sí mismo proporciona el río. La solución de este modelo indicará la cantidad de enfriamiento en las plantas a costo mínimo.

La Figura 2.17 muestra la solución factible inicial en la red transformada y el árbol de expansión apropiado de la misma. En la Figura 2.18 se presentan las subrutinas HFUN y COSTOF apropiadas para este problema y después, en el Cuadro 2.6, la codificación de los datos. Observe que $\Delta_{\min} = 1$ y $\Delta_{\max} = 4$, y puesto que el cambio de flujo para la siguiente resolución es la mitad del anterior, la solución óptima para cada Δ será entera.

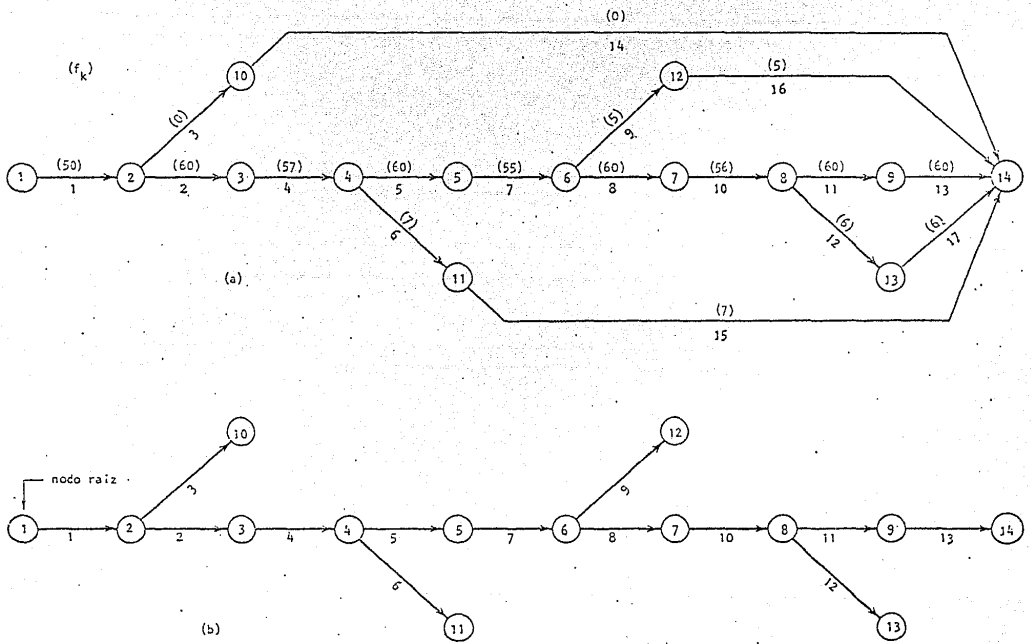


Figura 2.16
 (a) Solución Factible Inicial (b) Arbol de Expansión Enraizado en el Nodo 1

```

10      SUBROUTINE HFUN(K,HPLUS,HMIN,DELFF)
20 %
30      COMMON /L/H(100)/O/F(100)/K/C(100)/N/CL(100)
40 %
50 %      *****
60 %      *
70 %      *      CODIFICACION FORTRAN IV DEL ALGORITMO HFUN      *
80 %      *
90 %      *      PROPOSITO :
100 %     *      CALCULAR LOS COSTOS MARGINALES EN LOS ARCOS *
110 %     *      HACIA ADELANTE Y REFLEJADOS PARA EL PROBLEMA DE LA *
120 %     *      SECCION 2.7.3 ,
130 %     *
140 %     *****
150 %
160     IF ((F(K)+DELFF).LE. C(K)) GO TO 10
170     HPLUS = 9999.
180     GO TO 2
190 %
200     10 IF ( (K.EQ.3).OR.(K.EQ.12) )GO TO 1
210     HPLUS = H(K)
220     GO TO 2
230 %
240     1 HPLUS=(H(K)*(F(K)+DELFF)**2-H(K)*F(K)**2)/DELFF
250 %
260     2 IF ( ( F(K) - DELFF ) .GE. CL(K)) GO TO 30
270     HMIN = 9999.
280     GO TO 999
290 %
300     30 IF ( (K.EQ.3).OR.(K.EQ.12) ) GO TO 3
310     HMIN = -H(K)
320     GO TO 999
330 %
340     3 HMIN= (H(K)*(F(K)-DELFF)**2-H(K)*F(K)**2)/DELFF
350 %
360     999 CONTINUE
370     RETURN
380     END
390
400
410
420
430
440     SUBROUTINE COSTOF( K,COSTOK )
450 %
460     COMMON /L/H(100)/O/F(100)/K/C(100)/N/CL(100)
470 %
480 %     *****
490 %     *
500 %     *      CODIFICACION FORTRAN IV DEL ALGORITMO COSTOF      *
510 %     *
520 %     *      PROPOSITO :
530 %     *      CALCULAR EL TERMINO DE LA FUNCION OBJETIVO *
540 %     *      CORRESPONDIENTE AL ARCO K PARA EL PROBLEMA DE LA SEC- *
550 %     *      CION 2.7.3 ,
560 %     *
570 %     *****
580 %
590     IF (K.EQ.3).OR.(K.EQ.12) )GO TO 1
600     COSTOK = H(K)*F(K)
610     RETURN
620 %
630     1 COSTOK = H(K)*F(K)**2
640 %
650     RETURN
660     END

```

Figura 2.18

Cuadro 2.6

HOJA Nº V2

PROGRAMA NTCNVX

CODIFICACION DE DATOS PARA EL PROBLEMA DE LA SECCION 2.7.3

PROGRAMADOR Ing. Parra

	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80
1	1	5														
2	1	5	0.													
3	1	0.														
4	1	3	-3													
5	1	0.														
6	1	5	-5													
7	1	0.														
8	1	4	-4													
9	1	0.														
10	1	2														
11	1	0.														
12	1	0.														
13	1	0.														
14	1	0.														
15	1	0.														
16	1	0.														
17	1	0.														
18	1	0.														
19	1	0.														
20	1	0.														
21	1	0.														
22	1	0.														
23	1	0.														
24	1	0.														
25	1	0.														
26	1	0.														
27	1	0.														
28	1	0.														
29	1	0.														
30	1	0.														
31	1	0.														
32	1	0.														
33	1	0.														
34	1	0.														
35	1	0.														
36	1	0.														
37	1	0.														
38	1	0.														
39	1	0.														
40	1	0.														
41	1	0.														
42	1	0.														
43	1	0.														
44	1	0.														
45	1	0.														
46	1	0.														
47	1	0.														
48	1	0.														
49	1	0.														
50	1	0.														
51	1	0.														
52	1	0.														
53	1	0.														
54	1	0.														
55	1	0.														
56	1	0.														
57	1	0.														
58	1	0.														
59	1	0.														
60	1	0.														
61	1	0.														
62	1	0.														
63	1	0.														
64	1	0.														
65	1	0.														
66	1	0.														
67	1	0.														
68	1	0.														
69	1	0.														
70	1	0.														
71	1	0.														
72	1	0.														
73	1	0.														
74	1	0.														
75	1	0.														
76	1	0.														
77	1	0.														
78	1	0.														
79	1	0.														
80	1	0.														
81	1	0.														
82	1	0.														
83	1	0.														
84	1	0.														
85	1	0.														
86	1	0.														
87	1	0.														
88	1	0.														
89	1	0.														
90	1	0.														
91	1	0.														
92	1	0.														
93	1	0.														
94	1	0.														
95	1	0.														
96	1	0.														
97	1	0.														
98	1	0.														
99	1	0.														
100	1	0.														

RED DE FLUJO CON COSTO CONVEXO

REPRESENTACION DE LA RED TRANSFORMADA

NUMERO DE NODOS = 14

NUMERO DE ARCOS = 17

NODO I	FLUJO EXTERNO FIJO B(I)	POTENCIAL INICIAL PI(I)	APUNTADOR HACIA ATRAS PB(I)
1	50.00	0.00	0
2	10.00	0.00	1
3	-3.00	0.00	2
4	10.00	0.00	4
5	-5.00	0.00	5
6	10.00	0.00	7
7	-4.00	0.00	8
8	10.00	0.00	10
9	-2.00	0.00	11
10	0.00	0.00	3
11	0.00	0.00	6
12	0.00	0.00	9
13	0.00	0.00	12
14	0.00	0.00	13

ARCO K	N.ORIGEN O(K)	N.TERMINAL T(K)	COTA INF. CL(K)	COTA SUP. C(K)	COSTO H(K)	FLUJO FACTIBLE INICIAL F(K)
1	1	2	50.00	60.00	0.00	50.00
2	2	3	50.00	60.00	0.00	60.00
3	2	10	0.00	100.00	10.00	0.00
4	3	4	50.00	60.00	0.00	57.00
5	4	5	50.00	60.00	0.00	60.00
6	4	11	0.00	100.00	15.00	7.00
7	5	6	50.00	60.00	0.00	55.00
8	6	7	50.00	60.00	0.00	60.00
9	6	12	0.00	100.00	20.00	5.00
10	7	8	50.00	60.00	0.00	56.00
11	8	9	50.00	60.00	0.00	60.00
12	8	13	0.00	100.00	15.00	6.00
13	9	14	0.00	60.00	0.00	60.00
14	10	14	0.00	100.00	0.00	0.00
15	11	14	0.00	100.00	0.00	7.00
16	12	14	0.00	100.00	0.00	5.00
17	13	14	0.00	100.00	0.00	6.00

NODO RAIZ DEL ARBOL INICIAL = 1 CAMBIO DE FLUJO MINIMO = 1.000
 CAMBIO DE FLUJO MAXIMO = 4.000

OPCIONES DE REPORTES : ISOLUC = 0 IPRINT = 1

SOLUCION OPTIMA PARA EL VALOR DEL CAMBIO DE FLUJO 4.00

NODO I	POTENCIAL PI(I)	APUNTAOR HACIA ATRAS PB(I)
1	0.00	0
2	0.00	1
3	25.00	-4
4	25.00	-6
5	25.00	5
6	20.00	-9
7	20.00	8
8	20.00	10
9	40.00	-13
10	40.00	3
11	40.00	-15
12	40.00	-16
13	140.00	12
14	40.00	14

ARCO K	N.ORIGEN B(K)	N.TERMINAL T(K)	COTA INF. CL(K)	COTA SUP. C(K)	COSTO H(K)	FLUJO F(K)	COSTO DEL FLUJO
1	1	2	50.00	60.00	0.00	50.00	0.00
2	2	3	50.00	60.00	0.00	60.00	0.00
3	2	10	0.00	100.00	10.00	0.00	0.00
4	3	4	50.00	60.00	0.00	57.00	0.00
5	4	5	50.00	60.00	0.00	56.00	0.00
6	4	11	0.00	100.00	15.00	11.00	165.00
7	5	6	50.00	60.00	0.00	51.00	0.00
8	6	7	50.00	60.00	0.00	56.00	0.00
9	6	12	0.00	100.00	20.00	5.00	100.00
10	7	8	50.00	60.00	0.00	52.00	0.00
11	8	9	50.00	60.00	0.00	60.00	0.00
12	8	13	0.00	100.00	15.00	2.00	60.00
13	7	14	0.00	60.00	0.00	60.00	0.00
14	10	14	0.00	100.00	0.00	0.00	0.00
15	11	14	0.00	100.00	0.00	11.00	0.00
16	12	14	0.00	100.00	0.00	5.00	0.00
17	13	14	0.00	100.00	0.00	2.00	0.00

COSTO TOTAL DEL FLUJO 325.00

SOLUCION OPTIMA PARA EL VALOR DEL CAMBIO DE FLUJO 2.00

NUDD I	POTENCIAL PI(I)	APUNTADOR HACIA ATRAS PB(I)
1	0.00	0
2	0.00	1
3	5.00	-4
4	5.00	-6
5	5.00	5
6	0.00	-7
7	0.00	8
8	0.00	10
9	20.00	-13
10	20.00	-3
11	20.00	-15
12	20.00	-16
13	30.00	12
14	20.00	14

ARCO K	N.ORIGEN O(K)	N.TERMINAL T(K)	COTA INF. CL(K)	COTA SUP. C(K)	COSTO H(K)	FLUJO F(K)	COSTO DEL FLUJO
1	1	2	50.00	60.00	0.00	50.00	0.00
2	2	3	50.00	60.00	0.00	60.00	0.00
3	2	10	0.00	100.00	10.00	0.00	0.00
4	3	4	50.00	60.00	0.00	57.00	0.00
5	4	5	50.00	60.00	0.00	56.00	0.00
6	4	11	0.00	100.00	15.00	11.00	165.00
7	5	6	50.00	60.00	0.00	51.00	0.00
8	6	7	50.00	60.00	0.00	54.00	0.00
9	6	12	0.00	100.00	20.00	7.00	140.00
10	7	8	50.00	60.00	0.00	50.00	0.00
11	8	9	50.00	60.00	0.00	60.00	0.00
12	8	13	0.00	100.00	15.00	0.00	0.00
13	9	14	0.00	60.00	0.00	60.00	0.00
14	10	14	0.00	100.00	0.00	0.00	0.00
15	11	14	0.00	100.00	0.00	11.00	0.00
16	12	14	0.00	100.00	0.00	7.00	0.00
17	13	14	0.00	100.00	0.00	0.00	0.00

COSTO TOTAL DEL FLUJO 305.00

SOLUCION OPTIMA PARA EL VALOR DEL CAMBIO DE FLUJO 1.00

NODO I	POTENCIAL PI(I)	APUNTADOR HACIA ATRAS PB(I)
1	0.00	0
2	0.00	1
3	0.00	2
4	0.00	4
5	0.00	5
6	-5.00	-9
7	-5.00	8
8	-5.00	10
9	15.00	-13
10	15.00	-14
11	15.00	6
12	15.00	-14
13	15.00	-17
14	15.00	15

ARCO K	N.ORIGEN O(K)	N.TERMINAL T(K)	COTA INF. CL(K)	COTA SUP. C(K)	COSTO H(K)	FLUJO F(K)	COSTO DEL FLUJO
1	1	2	50.00	60.00	0.00	50.00	0.00
2	2	3	50.00	60.00	0.00	59.00	0.00
3	2	10	0.00	100.00	10.00	1.00	10.00
4	3	4	50.00	60.00	0.00	54.00	0.00
5	4	5	50.00	60.00	0.00	55.00	0.00
6	4	11	0.00	100.00	15.00	11.00	165.00
7	5	6	50.00	60.00	0.00	50.00	0.00
8	6	7	50.00	60.00	0.00	55.00	0.00
9	6	12	0.00	100.00	20.00	5.00	100.00
10	7	8	50.00	60.00	0.00	51.00	0.00
11	8	9	50.00	60.00	0.00	60.00	0.00
12	8	13	0.00	100.00	15.00	1.00	15.00
13	9	14	0.00	60.00	0.00	60.00	0.00
14	10	14	0.00	100.00	0.00	1.00	0.00
15	11	14	0.00	100.00	0.00	11.00	0.00
16	12	14	0.00	100.00	0.00	5.00	0.00
17	13	14	0.00	100.00	0.00	1.00	0.00

PT=1.9 IO=1.1

COSTO TOTAL DEL FLUJO 290.00

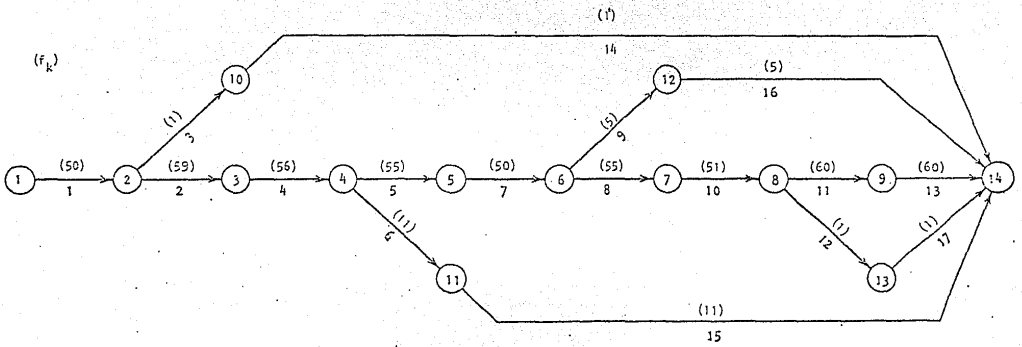


Figura 2.19
Red con el Flujo Optimo

CAPITULO 3

REDES DE FLUJO CON COSTOS CONCAVOS

En el capítulo previo se consideró el problema de redes de flujo cuando el costo en los arcos es una función convexa o lineal del flujo en el arco. Tal problema se trató como una simple variante del caso lineal, y aunque el procedimiento de solución no es tan eficiente como para caso lineal, es posible resolver problemas de redes con varios miles de arcos. Desafortunadamente el problema de redes con costos cóncavos no se resuelve con tanta facilidad. En este caso, la presencia de poco menos de 100 arcos con función de costo cóncava puede resultar en un inaceptable costo computacional para lograr la solución óptima o una cercana a ésta. La razón de este resultado es que para el problema lineal y el problema convexo, un mínimo local también es un mínimo global, y por consiguiente, las pruebas que indican el mínimo local son suficientes para indicar el mínimo global. En cambio, para el problema cóncavo pueden existir muchos mínimos locales, lo cual hace necesario desarrollar algún procedimiento que busque sobre todos estos mínimos para encontrar el mínimo global. Tal búsqueda representa un formidable problema computacional. Todos los algoritmos que resuelven el problema cóncavo conllevan un costo computacional acotado sólo por funciones exponenciales, esto es, el esfuerzo computacional tiene la forma K^{m_c} , donde m_c es el número de arcos con funciones de costo cóncavas y K es alguna constante.

En este capítulo se describe un algoritmo de enumeración implícita adaptado de aquellos usados para resolver los problemas de programación entera binaria. El costo computacional de este algoritmo, que tiene por denominación genérico "branch and bound", es una función de 2^{m_c} . Ciertamente ésta no es una cota deseable, sin embargo, se pueden resolver pequeños pero, impor--

tantes y prácticos modelos de redes con costo cóncavo en los arcos.

3.1 APLICACIONES GENERALES

En la Figura 3.1 se muestra una función cóncava. Cuando se traza una línea recta entre cualesquiera dos puntos de una función cóncava, esta queda enteramente en la función o debajo de ella. Si la función es continua y diferenciable la concavidad se verifica cuando la segunda derivada es negativa en los puntos factibles.

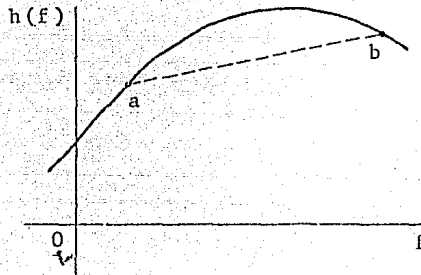


Figura 3.1
Una Función Cóncava

En las Figuras 3.2a y 3.2b se ilustran dos importantes casos. La primera refleja economías de escala, es decir, el costo unitario de producción disminuye cuando aumenta la cantidad producida. Esta figura podría representar el costo de compra de alguna materia prima en la que hay descuentos para cantidades grandes. También podría representar el costo de construcción de una edificación en donde el flujo es el tamaño del mismo. A menudo los costos de construcción presentan economías de este tipo. La segunda figura ilustra una función de costo más idealizada con costo cero para un flujo cero. Cuando el flujo es mayor que cero se incurre en un costo fijo h_F además del costo lineal por unidad de Flujo, h_v . Esta curva podría representar el costo de construcción y operación de un medio de

transporte. Un flujo igual a cero indica que la vía no se construye. En el costo fijo se incluyen los derechos de vía, el costo de diseño y otros costos independientes del tamaño. Los costos que dependen del tamaño se representan por la porción lineal de la curva.

La curva de la Figura 3.2b es la apropiada para representar el costo de reaprovisionamiento de un inventario. En este caso, el flujo es la cantidad de producto ordenado. El costo fijo es el costo que significa hacer el pedido; mientras que el costo variable es el costo unitario del producto.

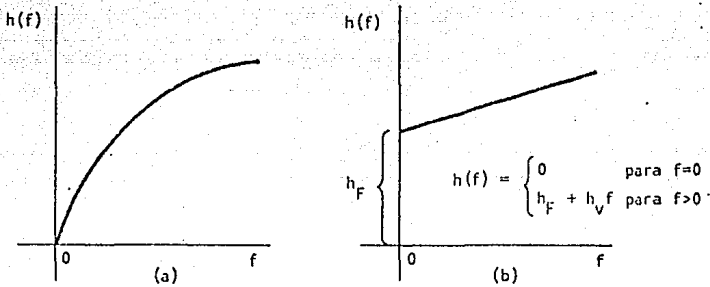


Figura 3.2

Ejemplos de Funciones de Costo Cóncavas
(a) Economías de Escala (b) Cargo Fijo

En este capítulo se consideran explícitamente solo curvas de costo como la que aparece en la Figura 3.2b. Esto no significa que se descarten de consideración curvas como la dada en la Figura 3.2a. Sucede que es posible representar tales curvas mediante el uso de arcos múltiples y aproximación por segmentación lineal, lo cual, se ilustra en la Figura 3.3. En el ejemplo se requieren tres arcos para proporcionar la aproximación a la curva. Si solo se incluyeran dos arcos, el modelo permitiría flujos con costo en la línea punteada. Por ejemplo, $f_1=0$, $f_2=c_1/2$ es una solución factible, sin embargo, tal solución nunca estará presente en una solución óptima puesto que existe otro flujo factible con costo menor, $f_1=c_1/2$ y $f_2=0$. Ob-

serve que el modelo por segmentación permite una capacidad total de c_1+c_2 mientras que la capacidad del arco original es c_2 . Para evitar esta posibilidad, se agrega un tercer arco con capacidad c_2 en serie con la segmentación.

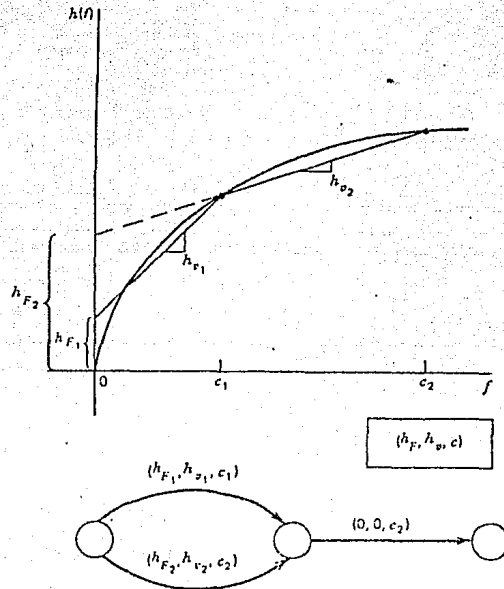


Figura 3.3

Aproximación por Segmentación Lineal para una Función Cóncava

Ahora, considere la función de costo

$$h(f) = h_F + h_V f^2 \quad 0 \leq f \leq c_2$$

Esta función tiene un cargo fijo y una función de costo convexa asociada con el flujo. En la Figura 3.4 se ilustra la correspondiente aproximación por segmentación lineal. Se recuerda que en la Sección 2.4 se describió el procedimiento general para aproximar funciones convexas por segmentación lineal.

TESIS NO DEBE
 SER
 DE
 BIBLIOTECA
 (h_F, h_V, c)

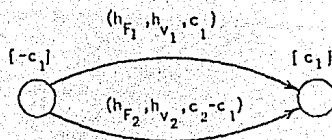
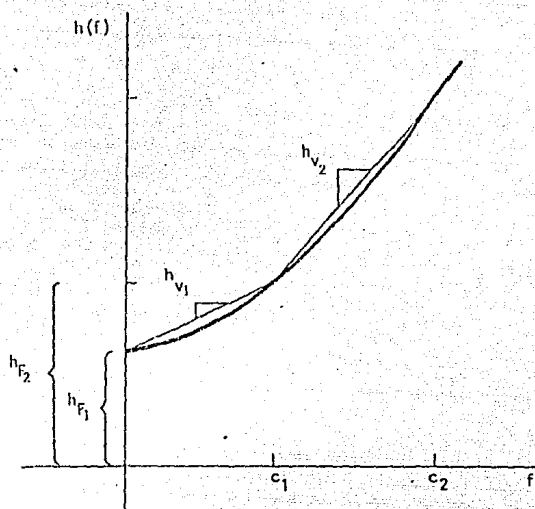


Figura 3.4

Aproximación por Segmentación Lineal para una Función Convexa con Cargo fijo

La figura 3.5 ilustra el bien conocido problema de la ubicación de plantas. Este problema se refiere a la construcción de las plantas industriales de una entidad que comercializa un producto en una región donde todavía no tiene ninguna instalación. Conociendo la demanda de este artículo procedente de cada ciudad de esta región, es necesario determinar el número de plantas de producción que habrían de construirse, así como su ubicación y tamaño, minimizando:

- los costos de producción y
- los costos de transporte desde cada ubicación a los depósitos de los clientes.

En la Figura 3.5, los nodos de la derecha representan a los clientes y su flujo externo, que es negativo, representa precisamente la demanda de los mismos. Los arcos que terminan en estos nodos representan los medios de transporte con costo lineal y capacidad restringida. Los arcos que se originan en el nodo fuente representan los sitios potenciales para las plantas industriales o edificaciones. Los valores de h_F y h_V en es

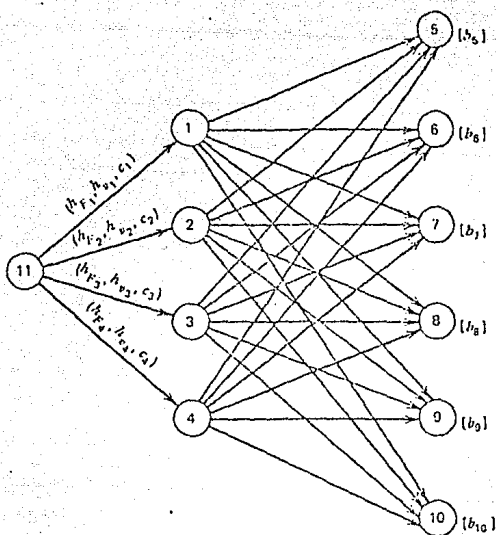


Figura 3.5

Red de Flujo para el Problema de la Ubicación de Plantas

te caso, son los costos fijos y los costos variables de construcción y operación de cada sitio. La capacidad en el arco es el tamaño máximo propuesto de la planta. Un algoritmo de flujo a costo mínimo podría seleccionar los sitios donde construir y los tamaños de las plantas de manera que el costo total sea mínimo. También se podrían incluir en el modelo arcos de transporte con cargo fijo para considerar el costo de crear el medio de transporte.

Problemas de diseño de redes físicas en los que es necesario seleccionar los arcos que deben incluirse en la red, pueden describirse usando arcos con costo cóncavo. Tales problemas podrían surgir en el diseño de redes de transporte y redes de tubos.

Los modelos para el control de la producción, en los cuales el flujo representa la cantidad producida, son más exactos si se modelan con costos cóncavos. En este caso los costos de establecer la producción o reordenar inventarios se modelan con cargo fijo.

3.2 NOTACION

Dada la red $D = [N, M]$, se identificará un subconjunto K_c de m_c arcos, es decir, $K_c = \{k_c(1), k_c(2), \dots, k_c(m_c)\}$ donde $k_c(i)$ es el índice del arco en la descripción de la red modelo de trabajo, la cual se describe en la Sección B.1.

Para cada arco cóncavo $k_c(i)$ se identifica un costo fijo $h_F(i)$ y un costo variable $h_V(i)$. Por consiguiente, la función de costo para los arcos con índice $k=k_c(i)$ es

$$h_k(f_k) = \begin{cases} 0 & f_k=0 \\ h_F(i) + h_V(i) \cdot f_k & 0 < f_k \leq c_k \end{cases}$$

donde la capacidad del arco c_k debe ser finita.

3.3 ENUMERACION EXHAUSTIVA

En el procedimiento de enumeración exhaustiva se asocia la variable x_i con el i -ésimo arco cóncavo. Se dice que el i -ésimo arco está "cerrado", y no se permite flujo en él, si $x_i = 0$. Si $x_i = 1$, entonces el i -ésimo arco está abierto, es decir, se permite un flujo entre 0 y c_k , donde $k = k_c(i)$. El vector X es de la forma $[x_1, x_2, \dots, x_{m_c}]$.

Ahora bien, definidas las variables X se puede escribir el problema de costo cóncavo como un problema de programación entera binaria $(0,1)$.

$$\sum_{i=1}^{m_c} h_{\bar{v}}(i)x_i + \sum_{k=1}^m h_k f_k \quad (1)$$

$$h_k = h_{\bar{v}}(i) \text{ para } k = k_c(i)$$

$$\text{s.a. } Af = b$$

$$0 \leq f_k \leq c_k \text{ para } k \in M$$

$$f_k \leq x_i c_k \text{ para } i = 1, \dots, m_c \text{ y } k = k_c(i)$$

$$x_i = 0, 1 \text{ para } i = 1, \dots, m_c$$

Las restricciones que no contienen las variables X representan la conservación de flujo y acotamiento del problema de redes original. Las restricciones $f_k \leq x_i c_k$ para arcos cóncavos fuerzan al flujo variable f_k a ser cero cuando $x_i = 0$ y establecen la cota superior cuando $x_i = 1$.

Una técnica de solución para este problema es una enumeración exhaustiva de todas las posibles realizaciones de X . Una realización o solución particular al problema se logra asignando un 1 o 0 a cada x_i . Si se enumeran todas las posibles asignaciones y se evalúa cada una, se obtiene la solución óptima es cogiendo la que tenga el valor más pequeño. Sin embargo, el número de asignaciones posibles es de 2^{m_c} y éste es muy grande --

aun para valores moderados de m_c .

El proceso de enumeración exhaustiva es conceptualmente fácil de adquirir. Para un problema de tres variables hay $2^3=8$ posibles soluciones que se obtienen listando todos los posibles números binarios de tres dígitos.

$$\begin{array}{l} (x_1, x_2, x_3) \\ (0 \ 0 \ 0) \\ (0 \ 0 \ 1) \\ (0 \ 1 \ 0) \\ (0 \ 1 \ 1) \\ (1 \ 0 \ 0) \\ (1 \ 0 \ 1) \\ (1 \ 1 \ 0) \\ (1 \ 1 \ 1) \end{array}$$

En el algoritmo para computador este proceso de enumeración se ejecuta construyendo un árbol binario como se ilustra en la Figura 3.6 donde los círculos se denominan vértices y los segmentos de línea ramas. Cada vértice representa un conjunto de realizaciones del vector X . El único vértice al nivel 0 representa el conjunto de todas las realizaciones (8 en el ejemplo). Los vértices del nivel 1 se forman haciendo una decisión binaria en una de las variables de X (x_1 en la ilustración). La rama de la izquierda representa todas las soluciones con $x_1=1$. Esto se representa por el símbolo 1 en el vértice 9. La rama de la derecha representa todas las soluciones con $x_1=0$. El símbolo $\bar{1}$ en el vértice 2 representa esta decisión. Ramificando en el vértice 9 sobre la variable x_2 se obtienen los vértices 13 y 10. El vértice 13 representa todas las realizaciones con $x_1=1$ y $x_2=1$. De esta manera, el proceso de ramificación continúa hasta generar todos los vértices del último nivel en el árbol, los cuales, representan las ocho soluciones individuales del problema.

Cada vértice del árbol se puede identificar por una secuencia de números que indican las decisiones binarias hechas -

para alcanzar el vértice. Los números en la secuencia representan las variables fijadas. Si en la secuencia aparece un 1 indica que la variable x_1 ha sido fijada con valor 1 ; mientras -- que si aparece un $\bar{1}$ indica que x_1 fue fijada con valor 0 . Las variables libres, es decir, las que no tienen asignado un valor no aparecen en la secuencia. Por ejemplo, el vértice 6 de la Figura 3.5 puede ser referenciado por la secuencia $\bar{1}2$, indicando que $x_1=0$, $x_2=1$ y x_3 libre.

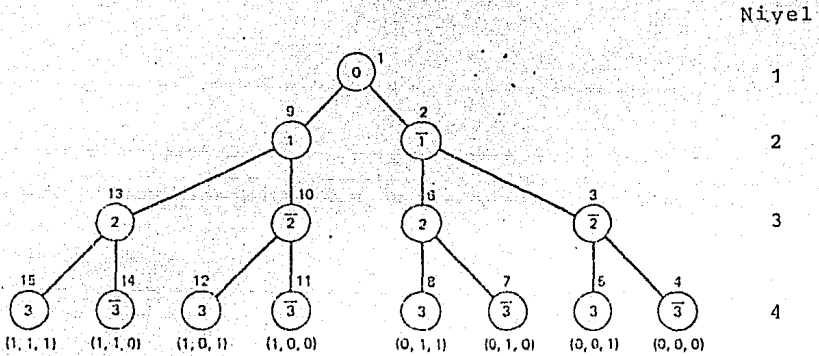


Figura 3.6

Árbol Binario con Variables Fijadas en Orden Natural

Al proceso de crear dos nuevos vértices a partir de otro se le llama la separación. Si una secuencia $\{s_1\}$ identifica un vértice del árbol al nivel 1, entonces a los dos vértices nuevos se les asigna las secuencias $\{s_1, 1\}$ y $\{s_1, \bar{1}\}$. Es claro que este proceso separa el conjunto de soluciones definidas por $\{s_1\}$ en dos subconjuntos mutuamente exclusivos que juntos comprenden el conjunto definido por s_1 .

No hay necesidad de explorar las variables en orden numérico o que el orden sea el mismo para todas las ramas del árbol. La Figura 3.7 ilustra una posible alternativa.

El algoritmo para computadora explora los vértices sin guardar explícitamente el árbol en la memoria. Para hacerlo de esta manera, se necesita una notación que describa la trayecto-

ria (todas las trayectorias se originan en el vértice 0) del vértice en el proceso dinámico de enumeración. Esto se logra usando el símbolo subraya junto con la notación de secuencia. Si el símbolo $\underline{1}$ aparece en la secuencia indica que $x_1=1$ y que la rama alternativa donde $x_1=0$ ya ha sido explorada. El símbolo $\bar{1}$ indica que $x_1=0$ y que la rama alternativa para $x_1=1$ ya ha sido explorada. Para explorar el árbol de la Figura 3.7 en el orden definido por los números de los vértices se necesitarían las siguientes secuencias.

vértice	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
secuencia	0	3	32	321	32 $\bar{1}$	32 $\bar{2}$	32 $\bar{1}$	32 $\bar{1}$	3 $\bar{1}$	3 $\bar{1}$	3 $\bar{1}$ 2	3 $\bar{1}$ 2	3 $\bar{1}$	3 $\bar{1}$ 2	3 $\bar{1}$ 2

Las notaciones de secuencia y subraya permite que el árbol se investigue en cualquier orden con la disposición de que una trayectoria sea extendida a su nivel más alto antes de explorar otra. A este procedimiento se le llama *rastreo hacia atrás* ("back track") debido a que la trayectoria se extiende lo más lejos posible y luego el proceso rastrea el último vértice que tiene una rama sin explorar. El algoritmo que seguida-

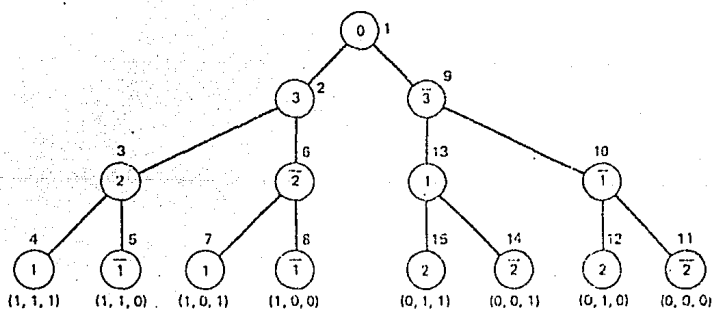


Figura 3.7

Un Arbol Binario con Variables Fijadas en Orden Arbitrario

mente se presenta permite explorar el árbol de esta manera.

ALGORITMO XHAUST

PROPOSITO: Controlar las operaciones del procedimiento de enumeración exhaustiva que resuelve el problema de redes de flujo a costo mínimo con -- cargo fijo.

1. INICIAR. Comenzar con una secuen-- cia vacía. Hacer $ZB := \infty$ y $XB := 0$ con todas las variables libres.
2. SEPARACION. Si no hay variables li bres, ir al paso 3. En caso contrari o, escoger una variable libre i y determinar qué valor se le puede asignar. Si se le asigna el valor 1, agregar i a la secuencia. Si se le asigna un 0 agregar \bar{i} a la se-- cuencia. Repetir el paso 2.
3. EVALUACION. Todas las variables X- están fijas y se encuentra la soluci ón mediante el llamado de la subru rutina KILTER. Determinar el valor de la función objetivo Z para X; - ir al paso 4.
4. REEMPLAZO. Si Z es menor que ZB reem emplazar ZB por Z ($ZB := Z$) y XB por X ($XB := X$). Ir al paso 5.
5. RASTRE ATRAS. Ir de derecha a iz-- quierda en la secuencia y borrar - todos los símbolos subrayados hasta encontrar un símbolo no subrayado; subrayarlo y cambiar el estado del símbolo (i se convierte en \bar{i} , - \bar{i} se convierte i). Ir al paso 2. - Si todos los símbolos en la secuenci

cia están subrayados, terminar; la enumeración se han completado. La solución óptima es XB con valor ZB.

Este algoritmo usa XB para guardar "la mejor solución actual" ("incumbent solution") en el proceso. ZB es el valor de la función objetivo para la mejor solución actual, la cual se conserva hasta encontrar otra mejor. Cuando el algoritmo termina ZB es la solución óptima.

3.4. ENUMERACION IMPLICITA

El algoritmo de enumeración exhaustiva no es práctico en la mayoría de los casos, ya que el número de posibles soluciones es muy grande aún para problemas relativamente pequeños (cerca de 1,000 para $m_c=10$; 1,000,000 para $m_c=20$). Por esta razón sería muy conveniente contar con un procedimiento (prueba) que permita "sondear" algunos vértices del árbol sin enumerarlos. Sondear un vértice es descartarlo sin previa evaluación en la exploración de una solución. Debido a que los vértices sondeados son en cierto sentido enumerados sin ser generados, se llama a este proceso *enumeración implícita*.

En esta sección se considera una prueba de sondeo que usa una cota inferior sobre la función objetivo. Suponga que se dispone de una solución factible X_B con valor Z_B . El vértice s_1 representa algún conjunto de soluciones factibles que pueden ser enumeradas e identificadas asignando valores a las variables libres en cada combinación posible. Por algún método se obtiene una cota inferior Z_{LB} (solución con variables continuas en el vértice s_1) sobre todas las soluciones de este conjunto. Si sucede que

$$Z_{LB} \geq Z_B$$

entonces todas las soluciones en el conjunto pueden ser juzgadas no óptimas o al menos no mejores que la solución que se tiene a mano. Por tanto, todos los vértices del árbol que siguen al vértice bajo consideración pueden ser

enumerados implícitamente. Por otra parte, si $Z_{LB} < Z_B$ los vértices no pueden ser enumerados implícitamente.

Ahora bien, la eficiencia de esta prueba depende de dos cosas:

1. La calidad de la cota inferior.
2. La sensibilidad del valor de la función objetivo a -- las variables de decisión en X.

La calidad de la cota inferior normalmente es relativa a la dificultad computacional en obtenerla. En muchos problemas, una cota inferior "pobre" se puede obtener con facilidad; mientras que una "buena" cota inferior solo con considerable esfuerzo computacional.

La sensibilidad del valor de la función objetivo a cambios en los valores de las variables de decisión es una característica del problema que se resuelve. Aquellos problemas para los cuales el valor óptimo de Z es significativamente menor que otros valores con otras variables de decisión, tendrán un aumento en la eficacia de la prueba de acotamiento. Si hay muchas soluciones con valores de Z cercanos, la prueba no será tan eficaz.

Los procedimientos de enumeración implícita permiten al analista resolver problemas prácticos, pero no tienen la capacidad para resolver problemas de cientos de variables que sí pueden resolver los algoritmos para redes con costos lineales o -- convexos. Esta es la razón de la gran diferencia entre la dificultad de resolver problemas de redes de flujo con costos cóncavos y con costos lineales o convexos.

Ahora se fortalece la prueba de eliminación de vértices mediante el uso de un parámetro que especifica el porcentaje de desviación permitido con respecto al óptimo. Si PER representa tal porcentaje, entonces la prueba de sondeo es:

Si $Z_{LB} \geq Z_B - \frac{PER}{100} |Z_B|$ entonces descartar el vértice bajo consideración.

El uso de esta prueba puede descartar (o desviarse dema-

siado de) la solución óptima. Sin embargo, se sabe que la solución resultante está dentro de un porcentaje, PER, de la óptima. Esto a menudo es un compromiso útil para obtener una "buena" solución con un costo computacional aceptable, aunque no sea necesariamente la óptima. Una posible estrategia es resolver el problema con valores de PER sucesivamente menores hasta que se asegure la solución óptima (PER = 0) o se acabe el presupuesto para las corridas en la computadora.

3.5 DETERMINACION DE UNA COTA INFERIOR

En un paso intermedio del proceso de enumeración, ciertas de las variables de decisión X se hacen igual a 1 y ciertas igual a 0; el resto de las variables son libres. Por conveniencia se define W^+ como el conjunto de variables fijadas en 1, W^- como el conjunto de variables fijadas en 0, y W^0 como el conjunto de variables libres. Siguiendo esta notación, el modelo matemático del problema en su paso intermedio tiene la forma

$$\text{Min.} \quad \sum_{i \in W^+} h_F(i) + \sum_{i \in W^0} h_F(i)x_i + \sum_{k=1}^m h_k f_k \quad (2)$$

$$h_k = h_v(i) \quad \text{para } k = k_c(i) \quad i=1, \dots, m_c$$

$$\text{s.a} \quad Af = b \quad (2.1)$$

$$0 \leq f_k \leq c_k \quad \text{para } k \in M \quad (2.2)$$

$$f_k \leq x_i c_k \quad \text{para } i \in W^0, \quad k = k_c(i) \quad (2.3)$$

$$f_k = 0 \quad \text{para } i \in W^-, \quad k = k_c(i) \quad (2.4)$$

$$x_i = 0, 1 \quad \text{para } i \in W^0 \quad (2.5)$$

Note que en los arcos cerrados el flujo se fuerza a ser cero, y que la suma de los cargos fijos para los arcos abiertos aparece como un término constante en la función objetivo.

La cuestión sería ahora obtener una cota inferior para todas las soluciones asociadas con W^0 . Para lograrla, primero se elimina la condición de que x_i sea entera y se reemplaza por

$$0 \leq x_i \leq 1 \quad \text{para } i \in W^0 \quad (2,6)$$

Al nuevo problema que resulta de reemplazar (2.5) por (2.6) se le llama la *relajación* del problema (2) debido a que la condición de integralidad ha sido "relajada". Puesto que la región factible se ha extendido con la relajación, el valor de la función objetivo del problema relajado debe ser una cota inferior de la solución del problema entero.

Ahora, note que en la solución óptima siempre se tendrá que

$$x_i = \frac{f_k}{c_k} \quad \text{para } i \in W^0, \quad k = k_c(i)$$

Esto es cierto debido a que el coeficiente para x_i es positivo en la función objetivo y x_i solo aparece en la restricción simple que limita a f_k . Por tanto se puede reemplazar x_i usando la relación $x_i = f_k / c_k$. De esta manera, el coeficiente para f_k puede expresarse como

$$h_k = \frac{h_F(i)}{c_k} + h_v$$

Usando este resultado, se obtiene la formulación matemática del problema relajado.

$$Z_{LB} = \text{Min.} \quad \sum_{i \in W^+} h_F(i) + \sum_{k=1}^m h_k f_k \quad (3)$$

donde $h_k = h_v(i)$ para $k = k_c(i), i \in W_i^+$

$$h_k = \frac{h_f(i)}{c_k} + h_v(i) \quad \text{para } k = k_c(i), i \in W_i^0$$

$$\text{s.a.} \quad Af = b$$

$$0 \leq f_k \leq c_k \quad \text{para } k \in M$$

$$f_k = 0 \quad \text{para } k \in k_c(i), i \in W_i^-$$

Este es un problema de redes de flujo con costos lineales que se puede resolver con el algoritmo SIMPLEX o el algoritmo de desviaciones, el cual se presenta en el Apéndice B.

En la Figura 3.8 se recoge esquemáticamente la relación entre el costo linealizado para f_k en la relajación y la función de costo cóncavo a la cual reemplaza. La función de costo lineal es exacta solo en los puntos extremos, $f_k=0$ y $f_k=c_k$; mientras que en los demás puntos, la función lineal es una subestimación de la función cóncava. Esto es una justificación adicional al hecho de que la solución del problema relajado proporciona una cota inferior a la solución del problema original de programación entera mixta.

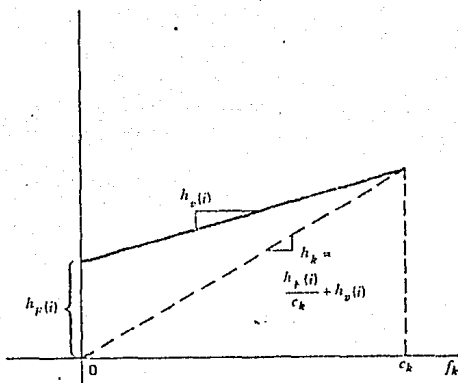


Figura 3.8

Subestimación Lineal para un Arco
con Función de Costo Cóncava

El algoritmo que se describe verbalmente en la sección que sigue, usa una forma ligeramente diferente del problema-relajado (3), en la cual se reemplaza la restricción

$$f_k = 0 \quad \text{para } k = k_c(i), i \in W^-$$

por

$$\begin{cases} h_k = R \\ 0 \leq f_k \leq c_k \end{cases} \left\{ \begin{array}{l} \text{para } k = k_c(i), i \in W_i^- \end{array} \right.$$

donde R es un número positivo muy grande

Con un costo muy grande en los arcos cerrados, el flujo en estos será distinto de cero solo si tal flujo es necesario para forzar la solución factible. Entonces, el algoritmo debe chequear los flujos en los arcos cerrados. Si se detecta alguno mayor que cero, se "enciende" un indicador de no factibilidad y se sondea (descarta) el vértice. Este procedimiento tiene la ventaja de que durante el proceso de enumeración sólo se cambian los costos en los arcos, el conjunto de restricciones nunca es cambiado.

Obtenida la solución óptima al problema relajado es posible redondearla hacia una solución factible del problema original. Esto se logra haciendo

$$\begin{cases} x_i = 1 & \text{si } f_k > 0 \\ x_i = 0 & \text{si } f_k = 0 \end{cases} \left\{ \begin{array}{l} \text{para } i \in W^0 \text{ y } k = k_c(i) \end{array} \right.$$

Es claro que las variables fijadas ya tienen un valor de 1 o 0. La solución redondeada se representa por XR y el correspondiente valor objetivo por ZR, donde

$$ZR = \sum_{i=1}^{m_c} h_F(i) x_i + \sum_{k=1}^m h_k f_k$$

y

$$h_k = h_v(i) \text{ para } k = k_c(i)$$

Cuando la solución redondeada es mejor que la mejor solución actual ($ZR < ZB$), ésta se reemplaza por la redondeada.

3.6 ALGORITMO DE ENUMERACION IMPLICITA

En la presente sección se recogen las líneas generales del algoritmo de enumeración implícita. En este algoritmo se desarrollan tres vectores para rastrear la enumeración de varia

bles en el árbol. Ya se estableció la notación en la sección 3.3, ahora se define la estructura de datos para manejarla.

P : Define la secuencia de variables en el árbol. $P(L)=J$ indica que la variable J se considera en el nivel L .

W : Se usa para rastrear los valores asignados a las variables fijadas.

$W(i) > 0$ si x_i se fija en 1

$W(i) = 0$ si x_i es libre

$W(i) < 0$ si x_i se fija en 0

De esta manera, W reemplaza a los conjuntos W^+ , W^- , y W^0 -- usados previamente. El valor numérico de $W(i)$ es el nivel de fijación de la variable.

U : Contiene la información de las variables subrayadas. Si $U(L)=1$, el correspondiente elemento en $P(L)$ se subraya. Si $V(L)=0$, no se subraya

Ejemplo: Para el vértice 11 de la Figura 3.7 se tiene

Secuencia	P	W	U
<u>3</u> <u>1</u> <u>2</u>	3 1 2	-2 -3 -1	1 0 0

ALGORITMO ENUMER

PROPOSITO: Controlar las operaciones del procedimiento de enumeración implícita que resuelve el problema de redes de flujo a costo mínimo con -- cargo fijo.

1. INICIAR. Hacer las aproximaciones lineales iniciales para todos los arcos con cargo fijo. Inicializar en cero el indicador de nivel -- (L=0) y los vectores de secuencia P, de valores asignados a las variables W y el de variables subrayadas U.

2. SONDEO. Resolver el problema relajado (es decir, con las aproximaciones lineales) asociado con el -- vértice del árbol de soluciones. -- Si el valor objetivo del problema relajado, Z (cota inferior al conjunto de soluciones del nodo), es mayor o igual a la mejor solución actual, $ZB - PER * |ZB| / 100$, ir al paso 4. Por el contrario, es decir, -- si Z es menor que la mejor solución actual encontrar la solución redondeada XR y su valor objetivo asociado ZR. Si ZR es mayor o -- igual que ZB, ir al paso 3. Si es menor, actualizar la mejor solución actual haciendo $ZB = ZR$. Si ZR es igual a Z, no se podrá obtener una mejora; ir al paso 4. En caso contrario, ir al paso 3.

3. SEPARACION. Encontrar la variable libre a separar (J). Si no hay ninguna separable, ir al paso 4. En caso contrario, avanzar al siguiente nivel en el árbol ($L=L+1$). Agregar la variable a la secuencia -- [$P(L)=J$]. Hacer cero la subraya para este nivel [$U(L)=0$]. Asignar a la variable su valor (0 ó 1) en el nivel y regresar al paso 2.
4. RASTREAATRAS. Si la subraya para el nivel es igual a 0, encontrar la variable J [$J=P(L)$] del nivel, -- asignar un 1 a la subraya de este nivel [$U(L)=1$], encontrar el nuevo valor para J e ir al paso 5. Si la subraya es 1, subir un nivel en el árbol [$L=L-1$]. Si se encuentra el nivel 0, es decir, si el nuevo valor de L es 0, terminar; se ha completado el proceso. Si no se encuentra el nivel 0, repetir el proceso 4.
5. CAMBIO. Liberar todas las variables fijadas en un nivel igual o mayor que L. Asignar a la variable J el valor determinado en el paso 4. Regresar al paso 2.

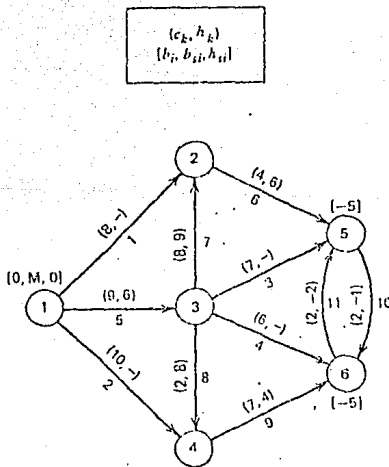
Este algoritmo y todas las subrutinas necesarias para su ejecución se describen con diagramas de flujo en el Apéndice B. Así mismo se proporciona ahí el algoritmo global NTCNCV que lee los datos para la red y llama a ENUMER

EJEMPLO NUMERICO

Para ilustrar el uso del algoritmo ENUMER, considere la red de la Figura 3.9. Antes de aplicar ENUMER se debe hacer lo siguiente:

1. Leer los datos correspondientes a nodos y arcos de la red y crear el modelo de red circular equivalente como el de la Figura 3.10.
2. Leer los datos correspondientes a los arcos con cargo fijo y los parámetros PER y BIG. El último representa el número "muy grande" que se ha mencionado antes.
3. Leer los flujos de una solución factible que satisfaga la conservación de flujo en cada nodo. Tal solución debe encontrarse previamente. En la Figura 3.10 se proporciona una solución factible para el ejemplo.
4. Aisgnar a ZB un valor muy grande (BIG).

Como puede observarse en la Figura 3.9, hay cuatro arcos que tienen cargo fijo. Estos se identifican en la red sustituyendo su parámetro de costo por un guión.



Arco con cargo fijo	h_f	h_v
1	25	4
2	15	6
3	20	4
4	25	3

PER = 1

BIG = 999

Figura 3.9

Red Ejemplo con Cuatro Arcos con Cargo Fijo

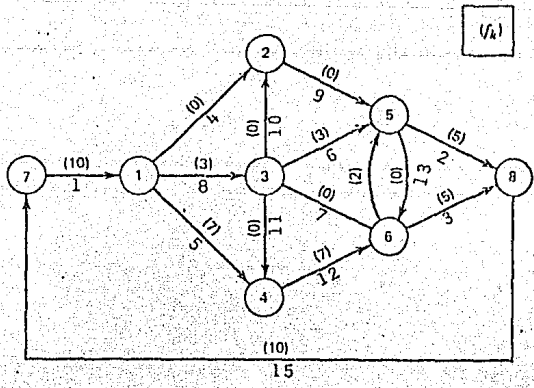


Figura 3.10

Red Circulatoria Equivalente con la Solución Factible Inicial

Después del llamado de ENUMER se inicializan en cero el indicador de nivel, P, W y U y se resuelve el problema de la Figura 3.10. En este problema todas las variables son libres y los costos en los arcos de la red con cargo fijo (4,5,6 y 7) con las aproximación lineales.

i	$k = k_c(i)$	$h'_k = h_F(i)/c_k + h_v(i)$
1	4	$7.125 = 25/8 + 4$
2	5	$7.500 = 15/10 + 6$
3	6	$6.857 = 20/7 + 4$
4	7	$7.166 = 95/6 + 3$

La solución a este problema relajado es

$$F_0 = [10, 5, 5, 0, 7, 3, 0, 3, 0, 0, 0, 7, 0, 2, 10] \quad , \quad Z_0 = Z_{LB}^0 = 115.05$$

y la solución redondeada resultante $XR = [0, 1, 1, 0]$, $ZR=131$ se convierte en la primera mejor solución actual, esto es, $XR=131=ZB$. Note que ZR es mayor que el objetivo relajado.

Seguidamente se ejecuta la fase de SEPARACION. En ésta, no sólo se decide cuál de las cuatro variables actualmente libres se fija, también se decide la dirección de la ramificación, es decir, el valor que asume la variable. El método usado en la separación se basa en la máxima diferencia entre los costos cóncavos y linealizados. En el vértice 0 las diferencias entre estos costos son

i	$k = k_c(i)$	$DIF_k = h_F(i) + h_v(i) * f_k - h_k' * f_k$
1	4	0. : pues $F(4) = 0$
2	5	4.5 = $15 + 6 * 7 - 7.500 * 7$
→ 3	6	11.43 = $20 + 4 * 3 - 6,857 * 3$
4	7	0 : pues $F(7) = 0$

Es claro que la separación se hace sobre x_3 pues la diferencia mayor se obtiene para el arco 6 que es el arco asociado a la variable 3 en la red. Como el flujo en el arco 6 [$F(6)=3$] es menor que la mitad de su capacidad [$c(6)=7$], se cierra este arco, es decir, x_3 se fija en cero. Esta decisión se muestra en la Figura 3.11, la cual ilustra el árbol de enumeración implícita. La secuencia y los vectores P, W, y U desarrollados por el algoritmo para cada vértice se resumen en el Cuadro 3.1.

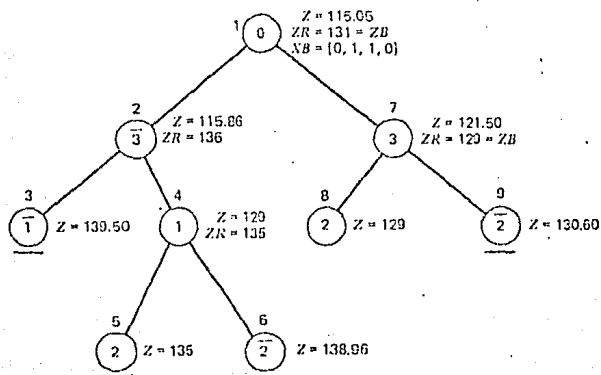


Figura 3.11

Árbol de Ramificación y Acotamiento para la red de la Figura 3.9

CUADRO 3.1. VECTORES DESARROLLADOS POR EL ALGORITMO DE ENUMERACION

Vértice	Secuencia	P	W	U
1	0	0 0 0 0	0 0 0 0	0 0 0 0
2	$\bar{3}$	3 0 0 0	0 0 -1 0	0 0 0 0
3	$\bar{3} \bar{1}$	3 1 0 0	-2 0 -1 0	0 0 0 0
4	$\bar{3} \underline{1}$	3 1 0 0	2 0 -1 0	0 1 0 0
5	$\bar{3} \underline{1} 2$	3 1 2 0	2 3 -1 0	0 1 0 0
6	$\bar{3} \underline{1} \underline{2}$	3 1 2 0	2 -3 -1 0	0 1 1 0
7	$\underline{3}$	3 0 0 0	0 0 1 0	1 0 0 0
8	$\underline{3} 2$	3 2 0 0	0 2 -1 0	1 0 0 0
9	$\underline{3} \underline{2}$	3 2 0 0	0 -2 1 0	1 1 0 0

La solución al problema relajado del vértice 2 es menor que la mejor solución actual ($Z_2 = 115.86 < ZB - ZB * PER / 100 = ZA = 129$), por lo tanto el proceso de enumeración implícita conduce al --vértice 3. Como $Z_3 = 139$ es mayor que ZA se sondea el vértice, -lo cual se indica con una línea debajo del vértice 3.

Puesto que en la secuencia correspondiente al vértice 3 no hay ningún elemento subrayado el procedimiento se dirige al vértice 4. El proceso de enumeración continúa y se evalúan explícitamente 9 vértices de los 31 posibles en el árbol de enumeración completa. Después de esto, se obtiene que la solución -óptima es $XB = [0, 1, 1, 0]$, $ZB = 129$ y corresponde al vértice 7. Los flujos óptimos se proporcionan en la Figura 3.12, donde los arcos cerrados 4 y 7 no se dibujan.

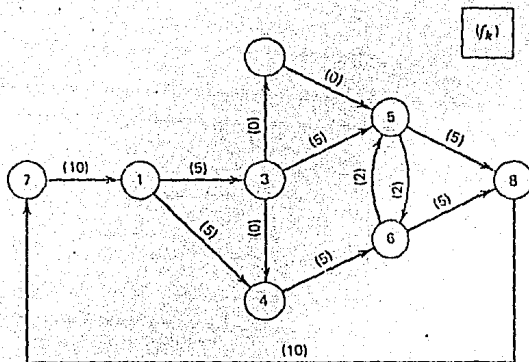


Figura 3.12

Solución Óptima para la Red de la Figura 3.9

3.7 ORGANIZACION Y FORMATO DE DATOS PARA EL PROGRAMA NTCNCV

Cuando el analista haya formulado su problema de optimización como un modelo de redes, es decir, ha enumerado los nodos consecutivamente y determinado los parámetros de nodo y arco, puede proceder a codificar los datos. En el Cuadro 3.2 se describe totalmente la organización y formato de los datos necesarios para hacer uso del programa NTCNV.

Cuadro 3.2 CODIFICACION DE DATOS PARA EL PROGRAMA NTCNCV

TARJETA TIPO	COLUMNA	FORMATO	VARIABLE	DESCRIPCION
1	1- 5	I5	N	Número de nodos en la red - sin considerar el nodo de holgura.
DATOS DE NODO				
2	1- 5	I5	I	Número asignado al nodo.
	6-15	F10.0	BF	Flujo externo fijo del nodo I.
	16-25	F10.0	BS	Cota superior del flujo de holgura en el nodo I.
	26-35	F10.0	CS	Costo unitario del flujo de holgura en el nodo I.
COMENTARIO : Se utilizan una tarjeta de este tipo por cada nodo de la red. Para terminar la lectura de los parámetros de nodo se agrega una tarjeta en blanco al final.				
3	1- 5	I5	I	Nodo origen del arco.
	6-10	I5	J	Nodo terminal del arco.
	11-20	F10.0	LOWER	Cota inferior del arco.
	21-30	F10.0	UPPER	Cota superior del arco.
	31-40	F10.0	COST	Costo unitario del arco o-

Cuadro 3.2 (pag. 2/4)

TARJETA TIPO	COLUMNA	FORMATO	VARIABLE	DESCRIPCION
				término constante (multi-- plicador) de la función -- convexa.
COMENTARIO : Se utiliza una tarjeta de este tipo por - cada arco de la red. Para indicar el final de la lectu ra de los datos de arcos se agrega una tarjeta en blan co al final.				
FLUJO FACTIBLE INICIAL				
4	1-10	F10.0	F(1)	Flujo inicial en el arco 1
	11-20	F10.0	F(2)	Flujo inicial en el arco 2
	----	----	----	-----
	71-80	F10.0	F(8)	Flujo inicial en el arco 8
COMENTARIO : Se utilizan tantas tarjetas de este tipo- como sean necesarias para especificar el flujo inicial- en cada arco de la red. La condición sobre el flujo -- factible inicial F es que satisfaga la conservación de flujo en cada nodo. F se debe determinar para la red- circulatoria equivalente. En la sección B.1.1 se des- cribe el simple procedimiento para convertir una red - en red circulatoria. Así mismo, se describe ahí la ma- nera como se enumeran sus arcos.				
PARAMETROS ASOCIADOS CON EL PROBLEMA CONCAVO				
5	1-10	F10.0	EPS	Número pequeño usado para- comparar con cero. Por -- ejemplo, 0.001.
5	11-20	F10.0	BIG	Número grande usado para - comparar con infinito (∞).
	21-25	F5.0	PER	Porcentaje de desviación - permitido con respecto al- valor objetivo óptimo.

Cuadro 3.2 (pag. 3/4)

TARJETA TIPO	COLUMNA	FORMATO	VARIABLE	DESCRIPCION
6	1- 5	I5	KC(I)	valor objetivo óptimo. Indice del arco en la red- circulatoria equivalente.
	6-15	F10.0	HF(I)	Cargo fijo del arco.
	16-25	F10.0	HV(I)	Costo lineal del arco.

COMENTARIO : Se utiliza una tarjeta de este tipo por cada arco de la red con cargo fijo. I es el índice o número que el analista le ha asociado al arco en la red original; el cual, se incrementa en el algoritmo desde uno hasta el número de arcos con cargo fijo presentes en la red, m_c . Para indicar el final de la lectura de estos parámetros se agrega una tarjeta en blanco al final.

OPCIONES DE REPORTE

7	1	I1	ISOLUC	Si ISOLUC ≠ 0 no se ejecuta ningún cálculo. Se imprime la representación de la red circulatoria y los flujos factibles iniciales. Esto permite al analista chequear los datos y confirmar que el flujo inicial corresponda con el arco y que el índice del arco con cargo fijo en la red original corresponda con su asociado en la red circulatoria. Si ISOLUC=0 también se imprimen los datos.
	2	I1	IPRINT	0 : se imprime la solución

Cuadro 3.2 (pag. 4/4)

TARJETA	COLUMNA	FORMATO	VARIABLE	DESCRIPCION
TIPO				

óptima.

1 : además se imprime la -
solución del problema-
relajado en cada vérti
ce y la solución redon
deada asociada.

NOTAS

1. El valor de las variables enteras (FORMATO I) debe ajustarse a la derecha en el campo asignado.
Ejemplo. Si la variable N tiene formato I5 y debe asumir el -
valor 15 se codifica 15
2. El valor de las variables reales (FORMATO F) se puede colocar en cualquier posición adentro del campo asignado. Si no se co
difica el punto, y puesto que todos los formatos incluidos -
tienen la forma FYY.0, el valor debe ajustarse a la derecha.
Ejemplos. Si el formato es F8.0 entonces 15.2 se puede codifi
car como 15.2 , 15.2 , ..., 15.2 ; 75 se puede co
dificar como 75. , 75. , ..., 75

3.8 APLICACIONES RESUELTAS CON EL PROGRAMA NTCNCV

En este apartado se presentan algunos ejemplos de redes de flujo con costo cóncavo y se resuelven haciendo uso del programa NTCNCV. Con esto se pretende fundamentalmente que el usuario, se familiarice con el uso del programa. Por otra parte, se hace un intento para introducir al analista en la formulación de problemas de optimización como problemas de redes. En cada aplicación se proporciona la descripción del problema, la red modelo para resolverlo, la red circulatoria creada por el programa, la codificación de los datos y los resultados de la corrida.

3.8.1 Aplicación 1

En la sección 3.6 se explicó la resolución de un ejemplo-número (ver Figura 3.9) y se dieron ahí la red circulatoria equivalente a la original, el árbol de ramificación y acotamiento-correspondiente y los vectores P, W y U asociados con cada vértice. El mismo ejemplo se resuelve aquí con el programa NTCNVC soliciéndose el reporte opcional dado por IPRINT = 1, precisamente para obtener los flujos óptimos de cada problema relajado y los vectores mencionados que lo identifican.

El cuadro 3.3 recoge la codificación de los datos y en la Figura 3.10 aparece la red circulatoria construida por el programa en la que no se especifican los parámetros de los tres arcos-adicionales, los cuales, se dan a continuación.

Arco	Indice	Cota inferior	Cota superior	Costo
(7,1)	1	0	M	0
(5,8)	2	5	5	0
(6,8)	3	5	5	0

La enumeración de los arcos artificiales es secuencial comenzando en 1 y se efectúa de acuerdo con el orden (es arbitrario) en que se codifiquen los datos de nodo. Por ejemplo, si los datos del nodo 5 se hubieran codificado primero que los del nodo 1, el arco (5,8) tendría el índice 1 en la red circulatoria. Similarmente se enumeran los arcos originales, pero obviamente comenzan

Cuadro 3.3

HOJA Nº 1/2

PROGRAMA NTCNCV

CODIFICACION DE DATOS PARA EL PROBLEMA DE LA SECCION 3.8.1

PROGRAMADOR Ing. Parra

5	6	7	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80
6																	
1	0.				1	0	0.	1									
2																	
3																	
4																	
5	-5.																
6	-15.																
7		2	0.					8									
8		4	0.					0.									
9		5	0.					7									
10		6	0.					6									
11		3	0.					9									
12		5	0.					4									
13		2	0.					8									
14		4	0.					2									
15		6	0.					7									
16		4	0.					1									
17		5	0.					2									
18																	
19	1	5.				5.		0.		7.		3.		0.		3.	
20	0.	0.			0.		7.		0.		2.		1	0.			
21	0.	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
22	14		12	5.				4.									

PROGRAMA NTCNCV

Cuadro 3.3

HOJA N° 2/2

PROGRAMADOR Ing Parra

0	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100
1	5																			
2	6		1	5																
3	7		1	2	5															
4	0																			
5																				
6																				
7																				
8																				
9																				
10																				
11																				
12																				
13																				
14																				
15																				
16																				
17																				
18																				
19																				
20																				
21																				
22																				
23																				
24																				
25																				
26																				
27																				
28																				
29																				
30																				
31																				
32																				
33																				
34																				
35																				
36																				
37																				
38																				
39																				
40																				
41																				
42																				
43																				
44																				
45																				
46																				
47																				
48																				
49																				
50																				
51																				
52																				
53																				
54																				
55																				
56																				
57																				
58																				
59																				
60																				
61																				
62																				
63																				
64																				
65																				
66																				
67																				
68																				
69																				
70																				
71																				
72																				
73																				
74																				
75																				
76																				
77																				
78																				
79																				
80																				
81																				
82																				
83																				
84																				
85																				
86																				
87																				
88																				
89																				
90																				
91																				
92																				
93																				
94																				
95																				
96																				
97																				
98																				
99																				
100																				

do con el índice que sigue al último arco artificial. El arco de retorno tiene el índice más alto en la red. Los resultados de la corrida se presentan seguidamente.

RED DE FLUJO CON COSTO CONCAVO

REPRESENTACION DE LA RED CIRCULATORIA

NUMERO DE NODOS = 8 NUMERO DE ARCOS = 15

LOS FLUJOS EXTERNOS Y EL POTENCIAL INICIAL DE CADA NODO ES 0.

ARCO K	N.ORIGEN O(K)	N.TERMINAL T(K)	COTA INF. CL(K)	COTA SUP. C(K)	COSTO H(K)	FLUJO FACTIBLE INICIAL F(K)
1	7	1	0	100	0.00	10
2	5	8	5	5	0.00	5
3	6	8	5	5	0.00	5
4	1	2	0	8	0.00	0
5	1	4	0	10	0.00	7
6	3	5	0	7	0.00	3
7	3	6	0	6	0.00	0
8	1	3	0	9	6.00	3
9	2	5	0	4	6.00	0
10	3	2	0	8	9.00	0
11	3	4	0	2	8.00	0
12	4	6	0	7	4.00	7
13	5	6	0	2	-1.00	0
14	6	5	0	2	-2.00	2
15	8	7	0	175	0.00	10

OPCIONES DE REPORTES : ISOLUC = 0 IPRINT = 1

DATOS ASOCIADOS CON EL PROBLEMA CONCAVO

INDICE DEL ARCO EN LA RED ORIGINAL I	INDICE DEL ARCO EN LA RED CIRCULATORIA K = KC(I)	CARGO FIJO HF(I)	COSTO LINEAL HV(I)	APROXIMACION LINEAL PARA EL PROBLEMA RELAJADO H(K)
1	4	25.00	4.00	7.125
2	5	15.00	6.00	7.500
3	6	20.00	4.00	6.857
4	7	25.00	3.00	7.167

VALOR PARA COMPARAR CON CERO 0.00010

VALOR PARA COMPARAR CON INFINITO 99999.

PORCENTAJE DE DESVIACION CON RESPECTO AL VALOR OPTIMO 1.0

PROBLEMA RELAJADO EN EL VERTICE 1

P
0 0 0 0

W
0 0 0 0

U
0 0 0 0

X
0.000 0.700 0.429 0.000

NODO I	POTENCIAL PI(I)	APUNTAADOR HACIA ATRAS PB(I)
1	1	-8
2	8	4
3	7	8
4	8	5
5	13	0
6	12	0
7	1	-1
8	1	-15

ARCO K	N.ORIGEN O(K)	N.TERMINAL T(K)	COTA INF. CL(K)	COTA SUP. C(K)	COSTO H(K)	FLUJO F(K)	COSTO DEL FLUJO
1	7	1	0	100	0.00	10	0.00
2	5	8	5	5	0.00	5	0.00
3	6	8	5	5	0.00	5	0.00
4	1	2	0	8	7.13	0	0.00
5	1	4	0	10	7.50	7	52.50
6	3	5	0	7	8.86	3	20.57
7	3	6	0	6	7.17	0	0.00
8	1	3	0	7	6.00	3	18.00
9	2	5	0	4	6.00	0	0.00
10	3	2	0	8	9.00	0	0.00
11	3	4	0	2	8.00	0	0.00
12	4	6	0	7	4.00	7	28.00
13	5	6	0	2	-1.00	0	0.00
14	6	5	0	2	-2.00	2	-1.00
15	8	7	0	175	0.00	10	0.00

COSTO TOTAL DEL FLUJO

115.07

MEJOR SOLUCION ACTUAL = ZB = 99999.00

ZA = ZB - PER/100 * /ZB/ = 98999.01

SOLUCION REDONDEADA

XR
0 1 1 0

COSTO TOTAL DEL FLUJO = ZR = 131.00

PROBLEMA RELAJADO EN EL VERTICE 2

P 3 0 0 0

W 0 0 -1 0

U 0 0 0 0

X 0.375 0.700 0.000 0.000

NODO I	POTENCIAL PI(I)	APUNTADOR HACIA ATRAS PR(I)
1	1	-8
2	8	4
3	7	8
4	8	5
5	14	9
6	13	13
7	1	-1
8	1	-15

ARCO K	N.ORIGEN O(K)	N.TERMINAL T(K)	COTA INF. CL(K)	COTA SUP. C(K)	COSTO H(K)	FLUJO F(K)	COSTO DEL FLUJO
1	7	1	0	100	0.00	10	0.00
2	5	8	5	5	0.00	5	0.00
3	6	8	5	5	0.00	5	0.00
4	1	2	0	8	7.13	3	21.38
5	1	4	0	10	7.50	7	52.50
6	3	5	0	7	99997.00	0	0.00
7	3	6	0	6	7.17	0	0.00
8	1	3	0	9	6.00	0	0.00
9	2	5	0	4	6.00	3	18.00
10	3	2	0	8	9.00	0	0.00
11	3	4	0	2	8.00	0	0.00
12	4	6	0	7	4.00	7	28.00
13	5	6	0	2	-1.00	0	0.00
14	6	5	0	2	-2.00	2	-4.00
15	8	7	0	175	0.00	10	0.00

COSTO TOTAL DEL FLUJO 115.88

MEJOR SOLUCION ACTUAL = ZB = 131.00

ZA = ZB - PER/100 * ZB/ = 129.69

SOLUCION REDONDEADA

XR 1 1 0 0

COSTO TOTAL DEL FLUJO = ZR = 136.00

PROBLEMA RELAJADO EN EL VERTICE 3

P
3 1 0 0
W
-2 0 -1 0
U
0 0 0 0
X
0.000 0.700 0.000 0.000

NODO I	POTENCIAL PI(I)	APUNTA DOR HACIA ATRAS PB(I)
1	1	-8
2	16	10
3	7	8
4	8	5
5	22	9
6	14	7
7	1	-1
8	1	-15

ARCO K	N.ORIGEN D(K)	N.TERMINAL T(K)	COTA INF. CL(K)	COTA SUP. C(K)	COSTO H(K)	FLUJO F(K)	COSTO DEL FLUJO
1	7	1	0	100	0.00	10	0.00
2	5	0	5	5	0.00	5	0.00
3	6	8	5	5	0.00	5	0.00
4	1	2	0	8	99999.00	0	0.00
5	1	4	0	10	7.50	7	52.50
6	3	5	0	7	99999.00	0	0.00
7	3	6	0	6	7.17	0	0.00
8	1	3	0	9	6.00	3	18.00
9	2	5	0	4	6.00	3	18.00
10	3	2	0	8	9.00	3	27.00
11	3	4	0	2	8.00	0	0.00
12	4	6	0	7	4.00	7	28.00
13	5	6	0	2	-1.00	0	0.00
14	6	5	0	2	-2.00	2	-4.00
15	0	7	0	175	0.00	10	0.00

COSTO TOTAL DEL FLUJO 139.50

MEJOR SOLUCION ACTUAL = ZB = 131.00

ZA = ZB - PER/100 * /ZB/ = 129.69

PROBLEMA RELAJADO EN EL VERTICE 4

P 3 1 0 0
W 2 0 -1 0
U 0 1 0 0
X 1.000 0.600 0.000 0.000

NODO I	POTENCIAL PI(I)	APUNTAADOR HACIA ATRAS PB(I)
1	12	-5
2	16	10
3	18	8
4	19	-12
5	24	9
6	23	13
7	12	-1
8	12	-15

ARCO K	N.ORIGEN O(K)	N.TERMINAL T(K)	COTA INF. CL(K)	COTA SUP. C(K)	COSTO H(K)	FLUJO F(K)	COSTO DEL FLUJO
1	7	1	0	100	0.00	10	0.00
2	5	8	5	5	0.00	5	0.00
3	6	8	5	5	0.00	5	0.00
4	1	2	0	8	4.00	4	16.00
5	1	4	0	10	7.50	6	45.00
6	3	5	0	7	9999.00	0	0.00
7	3	6	0	6	7.17	0	0.00
8	1	3	0	9	6.00	0	0.00
9	2	5	0	4	6.00	4	24.00
10	3	2	0	8	9.00	0	0.00
11	3	4	0	2	8.00	0	0.00
12	4	6	0	7	4.00	6	24.00
13	5	6	0	2	-1.00	1	-1.00
14	6	5	0	2	-2.00	2	-4.00
15	8	7	0	175	0.00	10	0.00

COSTO TOTAL DEL FLUJO 129.00

MEJOR SOLUCION ACTUAL = ZB = 131.00

ZA = ZB - PER/100 * ZB/ = 129.69

SOLUCION REDONDEADA

XR 1 1 0 0

COSTO TOTAL DEL FLUJO = ZR = 135.00

PROBLEMA RELAJADO EN EL VERTICE 5

P 3 1 2 0

W 2 3 -1 0

U 0 1 0 0

X 1.000 1.000 0.000 0.000

NODO I	POTENCIAL PI(I)	AFUNTADOR HACIA ATRAS PB(I)
1	13	-5
2	17	10
3	19	8
4	19	-12
5	24	-13
6	23	12
7	13	-1
8	13	-15

ARCO K	N.ORIGEN O(K)	N.TERMINAL T(K)	COTA INF. CL(K)	COTA SUP. C(K)	COSTO H(K)	FLUJO F(K)	COSTO DEL FLUJO
1	7	1	0	100	0.00	10	0.00
2	5	8	5	5	0.00	5	0.00
3	6	8	5	5	0.00	5	0.00
4	1	2	0	8	4.00	4	16.00
5	1	4	0	10	6.00	6	36.00
6	3	5	0	7	9999.00	0	0.00
7	3	6	0	6	7.17	0	0.00
8	1	3	0	9	6.00	0	0.00
9	2	5	0	4	6.00	4	24.00
10	3	2	0	8	7.00	0	0.00
11	3	4	0	2	8.00	0	0.00
12	4	6	0	7	4.00	6	24.00
13	5	6	0	2	-1.00	1	-1.00
14	6	5	0	2	-2.00	2	-4.00
15	8	7	0	175	0.00	10	0.00

COSTO TOTAL DEL FLUJO

135.00

MEJOR SOLUCION ACTUAL = ZB = 131.00

ZA = ZB - PER/100 * /ZB/ = 129.69

PROBLEMA RELAJADO EN EL VERTICE 6

P
3 1 2 0

W
2 -3 -1 0

U
0 1 1 0

X
1.000 0.000 0.000 1.000

NODO I	POTENCIAL PI(I)	APUNTADOR HACIA ATRAS PB(I)
1	13	-5
2	17	4
3	19	8
4	22	-12
5	27	-13
6	26	7
7	13	-1
8	13	-15

ARCO K	N.ORIGEN O(K)	N.TERMINAL T(K)	COTA INF. CL(K)	COTA SUP. C(K)	COSTO H(K)	FLUJO F(K)	COSTO DEL FLUJO
1	7	1	0	100	0.00	10	0.00
2	5	8	5	5	0.00	5	0.00
3	6	8	5	5	0.00	5	0.00
4	1	2	0	8	4.00	4	16.00
5	1	4	0	10	99999.00	0	0.00
6	3	5	0	7	99999.00	0	0.00
7	3	4	0	6	7.17	6	43.00
8	1	3	0	9	6.00	6	36.00
9	2	5	0	4	6.00	4	24.00
10	3	2	0	8	9.00	0	0.00
11	3	4	0	2	8.00	0	0.00
12	4	6	0	7	4.00	0	0.00
13	5	6	0	2	-1.00	1	-1.00
14	6	5	0	2	-2.00	2	-4.00
15	8	7	0	175	0.00	10	0.00

COSTO TOTAL DEL FLUJO 139.00

MEJOR SOLUCION ACTUAL = ZD = 131.00

ZA = ZB - PER/100 * /ZB/ = 129.69

PROBLEMA RELAJADO EN EL VERTICE 7

P
 3 0 0 0
 W
 0 0 1 0
 U
 1 0 0 0
 X
 0.000 0.500 1.000 0.000

NODO I	POTENCIAL PI(I)	APUNTAADOR HACIA ATRAS PB(I)
1	17	-5
2	24	-9
3	23	8
4	25	-12
5	27	-13
6	29	13
7	17	-1
8	17	-15

ARCO K	N.ORIGEN O(K)	N.TERMINAL T(K)	COTA INF. CL(K)	COTA SUP. C(K)	COSTO H(K)	FLUJO F(K)	COSTO DEL FLUJO
1	7	1	0	100	0.00	10	0.00
2	5	8	5	5	0.00	5	0.00
3	6	8	5	5	0.00	5	0.00
4	1	2	0	8	7.13	0	0.00
5	1	4	0	10	7.50	5	37.50
6	3	5	0	7	4.00	5	20.00
7	3	8	0	5	7.17	0	0.00
8	1	8	0	9	6.00	5	30.00
9	2	5	0	4	6.00	0	0.00
10	3	2	0	8	9.00	0	0.00
11	3	4	0	2	8.00	0	0.00
12	4	6	0	7	4.00	5	20.00
13	5	6	0	2	-1.00	2	-2.00
14	6	5	0	2	-2.00	2	-4.00
15	8	7	0	175	0.00	10	0.00

COSTO TOTAL DEL FLUJO 121.50

MEJOR SOLUCION ACTUAL = ZB = 131.00

ZA = ZB - PER/100 * /ZD/ = 129.69

SOLUCION REDONDEADA

XR
 0 1 1 0

COSTO TOTAL DEL FLUJO = ZR = 129.00

PROBLEMA RELAJADO EN EL VERTICE 0

P
3 2 0 0

W
0 2 1 0

U
1 0 0 0

X
0.000 1.000 1.000 0.000

NODO I	POTENCIAL PI(I)	APUNTAADOR HACIA ATRAS PB(I)
1	19	-5
2	26	-9
3	25	8
4	25	-12
5	29	-13
6	29	12
7	19	-1
8	19	-15

ARCO K	N.ORIGEN O(K)	N.TERMINAL T(K)	COTA INF. CL(K)	COTA SUP. C(K)	COSTO H(K)	FLUJO F(K)	COSTO DEL FLUJO
1	7	1	0	100	0.00	10	0.00
2	5	8	5	5	0.00	5	0.00
3	6	8	5	5	0.00	5	0.00
4	1	2	0	0	7.13	0	0.00
5	1	4	0	10	6.00	5	30.00
6	3	5	0	7	4.00	0	20.00
7	3	6	0	8	7.17	5	0.00
8	1	3	0	4	6.00	0	30.00
9	2	5	0	8	6.00	0	0.00
10	3	2	0	8	9.00	0	0.00
11	3	4	0	2	0.00	0	0.00
12	4	6	0	7	4.00	5	20.00
13	5	6	0	2	-1.00	2	-2.00
14	6	5	0	2	-2.00	2	-4.00
15	8	7	0	175	0.00	10	0.00

COSTO TOTAL DEL FLUJO 129.00

MEJOR SOLUCION ACTUAL = ZB = 129.00

ZA = ZB - PER/100 * /ZB/ = 127.71

PROBLEMA RELAJADO EN EL VERTICE 9

P
3 2 0 0

M
0 -2 1 0

U
1 1 0 0

X
0.125 0.000 1.000 0.500

NODO I	POTENCIAL PI(I)	APUNTADOR HACIA ATRAS PB(I)
1	19	-5
2	26	4
3	28	-6
4	31	-12
5	32	9
6	35	7
7	19	-1
8	19	-15

ARCO K	N.ORIGEN O(K)	N.TERMINAL T(K)	COTA INF. CL(K)	COTA SUP. C(K)	COSTO H(K)	FLUJO F(K)	COSTO DEL FLUJO
1	7	1	0	100	0.00	10	0.00
2	5	8	5	5	0.00	5	0.00
3	6	8	5	5	0.00	5	0.00
4	1	2	0	8	7.13	1	7.13
5	1	4	0	10	99999.00	0	0.00
6	3	5	0	7	4.00	6	24.00
7	3	6	0	6	7.17	3	21.50
8	1	3	0	9	4.00	9	54.00
9	2	5	0	4	4.00	1	6.00
10	3	2	0	8	9.00	0	0.00
11	3	4	0	2	8.00	0	0.00
12	4	6	0	7	4.00	0	0.00
13	5	6	0	2	-1.00	2	-2.00
14	6	5	0	2	-2.00	0	0.00
15	8	7	0	175	0.00	10	0.00

COSTO TOTAL DEL FLUJO 130.63

MEJOR SOLUCION ACTUAL = ZB = 129.00

ZA = ZB - PER/100 * /ZB/ = 127.71

SOLUCION OPTIMAXD
0 1 1 0

NODO I	POTENCIAL PI(I)	APUNTAOR HACIA ATRAS PB(I)
1	25	-8
2	32	-9
3	31	-6
4	31	5
5	35	14
6	35	12
7	25	-1
8	25	-15

ARCO K	N.ORIGEN O(K)	N.TERMINAL T(K)	COTA INF. CL(K)	COTA SUP. C(K)	COSTO H(K)	FLUJO F(K)	COSTO DEL FLUJO
1	7	1	0	100	0.00	10	0.00
2	5	8	5	5	0.00	5	0.00
3	6	8	5	5	0.00	5	0.00
4	1	2	0	8	?????.00	0	0.00
5	1	4	0	10	6.00	5	30.00
6	3	5	0	7	4.00	5	20.00
7	3	6	0	6	?????.00	0	0.00
8	1	3	0	9	6.00	5	30.00
9	2	5	0	4	6.00	0	0.00
10	3	2	0	8	9.00	0	0.00
11	3	4	0	2	8.00	0	0.00
12	4	6	0	7	4.00	5	20.00
13	5	6	0	2	-1.00	2	-2.00
14	6	5	0	2	-2.00	2	-4.00
15	8	7	0	175	0.00	10	0.00

COSTO TOTAL DEL FLUJO 129.00

NUMERO DE PROBLEMAS RELAJADOS RESUELTOS = 9.

PT=4.2 IO=1.2

3.8.2 Aplicación 2

Una compañía produce sillas y dispone de cuatro plantas. El costo de producción por silla y los niveles mínimos y máximos de producción mensual en cada planta se dan a continuación.

Planta	Cargo fijo	Costo (\$/silla)	Prod. Mínima	Prod. Máxima
1	5000	250	0	700
2	0	350	400	750
3	0	150	500	1200
4	0	200	250	700

El cargo fijo representa el costo de arrancar o establecer la producción en la planta.

Cada silla requiere de 10 kilos de madera y existen dos proveedores, A y B, que pueden surtir cualquier cantidad de madera. A cambio, la compañía garantiza comprar al menos 8000 kilos de madera por mes a cada proveedor. El costo de la madera es de \$11/Kg para el proveedor A y \$8.5/Kg para el B. Los costos de envío de madera de cada proveedor a las plantas (en \$/Kg) se resumen a continuación.

Proveedor	1	2	3	4
A	11	22	44	33
B	22	33	22	33

Además del costo de envío por silla, el proveedor A demanda una cantidad fija de \$1500 para poder surtir madera a la planta 1.

Las sillas se venden en cuatro ciudades (C1, C2, C3 y C4) y los costos de transporte (h_{ij}) en pesos por silla entre las plantas y las ciudades son:

Planta	Ciudad 1		Ciudad 2		Ciudad 3		Ciudad 4	
	H_F	h_v	H_F	h_v	H_F	h_v	H_F	h_v
1	10000	25		50		100	10000	30
2		150		300		350		150
3		150	6000	50		250		150
4		400		100	4000	50		200

H_F es el costo fijo mensual que implicaría establecer el medio de transporte entre planta y ciudad.

Finalmente, el siguiente cuadro muestra la demanda mínima y máxima mensual de sillas y el precio de venta en cada ciudad.

Ciudad	Precio de Venta	Demanda mínima	Demanda máxima
1	1000	1000	1500
2	750	300	400
3	1000	800	1300
4	900	700	1200

Se desea establecer el plan de compra de madera, así como la producción y distribución de sillas en cada planta de manera que se minimice el costo total.

La red modelo para este problema se muestra en la Figura 3.13a. Puesto que sólo existe un tipo de flujo, todos los flujos se pueden expresar en kilos de madera o en sillas. Arbitrariamente se ha decidido expresar el flujo en términos de sillas, entonces, también los costos relativos a la madera y limitaciones en los flujos se expresan en términos de sillas equivalentes en la Figura 2.13a. R denota un número muy grande y se usa como cota superior en los arcos que no tienen limitación en el flujo.

Claramente, el flujo en el arco (5,9) representará la producción en la planta 3, y la cantidad de sillas enviadas de la planta 4 a la Ciudad 3 estará representada por el flujo en el arco (10,13). Los nodos 1 y 2 representan a los proveedores A y

Arco con cargo $f j_0$	h_F	h_V
(1,3)	1500.	11.
(3,7)	5000.	250.
(7,11)	10000.	25.
(7,14)	10000.	30.
(9,12)	6000.	50.
(10,13)	4000.	50.

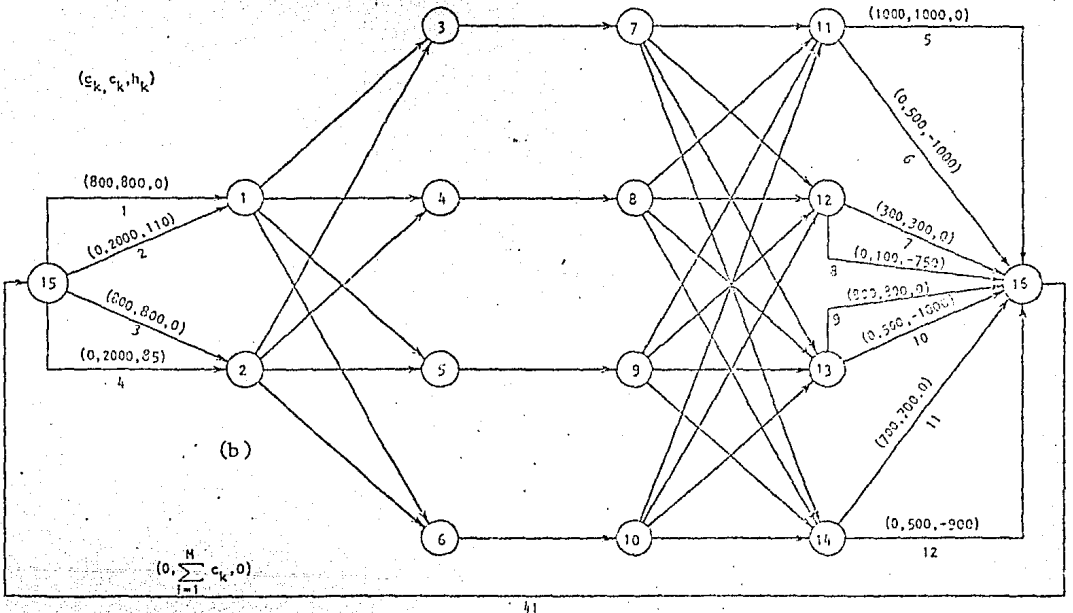
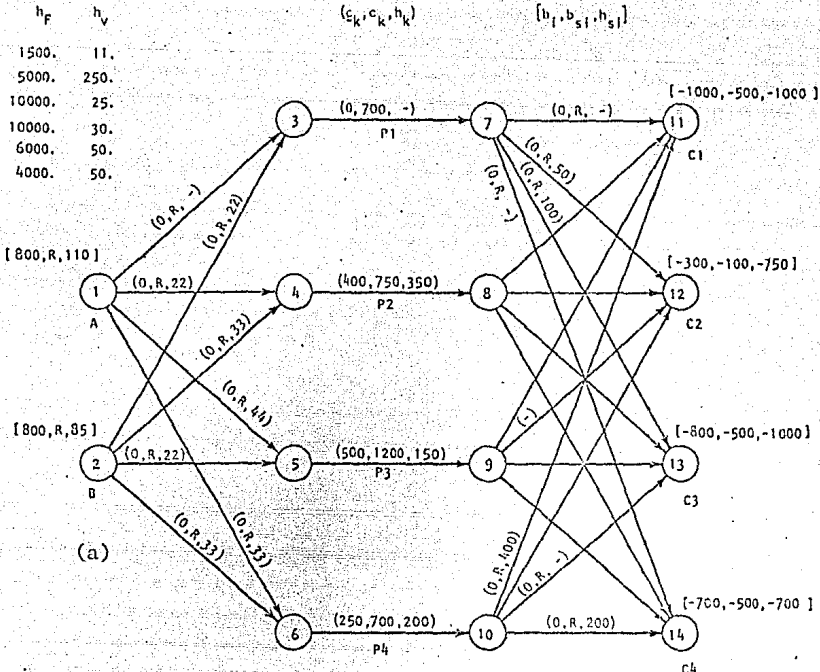


Figura 3.13
 (a) Red Modelo que Representa al Problema (b) Red Circulatoria Equivalente Creada por el Programa

B, respectivamente. Así, los arcos que unen los nodos 1 o 2 con los nodos 3,4,5 y 6 representan el transporte de los proveedores hacia las plantas. Similarmente, los arcos (3,7), (4,8), (5,9) y (6,10) representan las cuatro plantas y los arcos que unen los nodos 7 a 10 con los nodos 11 a 14 representan al transporte de las plantas hacia las ciudades.

Note como se ha usado el parámetro de flujo externo fijo para incorporar en el modelo la restricción de compra mínima en los nodos 1 y 2 y demanda mínima en los nodos 11,12,13 y 14. El máximo flujo de holgura en estos últimos nodos es la diferencia entre la demanda máxima y mínima de la ciudad. El signo negativo está presente en los costos de los flujos de holgura para transformar un beneficio en un costo.

En la Figura 3.13b se muestra la red circulatoria equivalente que construye el programa NTCNCV. Note que como hay 12 flujos externos de nodo el programa genera 12 arcos artificiales y los enumera en orden secuencial; por tanto, el arco de la red original cuyos datos se codifiquen primero tendrá el índice 13, en la red circulatoria, el arco cuyos datos se codifiquen seguidamente tendrá el índice 14, etc. La enumeración de los arcos de la Figura 3.14 corresponde a la codificación de datos del Cuadro 3.3. Observe en este cuadro que la corrida se hará para un porcentaje de desviación con respecto al óptimo (PER) de 1 y que se solicita el reporte IPRINT=0.

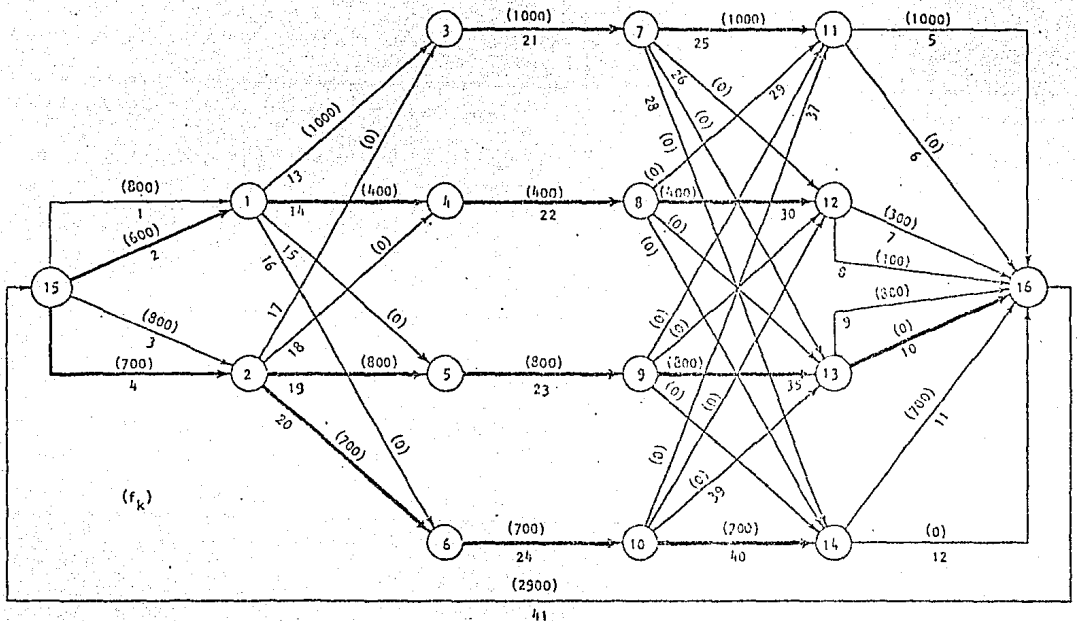


Figura 3.14
Solución Factible Inicial

Procedimiento para Encontrar una Solución Factible Inicial.

Cuando no es necesario satisfacer las restricciones de acotamiento en los flujos, se puede proceder como sigue para obtener un flujo que satisfaga la conservación del mismo en cada nodo.

1. Defina un árbol de expansión básico arbitrario
2. Asigne, a cada arco no básico, un flujo igual a su cota inferior o superior
3. Asigne un flujo a los arcos básicos, procediendo de las terminales del árbol hacia la raíz, para satisfacer la conservación de flujo en los nodos.

La solución factible de la Figura 3.14 se obtuvo siguiendo los puntos anteriores. El árbol de expansión seleccionado se ilustra con trazo grueso.

5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80
3	7	0		7		0									
4	8	1		7		3									
5	9	5		1		1									
6	1	2		7		2									
7	1	1		2											
7	1	2		2											
7	1	3		2		1									
7	1	4		2											
8	1	1		2		1									
8	1	2		2		3									
8	1	3		2		3									
8	1	4		2		1									
9	1	1		2		1									
9	1	2		2											
9	1	3		2		2									
9	1	4		2		1									
10	1	1		2		4									
10	1	2		2		1									
10	1	3		2											
10	1	4		2		2									
10	1	5		3											
10	1	6		3		7									
10	1	7		7											
10	1	8		8											
10	1	9		8											
10	1	10		8		1									
10	1	11		8											
10	1	12		8											
10	1	13		8											
10	1	14		8											
10	1	15		8											
10	1	16		8											
10	1	17		8											
10	1	18		8											
10	1	19		8											
10	1	20		8											
10	1	21		8											
10	1	22		8											
10	1	23		8											
10	1	24		8											
10	1	25		8											
10	1	26		8											
10	1	27		8											
10	1	28		8											
10	1	29		8											
10	1	30		8											
10	1	31		8											
10	1	32		8											
10	1	33		8											
10	1	34		8											
10	1	35		8											
10	1	36		8											
10	1	37		8											
10	1	38		8											
10	1	39		8											
10	1	40		8											
10	1	41		8											
10	1	42		8											
10	1	43		8											
10	1	44		8											
10	1	45		8											
10	1	46		8											
10	1	47		8											
10	1	48		8											
10	1	49		8											
10	1	50		8											
10	1	51		8											
10	1	52		8											
10	1	53		8											
10	1	54		8											
10	1	55		8											
10	1	56		8											
10	1	57		8											
10	1	58		8											
10	1	59		8											
10	1	60		8											
10	1	61		8											
10	1	62		8											
10	1	63		8											
10	1	64		8											
10	1	65		8											
10	1	66		8											
10	1	67		8											
10	1	68		8											
10	1	69		8											
10	1	70		8											
10	1	71		8											
10	1	72		8											
10	1	73		8											
10	1	74		8											
10	1	75		8											
10	1	76		8											
10	1	77		8											
10	1	78		8											
10	1	79		8											
10	1	80		8											
10	1	81		8											
10	1	82		8											
10	1	83		8											
10	1	84		8											
10	1	85		8											
10	1	86		8											
10	1	87		8											
10	1	88		8											
10	1	89		8											
10	1	90		8											
10	1	91		8											
10	1	92		8											
10	1	93		8											
10	1	94		8											
10	1	95		8											
10	1	96		8											
10	1	97		8											
10	1	98		8											
10	1	99		8											
10	1	100		8											

RED DE FLUJO CON COSTO CONCAVO

REPRESENTACION DE LA RED CIRCULATORIA

NUMERO DE NODOS = 16 NUMERO DE ARCOS = 41

LOS FLUJOS EXTERNOS Y EL POTENCIAL INICIAL DE CADA NODO ES 0.

ARCO K	N. ORIGEN O(K)	N. TERMINAL T(K)	COTA INF. CL(K)	COTA SUP. C(K)	COSTO H(K)	FLUJO FACTIBLE INICIAL F(K)
1	15	1	800	800	0.00	800
2	15	1	0	2000	110.00	100
3	15	2	800	800	0.00	800
4	15	2	0	2000	85.00	200
5	11	16	1000	1000	0.00	500
6	11	16	0	500	-1000.00	0
7	12	16	300	300	0.00	100
8	12	16	0	100	-750.00	300
9	13	16	800	800	0.00	500
10	13	16	0	500	-1000.00	0
11	14	16	700	700	0.00	500
12	14	16	0	500	-900.00	0
13	1	3	0	2000	0.00	500
14	1	4	0	2000	22.00	400
15	1	5	0	2000	44.00	0
16	1	6	0	2000	33.00	0
17	2	3	0	2000	22.00	0
18	2	4	0	2000	33.00	0
19	2	5	0	2000	22.00	500
20	2	6	0	2000	33.00	500
21	3	7	0	700	0.00	500
22	4	8	400	750	350.00	400
23	5	9	500	1200	150.00	500
24	6	10	250	700	200.00	500
25	7	11	0	2000	0.00	500
26	7	12	0	2000	50.00	0
27	7	13	0	2000	100.00	0
28	7	14	0	2000	0.00	0
29	8	11	0	2000	150.00	0
30	8	12	0	2000	300.00	400
31	8	13	0	2000	350.00	0
32	8	14	0	2000	150.00	0
33	9	11	0	2000	150.00	0
34	9	12	0	2000	0.00	0
35	9	13	0	2000	250.00	500
36	9	14	0	2000	150.00	0
37	10	11	0	2000	400.00	0
38	10	12	0	2000	100.00	0
39	10	13	0	2000	0.00	0
40	10	14	0	2000	200.00	500
41	16	15	0	61350	0.00	1700

OPCIONES DE REPORTES : ISOLUC = 0 IPRINT = 0

DATOS ASOCIADOS CON EL PROBLEMA CONCAVO

INDICE DEL ARCO EN LA RED ORIGINAL I	INDICE DEL ARCO EN LA RED CIRCULATORIA K = KC(I)	CARGO FIJO HF(I)	COSTO LINEAL HV(I)	APROXIMACION LINEAL PARA EL PROBLEMA RELAJADO H(K)
1	13	1500.00	11.00	11.750
2	21	5000.00	250.00	257.143
3	25	10000.00	25.00	30.000
4	28	10000.00	30.00	35.000
5	34	6000.00	50.00	53.000
6	39	4000.00	50.00	52.000

VALOR PARA COMPARAR CON CERO 0.00100

VALOR PARA COMPARAR CON INFINITO 9999999.

PORCENTAJE DE DESVIACION CON RESPECTO AL VALOR OPTIMO 1.0

SOLUCION OPTIMA

XD
1 1 1 0 1 1

NODO I	POTENCIAL PI(I)	AFUNTADOR HACIA ATRAS PB(I)
1	424	-13
2	413	4
3	435	17
4	446	18
5	435	19
6	446	20
7	1228	-27
8	1103	-29
9	1103	-33
10	1278	-39
11	1253	25
12	1153	34
13	1328	27
14	1253	32
15	328	41
16	320	10

ARCO K	N. ORIGEN O(K)	N. TERMINAL T(K)	COTA INF. CL(K)	COTA SUP. C(K)	COSTO H(K)	FLUJO F(K)	COSTO DEL FLUJO
1	15	1	800	800	0.00	800	0.00
2	15	1	0	2000	110.00	0	0.00
3	15	2	800	800	0.00	800	0.00
4	15	2	0	2000	85.00	1750	148750.00
5	11	16	1000	1000	0.00	1000	0.00
6	11	16	0	500	-1000.00	500	-500000.00
7	12	16	300	300	0.00	300	0.00
8	12	16	0	100	-750.00	0	0.00
9	13	16	800	800	0.00	800	0.00
10	13	16	0	500	-1000.00	50	-50000.00
11	14	16	700	700	0.00	700	0.00
12	14	16	0	500	-900.00	0	0.00
13	1	3	0	2000	11.00	400	4400.00
14	1	4	0	2000	22.00	400	8800.00
15	1	5	0	2000	44.00	0	0.00
16	1	6	0	2000	33.00	0	0.00
17	2	3	0	2000	22.00	300	6600.00
18	2	4	0	2000	33.00	350	11550.00
19	2	5	0	2000	22.00	1200	26400.00
20	2	6	0	2000	33.00	700	23100.00
21	3	7	0	700	250.00	700	17500.00
22	4	8	400	750	350.00	750	262500.00
23	5	9	500	1200	150.00	1200	180000.00
24	6	10	250	700	200.00	700	140000.00
25	7	11	0	2000	25.00	550	13750.00
26	7	12	0	2000	50.00	0	0.00
27	7	13	0	2000	100.00	150	15000.00
28	7	14	0	2000	999999.00	0	0.00
29	8	11	0	2000	150.00	100	15000.00
30	8	12	0	2000	300.00	0	0.00
31	8	13	0	2000	350.00	0	0.00
32	8	14	0	2000	150.00	650	97500.00
33	9	11	0	2000	150.00	850	127500.00
34	9	12	0	2000	50.00	300	15000.00
35	9	13	0	2000	250.00	0	0.00
36	9	14	0	2000	150.00	50	7500.00
37	10	11	0	2000	400.00	0	0.00
38	10	12	0	2000	100.00	0	0.00
39	10	13	0	2000	50.00	700	35000.00
40	10	14	0	2000	200.00	0	0.00
41	16	15	0	61350	0.00	3350	0.00
COSTO TOTAL DEL FLUJO							269850.00

NUMERO DE PROBLEMAS RELAJADOS RESUELTOS = 7

PI-2.5 IO-1.2

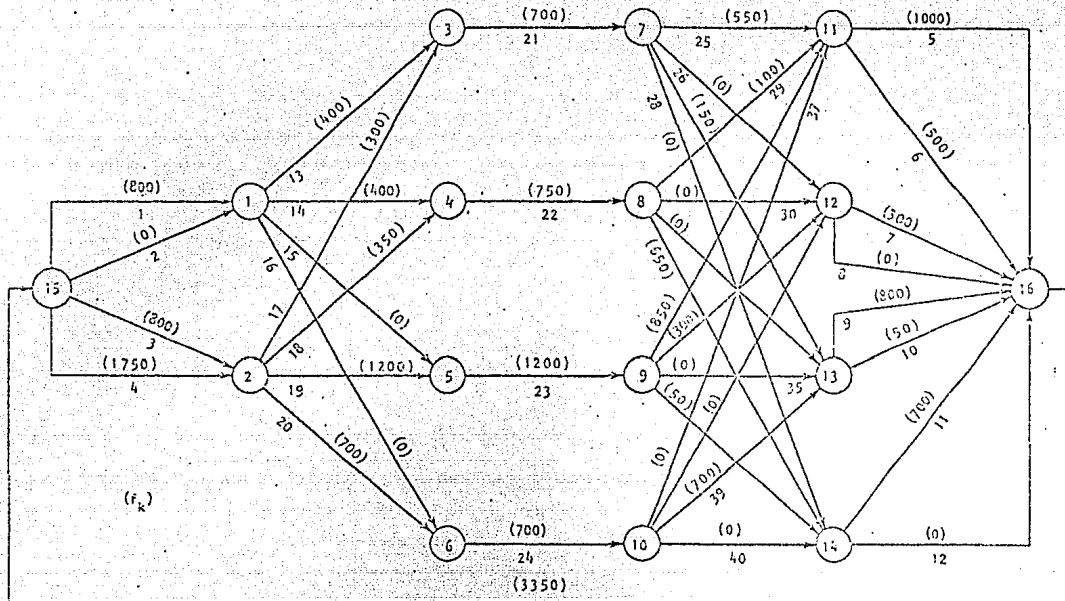


Figura 3.15
Red con el Flujo Optimo

APENDICE A

ALGORITMO DE RESOLUCION PARA REDES DE FLUJO CON COSTOS CONVEXOS

A.1 ALGORITMOS PARA MANIPULAR REDES

A.1.1. Representación De Redes

La representación de una red es el conjunto de constantes, listas, y matrices usadas para representar la estructura de la red y sus parámetros en un algoritmo para computadora. La representación que se use es particularmente importante con relación al costo del tiempo y al costo del espacio (memoria en el computador). El costo del tiempo es el tiempo consumido por un algoritmo en el computador digital para lograr su cometido. Este tiempo se puede describir en término del tiempo de cálculo (tiempo procesador) o del número de operaciones necesitadas para ejecutar todas las partes del procedimiento. El costo del espacio, medido en "palabras" de computador, se asocia con la cantidad de espacio (se supone de acceso directo) requerido para representar la información necesaria para la ejecución del algoritmo. Tiempo y espacio generalmente son variables complementarias en el diseño de procedimientos para computador. Usualmente una disminución en uno es acompañada por un incremento en el otro. La cantidad, tiempo o espacio, más importante al diseñar un algoritmo depende del total de re-

cursos disponibles. Los diseñadores de los algoritmos que se describen aquí, han concedido más importancia al espacio que al tiempo por considerar que el último generalmente está menos restringido que el primero. En consecuencia, la capacidad de esos algoritmos está limitada por requisitos de espacio.

Ahora bien, la representación que se usa es una representación orientada a arcos. En ésta, el costo de identificar los nodos origen y terminal de un arco dado es pequeño y como se verá esto se hace frecuentemente. La representación orientada a arcos más simples es la lista de arcos. Ejemplos de Listas de arcos son $O = [o_k]$ y $T = [t_k]$. Para un índice de arco k , los nodos origen y terminal son simplemente los k -ésimos elementos (celdas) en O y T , respectivamente. Similarmente, los parámetros de arco se representan en listas tales como $f = [f_k]$ y $C = [c_k]$. El costo del espacio de esta representación es $(p + 2)m$ palabras, donde, p es el número de parámetros por arco y m es el número de arcos. La Figura A.1 muestra la representación de listas de arco para la red ejemplo:

Arco (k)	1	2	3	4	5	6	7
O_k	2	5	1	1	3	2	5
t_k	4	2	3	2	4	3	1
f_k	2	1	3	1	3	0	1
C_k	2	1	3	3	5	1	2
h_k	-1	-1	1	5	3	2	1

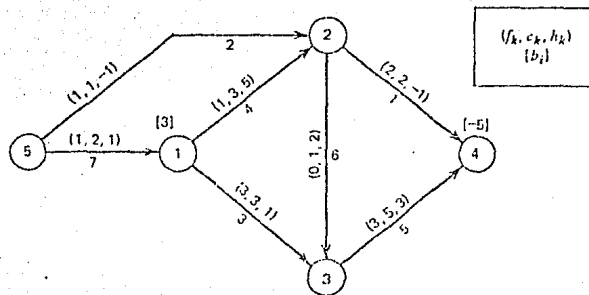


Figura A.1

Ejemplo de Representación por Listas de Arco

Para ser consistentes en la enumeración de los arcos de la red es necesario seguir algún método que la haga ordenada. El método que utiliza el algoritmo se conoce como *enumeración creciente por nodo origen*, en el cual, el primer arco enumerado que se origina en el nodo i , es aquel que tiene nodo terminal más pequeño. En la Figura A.2 se ilustra la enumeración resultante al aplicar este método, y en la Figura A.3 se presentan -- las listas de arco de la Figura A.1 cuando la enumeración es -- creciente por nodo origen.

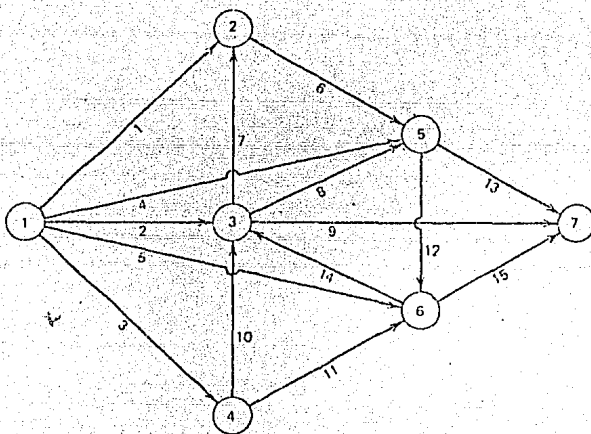


Figura A.2
Enumeración Creciente por Nodo Origen

Arco (k)	1	2	3	4	5	6	7
a_k	1	1	2	2	3	5	5
t_k	2	3	3	4	4	1	2
f_k	1	3	0	2	3	1	1
c_k	3	3	1	2	5	2	1
h_k	5	1	2	-1	3	1	-1

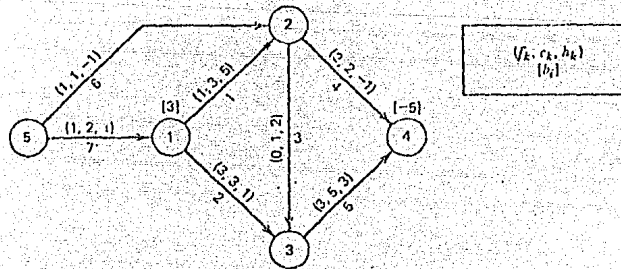


Figura A.3

Listas de Arco de la Figura A.1
Ordenadas por Nodo Origen Creciente

A.1.2. Presentación de Los Algoritmos

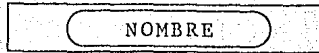
Todos los algoritmos se presentan en dos niveles de abstracción: verbal y en diagrama de flujo. Cada nivel tiene ciertas ventajas que ayudan a cubrir las desventajas del otro. El procedimiento en su declaración verbal es muy fácil de adquirir, sin embargo, las transferencias lógicas no son claras y se desprecia completamente la relación entre la estructura de datos y el procedimiento computacional. El diagrama de flujo es la representación gráfica del algoritmo, es decir, es un esquema visual que da los pasos del algoritmo y también el flujo de control entre los diferentes pasos. Por otra parte, ilustra la relación entre la estructura de datos y el procedimiento.

El formato del diagrama de flujo presentado aquí corresponde con la programación estructurada, es más compacto y más fácil de leer y comprender que el formato tradicional. Dada la diferencia entre nuestro formato y el tradicional, se hace necesario dar algunas explicaciones sobre el primero.

Una subrutina es un programa completo e independiente -- que se identifica por un nombre dentro de un rectángulo con los bordes redondeados (óvalo) como sigue:

NOMBRE

Cuando este símbolo aparece afuera de una caja (rectángulo), el diagrama de flujo que le sigue define la lógica de la subrutina. Cuando el símbolo aparece adentro de la caja, los pasos de la subrutina se ejecutan como parte del programa principal u otra subrutina. Esto equivale a la declaración "CALL" del lenguaje FORTRAN.

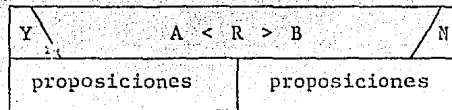


La proposición de asignación se simboliza por

$$A := B$$

Esto significa que el valor de la variable o expresión B es asignado a la variable A.

La proposición "IF" siempre aparece en una caja y tiene la forma



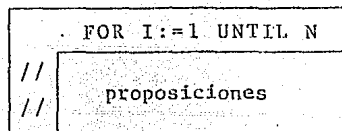
donde el símbolo R representa una relación lógica. Si la relación $A < R > B$ es cierta, se ejecutan las proposiciones de la caja que está debajo de Y, mientras que si es falsa, se ejecutan las proposiciones de la caja que está debajo de N.

La proposición "GO TO" aparece como

$$\text{NOMBRE } \circ \longrightarrow \text{NOMBRE}$$

En cualquier caso, el control se transfiere a la caja etiquetada con NOMBRE; en ella, la etiqueta se coloca en la esquina superior izquierda.

La proposición "DO" del FORTRAN se representa como



Las proposiciones de la caja interior se ejecutan para cada valor entero de la variable I , la cual se incrementa desde su valor inicial (uno en este caso) hasta su valor máximo en incrementos de 1. Los pares de diagonales indican el ámbito del "DO".

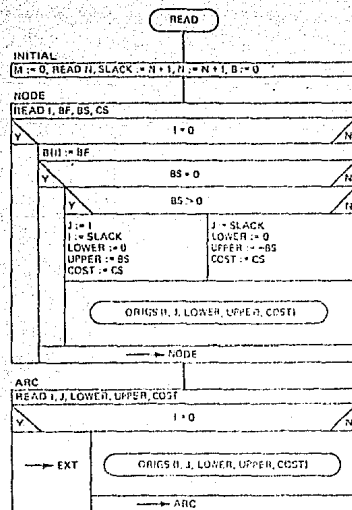
A.1.3. Lectura y Almacenamiento De La Red.

En esta sección se presentan los algoritmos READ y ORIGS. Estos se utilizan para leer, guardar y ordenar la información - para representar la estructura de la red y sus parámetros.

ALGORITMO READ

PROPOSITO: Leer y guardar los datos correspondientes a nodos y arcos para el problema de redes de flujo con costo mínimo.

1. INITIAL. Se inicializa el número de arcos $M:=0$. Se lee el número de nodos N , se crea un nodo artificial $SLACK:=N+1$ y se hacen cero todos los flujos externos, $B:=0$.
2. NODE. Leer el conjunto de datos correspondientes al nodo I . Tales datos son: el flujo externo fijo BF , la cota superior BS y el costo CS del flujo externo de holgura. Si el conjunto de datos está en blanco ($I=0$), ir al paso 3. En caso contrario, se almacena el flujo fijo. Si el flujo de holgura es cero, repetir el paso 2. Por el contrario, se crea un arco artificial y sus parámetros se guardan en una lista ordenada por nodo origen creciente.
3. ARC. Leer el conjunto de datos co-

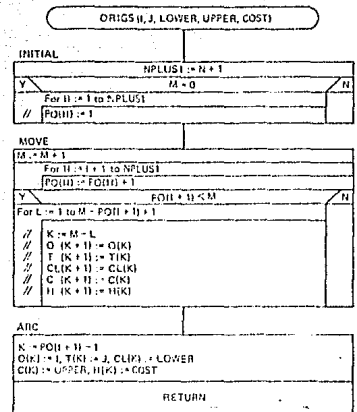


respondientes al arco (i, j) . Tales datos son: la cota inferior (LOWER), la cota superior (UPPER) y el costo por unidad de flujo (COST). Si el conjunto de datos está en blanco ($I=0$), regresar. Por el contrario, se crea un arco artificial y sus parámetros se guardan en la lista ordenada por nodo origen creciente.

ALGORITMO ORIGS

PROPOSITO: Captar el conjunto de datos correspondientes a un arco con función de costo convexa y lo almacena en una lista ordenada con enumeración creciente por nodo origen.

1. INITIAL. Si es la primera vez que el algoritmo es llamado, se asigna al apuntador P_0 de todos los nodos el valor 1. En caso contrario ir al paso 2.
2. MOVE. Incrementa el número de arcos en 1. Incrementa el apuntador P_0 de todos los nodos mayores que I en 1. Mueve todos los arcos sobre la nueva entrada a un índice mayor en la lista.
3. ARC. Inserta un arco en la última posición asignada al nodo I .



A.1.4. Representación De Arboles

Se estableció, en la sección 1.7, que los arcos de una solución básica para el problema de redes de flujo forman un árbol generador. Para ser consistentes, este árbol siempre se representa como un árbol dirigido. Puesto que el algoritmo usa una base, es importante que la estructura de datos usada para representar el árbol base permita recorrerlo y manipularlo fácilmente. Hay varias operaciones que deben ejecutarse en el árbol y son las siguientes:

1. Encontrar los nodos del subárbol con raíz (enraizado) en algún nodo i .
2. Eliminar (borrar) un arco.
3. Añadir un arco.

Ahora se presenta un ejemplo ilustrativo de la representación de árboles. Considere la red de expansión de la Figura A.6, en la cual solo se dibujan los arcos hacia adelante. En la Figura A.7 se muestran dos árboles de expansión dirigidos de esta red. El nodo raíz de cada árbol aparece abajo en la figura y se dice que tiene altura 0. Los arcos que se originan -

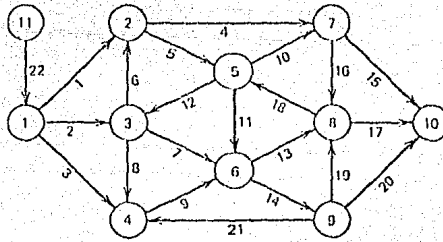


Figura A.6
Red Ejemplo

en el nodo raíz salen hacia arriba y sus nodos terminales se colocan a la misma altura o Línea horizontal. Estos tienen altura 1. Los arcos que se originan en nodos con altura 1 también se dirigen hacia arriba y terminan en nodos con altura 2. Este proceso continúa hasta alcanzar una altura en que no se origina ningún arco.

Puesto que el árbol es una subred de la red de expansión, puede incluir arcos hacia adelante y reflejados. La dirección de los arcos se escoge de manera que se asegure una trayectoria dirigida del nodo raíz a cualquier otro nodo en el árbol.

Un árbol $D_T = [N_T, M_T]$ puede describirse completamente empleando cualquiera de los métodos (Tres Etiquetas, Recorrido en Preorden, Recorrido en Inorden, Recorrido en Postorden) que existen para el manejo de estructuras de árbol. De nuevo, por consideraciones de espacio, los diseñadores de los algoritmos seleccionaron el método de <<tres etiquetas>> que es muy conocido y usado para representar árboles.

En la representación de Tres Etiquetas, a cada nodo se le asignan tres etiquetas que son suficientes para referenciarlo en la red. Tales etiquetas son las siguientes:

1. Apuntador hacia atrás, $P_B(i)$: es el único arco que termina en el nodo i .
2. Apuntador hacia adelante, $P_F(i)$: es el nodo más a la izquierda - que es terminal de un arco que se origina en el nodo i .
3. Apuntador derecho, $P_R(i)$: es el nodo que está inmediatamente a la derecha del nodo i - en la misma altura.

nodo	1	2	3	4	5	6	7	8	9	10	11
P_B	22	6	-8	3	-12	9	4	19	-21	17	0
P_F	4	7	5	9	0	0	0	10	8	0	1
P_R	0	0	6	0	2	0	0	0	3	0	0
P_D	1	4	3	2	4	3	5	4	3	5	0

nodo	1	2	3	4	5	6	7	8	9	10	11
P_B	22	6	2	3	18	7	10	13	14	20	0
P_F	3	0	6	0	7	9	0	5	10	0	1
P_R	0	0	4	0	0	2	0	0	8	0	0
P_D	1	3	2	2	5	3	6	4	4	5	0

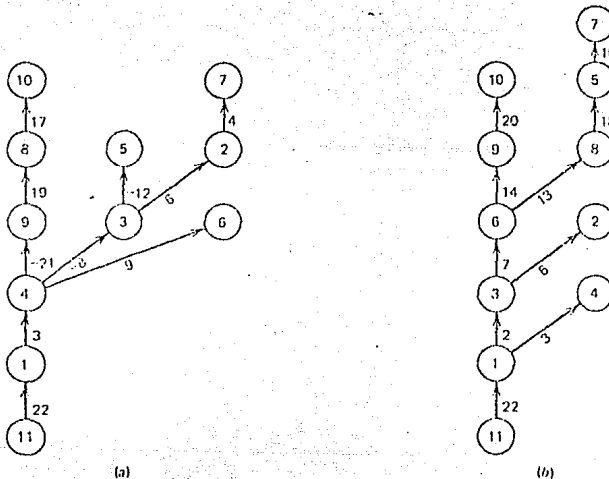


Figura A.7

Ejemplo de árboles de Expansión de la Figura A.6

En la Figura A.7 se recoge la representación de apuntadores asociada con cada árbol de expansión. $P_D(i)$ es la altura o nivel de cada nodo.

Un árbol tiene la característica de que cada nodo es terminal de a lo más un arco. Sin embargo de cada nodo puede originarse más de un arco. Por tanto, la asignación de apuntadores hacia atrás de un D_T dado es única; mientras que la asignación de apuntadores hacia adelante y derechos no es necesariamente única.

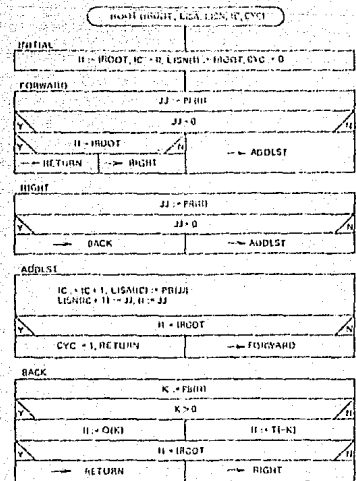
Construcción de un Subárbol con Raíz en un Nodo Dado.

Conocidas las características de un árbol, es inmediato deducir que hay un único subárbol con raíz en uno de los nodos del árbol. El algoritmo ROOT encontrará el subárbol con raíz en un nodo dado.

ALGORITMO ROOT

PROPOSITO: Encontrar la lista de arcos (LISA) y la lista de nodos (LISN) que están en el subárbol (árbol) dirigido enraizado en el nodo IROOT. La secuencia de nodos (arcos) en la lista, corresponde a un recorrido PREORDEN. Si los apuntadores detectan un ciclo se hace $CYC = 1$.

1. INITIAL. Se hace $II := IROOT$ y guarda en LISN(1).
2. FORWARD. Si el nodo II tiene su apuntador hacia delante distinto de cero, ir al paso 4. De otra manera, si el nodo II no es igual a IROOT, ir al paso 3. El subárbol consiste de un solo nodo, IROOT; regresar.
3. RIGHT. Si el nodo II no tiene apuntador hacia la derecha, deberá hacerse un rastreo hacia atrás; ir al paso 5. De otra manera ir al paso 4.
4. ADDLIST. Guarde arco y nodo encontrados en la última búsqueda. Si el nodo II es igual a IROOT existe un ciclo que regresa a IROOT, regresar. Por el contrario, ir al paso 2.
5. ATRAS. Rastrea hacia atrás del nodo II usando el apuntador hacia atrás. Si el nuevo nodo II es distinto de IROOT, ir al paso 3. De otra manera regresar.



Para ilustrar cómo trabaja ROOT considere el árbol de la

figura A.7a y el nodo raíz 4. El procedimiento busca de izquierda a derecha todas las trayectorias que se originan en el nodo-raíz. Se comienza con la trayectoria más a la izquierda trazándola mediante los apuntadores hacia adelante. Esta primera búsqueda descubre los arcos (-21, 19, 17) y los nodos (4, 9, 8, -- 10). Completada esta trayectoria, es decir cuando se encuentra un nodo sin apuntador hacia adelante, se encuentra el nodo con apuntador derecho distinto de cero que está más lejos de la raíz y se traza la trayectoria que se origina en él. Esto agrega los arcos (-8, -12) y los nodos (3, 5). El proceso continúa y al terminar el subárbol queda definido por

$$M_{T_4} = (-21, 19, 17, -8, -12, 6, 4, 9): \text{LISA}$$

$$N_{T_4} = (4, 9, 8, 10, 3, 5, 2, 7, 6) : \text{LINS}$$

Eliminación de un Arco de la Base

La eliminación de un arco de la base usando las tres etiquetas se logra mediante la modificación de la representación de apuntadores de árbol. Existen dos formas en que un arco $K_L(i_L, j_L)$ aparece:

1. Como un arco que "apunta adelante", es decir,

$$P_B(j_L) = K_L \text{ y } P_F(i_L) = j_L ;$$

2. o como un arco que "apunta hacia la derecha", es decir, existe un nodo s tal que $P_B(j_L) = K_L$ y $P_R(s) = j_L$

En la figura A.7a, los arcos -21 y -8 son arcos que apuntan hacia adelante y hacia la derecha, respectivamente.

En cualquier caso, el nodo j_L se convierte en la raíz de un subárbol cuando el arco K_L es eliminado del árbol original. Por tanto, después que se completa la eliminación, el nodo j_L no debe tener apuntador hacia atrás ni derecho. Para actualizar las tres etiquetas en el árbol original se hace el siguiente cambio, según sea el caso:

$$1. P_F(i_L) = P_R(j_L)$$

$$2. P_R(s) = P_R(j_L)$$

Luego simplemente $P_B(j_L) = 0$ y $P_R(j_L) = 0$.

En el caso 2, para identificar el nodo s (tal que $P_R(s) = (j_L)$), se ejecuta una simple búsqueda en los apuntadores derechos de los nodos comenzando en el nodo $P_F(i_L)$.

La figura A.8 muestra la eliminación de dos arcos de la Figura A.7.a. Observar que la eliminación de un arco de un árbol produce dos subárboles con todas sus propiedades. Las operaciones anteriores necesarias para eliminar un arco se ejecutan en la siguiente subrutina, DELTRE.

nodo	1	2	3	4	5	6	7	8	9	10	11
P_B	22	6	-8	3	-12	9	4	19	0	17	0
P_F	4	7	5	3	0	0	0	10	8	0	1
P_R	0	0	6	0	2	0	0	0	0	0	0
P_D	1	4	3	2	4	3	5	4	3	5	0

nodo	1	2	3	4	5	6	7	8	9	10	11
P_B	22	6	0	3	-12	9	4	19	-21	17	0
P_F	4	7	5	9	0	0	0	10	8	0	1
P_R	0	0	0	0	2	0	0	0	6	0	0
P_D	1	3	2	2	5	3	6	4	3	5	0

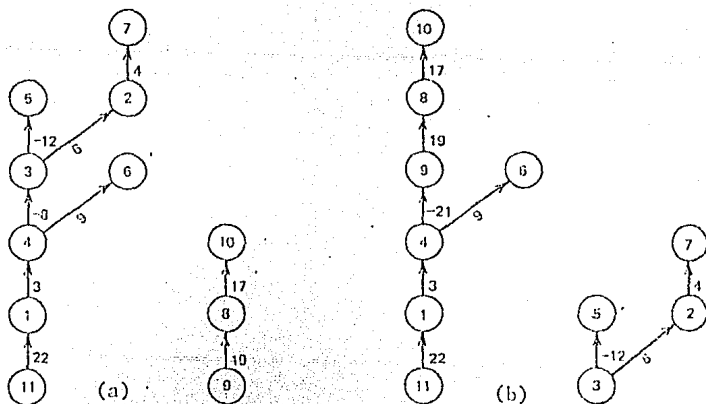


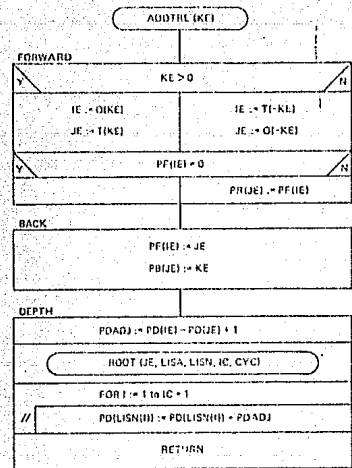
Figura A.8

Ejemplos de Eliminaciones de Arcos de la Figura A.7a (a) Eliminación del Arco -21(4,9) (b) Eliminación del Arco -8(4,3)

ALGORITMO DELTRE

PROPOSITO: Quitar un arco del árbol y actualizar su representación en términos de los apuntadores de tres etiquetas.

1. FORWARD. El arco $K_L(i_L, j_L)$ es eliminado del árbol. Si $P_F(i_L) = j_L$, entonces hacer $P_F(i_L) = P_R(j_L)$ e ir al paso 3. Si $P_F(i_L) \neq j_L$, ir al paso 2.
2. DERECHA. Encontrar el nodo s para el cual $P_R(s) = j_L$. Haga $P_R(s) = P_R(j_L)$ e ir al paso 3.
3. DELETE. Hacer el nodo j_L un nodo raíz; esto es, $P_B(j_L) = 0$ y $P_R(j_L) = 0$.



Agregando un Arco a un Bosque

Un bosque consiste de dos o más árboles dirigidos. Cuando un arco $K_E(i_E, j_E)$ se añade al bosque, tal que i_E está en un árbol y j_E es la raíz de otro, existen dos modificaciones en la representación de apuntadores. La primera distingue dos cosas:

1. Si $P_F(i_E) = 0$ entonces $P_F(i_E) = j_E$, es decir, se añade hacia adelante.
2. Si $P_F(i_E) = s$ entonces $P_R(j_E) = s$ y $P_F(i_E) = j_E$; es decir, se añade por la izquierda

La segunda modificación es:

$$P_B(j_E) = K_E$$

Esta es posible debido a que el nodo j_E es raíz de un árbol y, por tanto, su apuntador hacia atrás es cero antes de la operación. La Figura A.9 muestra el árbol que se obtiene al agregar el arco $-8(4,3)$ al bosque de la Figura A.3.b.

No fué necesario actualizar la altura de los nodos de los subárboles resultantes al eliminar un arco del árbol. Hay dos razones para este hecho. Primero, los subárboles resultantes tienen poco uso en los algoritmos que se describen más tarde; segundo, la profundidad relativa de los nodos se conserva.

Cuando se agrega un arco al bosque de árboles, si es necesario actualizar los apuntadores del subárbol (con raíz en j_E) que es unido al otro árbol. Una vez que se actualiza la representación de tres etiquetas, es fácil obtener la lista de nodos en tal subárbol, mediante el llamado de ROOT con el nodo j_E como raíz. Puesto que la altura relativa de los nodos del subárbol no ha cambiado, sólo es necesario ajustar la altura por una cantidad PDADJ en todos los nodos del subárbol. Claramente, la altura actualizada del j_E debe ser una unidad mayor que $P_D(i_E)$. Matemáticamente esta expresión es

$$P_D(j_E) + PDADJ = P_D(i_E) + 1$$

lo cual implica

$$PDADJ = P_D(i_E) - P_D(j_E) + 1$$

Agregando PDADJ a la altura de los nodos del subárbol, se completa la operación de agregar un arco y esta se describe en el algoritmo ADDTRE.

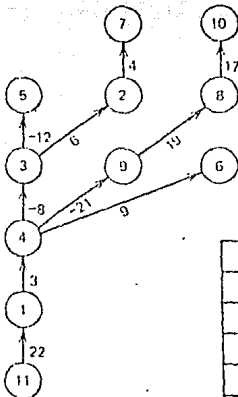


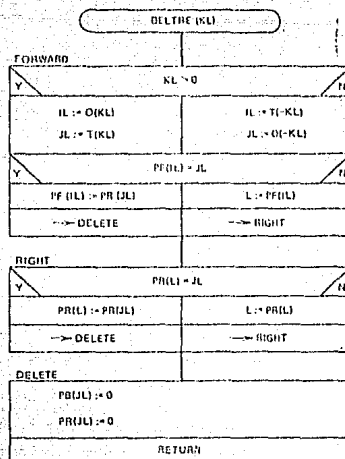
Figura A.9
Agregación del arco -8(4,3)
al Bosque de la Figura A.8b

nodo	1	2	3	4	5	6	7	8	9	10	11
P_B	22	6	-8	3	-12	9	4	19	-21	17	0
P_F	4	7	5	3	0	0	0	10	8	0	1
P_R	0	0	9	0	2	0	0	0	6	0	0
P_D	1	4	3	2	4	3	5	4	3	5	0

ALGORITMO ADDTRE

PROPOSITO: Agregar el arco $K_E(i_E, j_E)$ a un bosque. Los nodos i_E y j_E deberán estar en diferentes árboles y el nodo j_E deberá ser raíz de un árbol.

1. FORWARD. Obtenga los nodos origen y terminal de K_E . Si el apuntador hacia adelante de i_E es cero, ir al paso 2.
2. BACK. Se asigna al apuntador hacia atrás de j_E el valor de K_E y al apuntador hacia adelante de i_E el valor de j_E .
3. DEPTH. Actualizar la altura de los nodos en el subárbol con raíz en el nodo j_E .



Creación de los Apuntadores de Tres Etiquetas del Arbol Inicial.

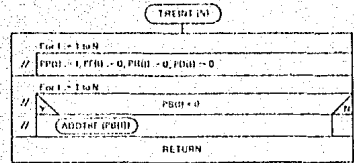
Una vez que se ha leído o se tenga un árbol inicial a través de sus apuntadores hacia atrás, es posible construir su representación en términos de los apuntadores de tres etiquetas. Esto se logra con el algoritmo TREINT.

Después de iniciar en cero los apuntadores P_F , P_R y P_D , TREINT simplemente hace un llamado de ADDTRE para cada arco en P_B .

ALGORITMO TREINT

PROPOSITO: Construir la representación de apuntadores de tres etiquetas para un árbol, teniendo conocimiento de los apuntadores hacia -- atrás.

1. Inicializa las Listas P_F , P_R y P_D en cero.
2. Llama a ADDTRE para cada apuntador hacia atrás (arco). Regresa.



A.1.5. Cambio De Flujo En Un Ciclo (Trayectoria)

El paso 5. del algoritmo CONVEX considera el cambio de flujo en un ciclo o trayectoria de la red original. El ciclo -- se describe por un conjunto de índices de arcos positivos y negativos. Por ejemplo, un ciclo en la Figura A.10 se describe -- por el conjunto de arcos $M_c = \{1, 3, -2\}$. Cambiar el flujo en esta secuencia de arcos por una cantidad Δ corresponde a incrementar el flujo en los arcos con índice positivo y decrementar el flujo en los arcos con índice negativo. De esta manera, si f_k es el flujo antes del cambio y f'_k el flujo después del cambio, se tiene que

$$f'_k = f_k + \Delta \quad \text{para } k \in M_c \text{ y } k > 0$$

$$f'_{-k} = f_{-k} - \Delta \quad \text{para } k \in M_c \text{ y } k < 0$$

Esta operación se ejecuta en el algoritmo FLOCHG

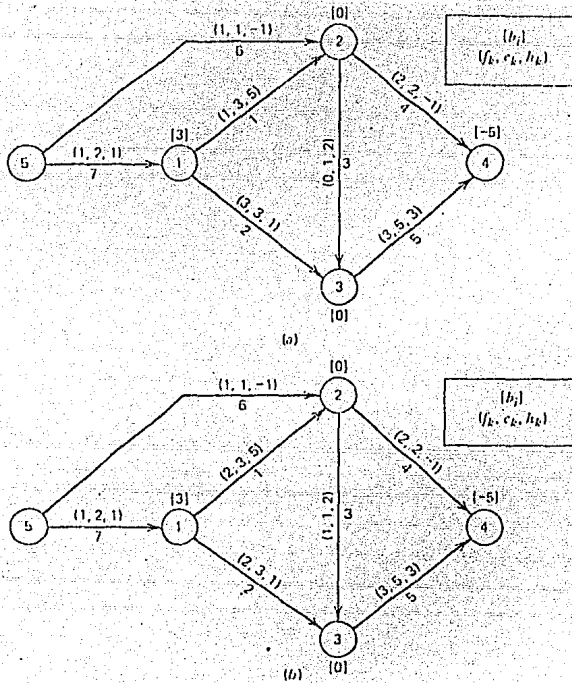


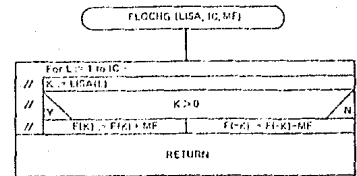
Figura A.10

Ejemplo de Cambio de Flujo en un Ciclo (a) Flujos antes del Cambio (b) Flujos después del Cambio en el ciclo, $M_c = (1,3,-2)$, por la Cantidad $\Delta=1$.

ALGORITMO FLOCHG

PROPOSITO: Cambiar los flujos en un ciclo (LISA). En general puede ser una trayectoria.

Para cada arco en el ciclo; si $k > 0$, incrementa el flujo en el arco k por MF. Si $k < 0$, disminuye el flujo en el arco $-k$ por MF.



A.2. ALGORITMO NTCNVX

NTCNVX es el nombre del algoritmo global que resuelve el problema de redes de flujo con costo mínimo convexo. Está constituido por un programa principal y 13 subrutinas, las cuales se relacionan según se ilustra en la Figura A.11. Todas las lecturas, la impresión de la representación de la red, de la solución factible inicial, del árbol de expansión y de la solución óptima se efectúan en el programa principal. La principal herramienta de este algoritmo es la subrutina CONVEX. En ésta se implementa el procedimiento de aproximación por segmentación implícita del Capítulo 2.

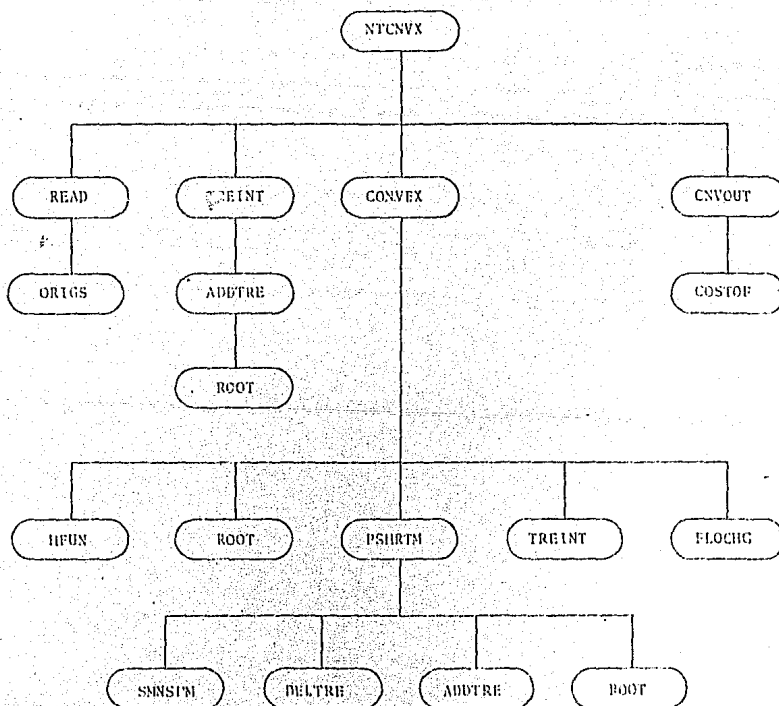


Figura A.11

Relación entre el Algoritmo Global y las Subrutinas Necesarias para Resolver Redes de Flujo con Costo Mínimo Convexo

ALGORITMO N'CNVX

PROPOSITO: Controlar las operaciones del procedimiento que resuelve el problema de redes de flujo a costo mínimo con función de costo convexa en los arcos.

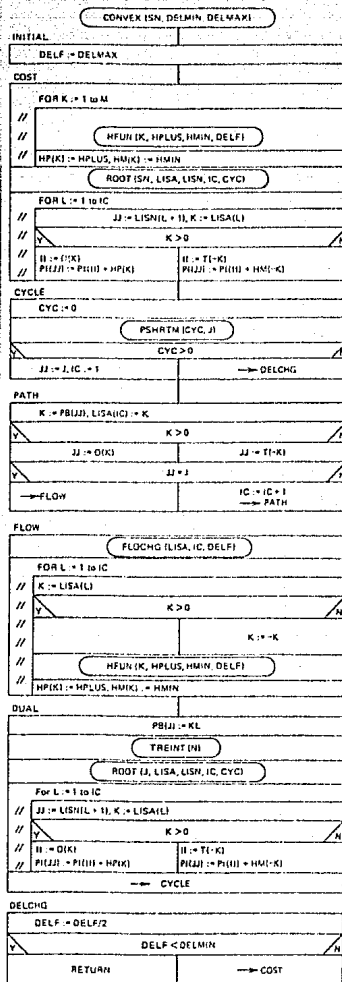
1. LECTURA. Leer y guardar los datos correspondientes a nodos y arcos creando, al mismo tiempo, la representación que usa un nodo de holgura y arcos de holgura en sustitución de los flujos externos de holgura. Leer los flujos factibles -- iniciales para la representación de red creada antes, así como el apuntador hacia atrás de cada nodo de un árbol de expansión de dicha red. Por último, leer el nodo raíz (SN) de este árbol, el cambio de flujo mínimo (DELMIN) y máximo -- (DELMAX) permitido, y las variables que controlan las opciones de reportes, ISOLUC Y IPRINT. Si ISOLUC es distinto de cero no se determina la solución. IPRINT puede asumir los valores 0 (solo se imprime la solución óptima para cada valor de Δ) y 1 (se imprime además cada solución factible como resultado de un cambio de flujo en la red, los costos marginales correspondientes y los arcos que forman un ciclo con costo total negativo si éste existe. Inicializar en cero los potenciales de nodo.

2. IMPRIMA. Imprimir el número de nodos, el número de arcos, los datos de cada nodo [flujo externo $B(I)$, potencial inicial $PI(I)$] y arco -- [nodo origen $O(K)$, nodo terminal $T(K)$, capacidad mínima $CL(k)$, capacidad máxima $C(k)$, costo lineal o término constante de la función -- convexa $H(K)$, flujo factible inicial (F_k)], el apuntador hacia -- atrás de cada nodo y las opciones de impresión.
3. SOLUCION. Encontrar la solución óptima para cada valor de Δ usando -- la subrutina CONVEX. Imprimir la -- solución óptima para el valor más-pequeño de Δ . Terminar.

ALGORITMO CONVEX

PROPOSITO: Encontrar el flujo a costo mínimo cuando el costo en los arcos es una función convexa del flujo en el arco. Las funciones convexas se definen en la subrutina HFUN. Se parte con un flujo factible inicial en el vector F y un árbol de expansión (generador) inicial.

1. INITIAL. La cantidad de flujo toma su máximo valor. ($\Delta = \text{DELF} := \text{DELMAX}$).
2. COST. Determine los costos marginales para un incremento y decremento de flujo en cada arco. Determine las variables duales π (PI) correspondientes.
3. CYCLE. Encuentre un ciclo con costo negativo en la red marginal. Si no hay alguno, termine con la solución óptima para el incremento actual; ir al paso 7. Si se encuentra un ciclo negativo ir al paso 4.
4. PATH. Encuentre los arcos en el ciclo usando los apuntadores hacia atrás.
5. FLOW. Cambie el flujo en el ciclo por la cantidad Δ . Vuelva a calcular los costos marginales en los arcos cuyo flujo fue cambiado, esto es, en los arcos del ciclo.
6. DUAL. Reformar el árbol que existía antes de que se formara el ciclo. (KL proviene de PSHRTM). Encuentre el conjunto de nodos en el



árbol enraizado en j y vuelva a -
 calcular los valores de π para -
 esos nodos. Regrese al paso 3.

7. DELCHG. Haga que el nuevo valor de Δ sea la mitad del actual. Si el -
 nuevo valor es menor que el mínimo deseado, termine. Por el contrario, regrese al paso 2.

ALGORITMO HFUN

PROPOSITO: Calcular los costos margi-
 nales en los arcos hacia adelante y
 reflejado para un cambio en el flujo
 de DELF.

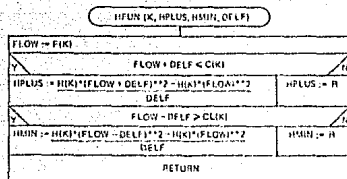
El usuario debe proporcionar HFUN. -
 El caso mostrado en el diagrama adya-
 cente calcula los costos para una --
 función cuadrática con término cons-
 tante u_k , esto es, $h_k(f_k) = u_k * f_k^2$.

HPLUS es el costo marginal para el -
 arco hacia adelante K.

HMIN es el costo marginal para el -
 arco reflejado -K.

FLOW es el flujo en el arco K.

DELF es el cambio de flujo permiti-
 do en el arco.



ALGORITMO PSHRTM

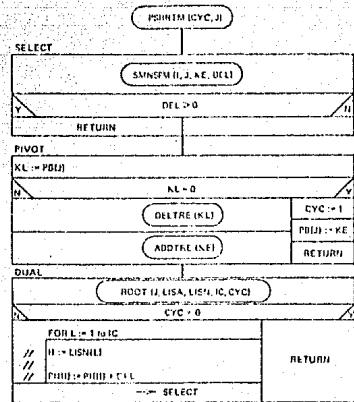
PROPOSITO: Encontrar un ciclo negativo en la red marginal. Comenzando -- con un árbol básico factible y con -- potenciales en los nodos que violan las condiciones de factibilidad del dual para algunos arcos no básicos, -- este algoritmo modifica, iterativamente, el árbol básico y los potenciales en los nodos para obtener una solución óptima. Si se encuentra un ciclo negativo se regresa $cyc = 1$.

1. SELECT. Encontrar un arco no básico $K_E(i, j)$ para el cual $\pi_i + h_{K_E} < \pi_j$. Si no existe tal arco, terminar; -- la solución es óptima. En caso contrario defina

$$d_E := \pi_i + h_{K_E} - \pi_j.$$

2. PIVOT. Sea $KL(i', j)$ el arco básico que termina en el nodo j . Borrar -- este arco del árbol básico. Añadir K_E al árbol básico.

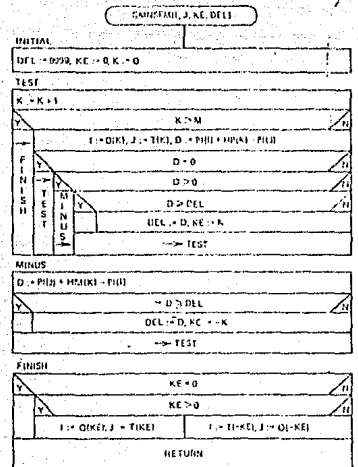
3. DUAL. Encontrar los nodos en el árbol con raíz en el nodo j . Si se encuentra un ciclo negativo en este paso ($CYC=1$), regresar; los flujos actuales no son óptimos. En caso contrario, sume d_E a los potenciales de todos los nodos en el -- conjunto y regresar el paso 1.



ALGORITMO SMNSPM

PROPOSITO: Encontrar el arco admisible (no básico) que tenga el valor más negativo de $D := \pi_i + h_k - \pi_j$. Esto es equivalente al criterio de entrada usual en el algoritmo SIMPLEX.

1. Sean $DEL := 9999$ y $KE := 0$
2. Para cada arco $k(i, j)$ calcular $D := \pi_i + h_k - \pi_j$. Si $D = 0$, tomar el siguiente arco. Si $D < 0$, el arco K es admisible, y si $D < DEL$, hacer $KE := K$ y $DEL = D$. Si $D > 0$, el arco $-K$ es admisible, y si $-D < DEL$, hacer $KE := K$ y $DEL := -D$. Tomar el siguiente arco.
3. Si $KE > 0$, hacer $i := o(KE)$ y $j := t(KE)$. Si $KE < 0$, hacer $i := t(-KE)$ y $j := o(-KE)$.



A.3 CODIFICACION FORTRAN IV DEL PROGRAMA NTCNVX

```

100 %
200 % PROGRAMA PRINCIPAL NTCNVX
300 %
400 %
500 % *****
600 % *
700 % * ESTA ES LA CODIFICACION FORTRAN DEL ALGORITMO : *
800 % *
900 % * ***** *
1000 % * * NTCNVX * *
1100 % * ***** *
1200 % *
1300 % * COMO SE PRESENTO EN EL APENDICE A (SECCION A.2) DE LA *
1400 % *
1500 % * ** TESIS ** *
1600 % *
1700 % * REALIZADA POR EL ING. MARIO PARRA UGALDE PARA OBTEN- *
1800 % * NER EL GRADO DE MAESTRO EN *
1900 % *
2000 % * INVESTIGACION DE OPERACIONES *
2100 % *
2200 % * DIVISION DE ESTUDIOS DE POSGRADO FACULTAD DE INGENIERIA *
2300 % * ( D.E.P.F.I. ) *
2400 % *
2500 % * UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO ( U.N.A.M. ) *
2600 % * AGOSTO DE 1982 *
2700 % *
2800 % * PROPOSITO : *
2900 % * NTCNVX SE DISENO PARA IMPLEMENTAR EL METODO DE " APRO- *
3000 % * XIMACION POR SEGMENTACION IMPLICITA " DE LA SECCION *
3100 % * 2.5 QUE RESUELVE EL PROBLEMA DE REDES DE FLUJO CON *
3200 % * COSTO MINIMO DONDE LOS ARCOS PUEDEN TENER UNA FUNCION *
3300 % * DE COSTO LINEAL O CONVEXA . *
3400 % * COSTO CONVEXA. *
3500 % *
3600 % * LAS SUBROUTINAS QUE COMPRENDE ESTE PROGRAMA SON : *
3700 % *
3800 % * READ , ORIG , TREINT , TREINT , ADDRE , DELTRE *
3900 % * ROOT , CONVEX , HFUN , PSHTM , SMNSPM , FLOCHG *
4000 % * CNVOUT Y COSTOF . *
4100 % *
4200 % * EL ANALISTA (USUARIO) DEBE PROPORCIONAR *
4300 % *
4400 % * 1. LA RED , *
4500 % * 2. UNA SOLUCION FACTIBLE INICIAL *
4600 % * 3. UN ARBOL DE EXPANSION (APROPIADO SI ES NECESARIO) *
4700 % * DE LA RED TRANSFORMADA (CON UN NODO DE HOLGURA) , *
4800 % * 4. LOS APUNTADORES HACIA ATRAS DEL ARBOL DE EXPANSION , *
4900 % * 5. EL CAMBIO DE FLUJO MINIMO Y MAXIMO , *
5000 % * 6. LA SUBROUTINA * HFUN * APROPIADA PARA CALCULAR LOS *
5100 % * COSTOS MARGINALES DE LOS ARCOS HACIA ADELANTE Y RE- *
5200 % * FLEJADOS , Y *
5300 % * 7. LA SUBROUTINA * COSTOF * APROPIADA PARA CALCULAR *
5400 % * CADA TERMINO DE LA FUNCION OBJETIVO QUE SE MINIMIZA. *
5500 % *
5600 % *
5700 % *****
5800 %
5900 % COMMON /A/O(120)/B/T(120)/C/PR(120)/D/FF(120)/E/FR(120)/G/H,N
6000 % * /K/C(120)/L/H(120)/M/B(120)/J/PO(120)
6100 % COMMON /H/CL(120)/O/F(120)/F/F1(120)
6200 % COMMON /Z/EPNEG,EPPOS
6300 % COMMON /O/PI(120)
6400 % COMMON /REPORT/PRINT
6500 %
6600 % DIMENSION LISA(120),LISH(120)
6700 % INTEGER PP,PD,CYC,AD
6800 % INTEGER O,T,PO,FF,FR,FO
6900 % INTEGER SC,SE,VR,SH,IR,V

```

```

7000 %
7100 %
7200 %
7300 % LECTURA
7400 %
7500     EPNEG = -.000001
7600     EPPOS = .000001
7700 %
7800     CALL READ
7900 %
8000 % LECTURA DEL FLUJO FACTIBLE EN CADA ARCO DE LA R. TRANSFORMADA
8100 %
8200     READ 11, (F(K), K=1, N)
8300     11 FORMAT (8F10.0)
8400 %
8500 % LECTURA DEL APUNTAOR HACIA ATRAS DE CADA NODO DE UN
8600 % ARBOL DE EXPANSION DE LA RED TRANSFORMADA
8700 %
8800     READ 37, (PB(I), I=1, N)
8900     37 FORMAT (20I4)
9000 %
9100 % CREACION DE LA REPRESENTACION DE APUNTAORES DE TRES ETIQUETAS
9200 %
9300     CALL TREINT(N)
9400 %
9500     DO 9 I =1, N
9600     9 PI(I) = 0.
9700 %
9800     READ 10, SN, DELMIN, DELMAX
9900     10 FORMAT (15, 2F10.0)
10000 %
10100     READ 12, ISOLUC, IPRINT
10200     12 FORMAT(2I1)
10300 %
10400 % IMPRESION
10500 %
10600     PRINT 1, N, N
10700 %
10800     1 FORMAT(1H1, 3(/), 10X, 'RED DE FLUJO CON COSTO CONVEXO', /, 10X,
10900     * 30('-',) 3(/), 10X, 'REPRESENTACION DE LA RED TRANSFORMADA',
11000     * 3(/), 10X, 'NUMERO DE NODOS = ', I4, 10X, 'NUMERO DE ARCOS = ', I4, /)
11100 %
11200     PRINT 2, (I, B(I), PI(I), PB(I), I=1, N )
11300 %
11400     2 FORMAT( 2(/), 10X, 'NODO', 4X, 'FLUJO EXTERNO FIJO', 4X, 'POTENCIAL',
11500     * ' INICIAL', 4X, 'APUNTAOR HACIA ATRAS', /, 12X, 'I', 12X, 'B(I)',
11600     * 17X, 'PI(I)', 16X, 'PB(I)', // (9X, I4, 5X, F11.2, 11X, F11.2, 18X, I5) )
11700 %
11800     PRINT 3, (K, B(K), T(K), CL(K), C(K), H(K), F(K), K=1, N )
11900 %
12000     3 FORMAT(4(/), 10X, 'ARCO', 4X, 'N. ORIGEN', 4X, 'N. TERMINAL', 5X, 'COTA INF',
12100     * ', ', 5X, 'COTA SUP.', 6X, 'COSTO', 4X, 'FLUJO FACTIBLE INICIAL', /,
12200     * 12X, 'K', 7X,
12300     * 'B(K)', 9X, 'T(K)', 9X, 'CL(K)', 9X, 'C(K)', 11X, 'H(K)', 11X, 'F(K)', //,
12400     * (8X, I5, 6X, I5, 9X, I5, 3X, F11.2, 2X, F11.2, 4X, F11.2, 4X, F11.2) )
12500 %
12600     PRINT 4, SN, DELMIN, DELMAX
12700     4 FORMAT(3(/), 10X, 'NODO RAIZ DEL ARBOL INICIAL ', I4, 10X,
12800     * 'CAMBIO DE FLUJO MINIMO = ', F9.3, /, 52X, 'CAMBIO DE FLUJO MAXIMO = '
12900     * , F9.3)
13000 %
13100     PRINT 5, ISOLUC, IPRINT
13200     5 FORMAT(2(/), 10X, 'OPCIONES DE REPORTE : ISOLUC = ', I1, 10X,
13300     * ' IPRINT = ', I1 )

```

```

14000      SUBROUTINE READ
14900 %
15000      COMMON/H/B(120)/G/H,N/B/T(120)
15100      INTEGER SLACK,T
15200      REAL LOWER
15300 %
15400 % *****
15500 % * * * * *
15600 % *   PROPOSITO : * * *
15700 % *           LEER Y ALMACENAR LOS DATOS CORRESPONDIENTES * *
15800 % *   A NODOS Y ARCOS PARA EL PROBLEMA DE REDES DE FLUJO CON *
15900 % *   COSTO MINIMO CONVEXO , * *
16000 % * * * * *
16100 % *****
16200 %
16300 %
16400 % INICIAR
16500 %
16600      READ 4,N
16700      4 FORMAT (I5)
16800 %
16900      M=0
17000      SLACK=N+1
17100      N=N+1
17200      DO 5 I=1,N
17300      5 B(I)=0
17400 %
17500 % NODO
17600 %
17700      2 READ 6,I,BF,BS,CS
17800      6 FORMAT (I5,3F10.0)
17900 %
18000      IF(I.EQ.0) GO TO 3
18100      B(I)=BF
18200      IF ( ABS(BS) .LE. .0001) GO TO 2
18300      IF(BS.GT.0.) GO TO 7
18400      J=SLACK
18500      LOWER=0
18600      UPPER=-BS
18700      COST=CS
18800      GO TO 8
18900 %
19000      7 J=I
19100      I=SLACK
19200      LOWER=0
19300      UPPER=BS
19400      COST=CS
19500 %
19600      8 CALL ORIGS(I,J,LOWER,UPPER,COST)
19700 %
19800      GO TO 2
19900 %
20000 % ARCO
20100 %
20200      3 READ 9,I,J,LOWER,UPPER,COST
20300      9 FORMAT (2I5,3F10.0)
20400 %
20500      IF(I.EQ.0) RETURN
20600 %
20700      CALL ORIGS(I,J,LOWER,UPPER,COST)
20800 %
20900      GO TO 3
21000 %
21100      END
21200
21300
21400
21500
21600
21700
21800
21900

```

```

22000 SUBROUTINE CONVEX (SN,DELHIN,DELMAX)
22100 %
22200 COMMON /Z/EPNEG,EPOS,O/F(120)/A/O(120)/D/T(120)/P/PI(120)
22300 COMMON /C/PR(120)/G/B,N/WB/DELF/JW1/HP(120)/JW2/HH(120)
22400 COMMON /JW3/KL,KE
22500 COMMON /REPORT/IPRINT
22600 DIMENSION LISA(120),LISH(120)
22700 INTEGER PB,SN,O,T,CYC
22800 LOGICAL NISHA
22900 %
23000 % *****
23100 % * * * * *
23200 % * PROPOSITO : * * * * *
23300 % * ENCUNTRAR EL FLUJO A COSTO MINIMO CUANDO EL * * * * *
23400 % * COSTO EN LOS ARCOS ES UNA FUNCION LINEAL O CONVEXA DEL * * * * *
23500 % * FLUJO EN EL ARCO. LAS FUNCIONES CONVEXAS SE DEFINEN EN * * * * *
23600 % * LA SUBROUTINA HFUN . SE PARTE DE UN FLUJO FACTIBLE * * * * *
23700 % * INICIAL EN EL VECTOR F Y UN ARROL DE EXPANSION INI - * * * * *
23800 % * CIAL. * * * * *
23900 % * * * * *
24000 % *****
24100 %
24200 %
24300 % INITIAL
24400 %
24500 DELF = DELMAX
24600 ITERAC = 0
24700 %
24800 IF( IPRINT.EQ.2 )PRINT 20, DELF
24900 %
25000 20 FORMAT(1H1,4(/),10X,'VALOR DEL CAMBIO DE FLUJO ',F8.2,/,
25100 * 10X,3A('-'))
25200 %
25300 % COST
25400 %
25500 12 CONTINUE
25600 DO 1 K = 1,H
25700 %
25800 CALL HFUN07(K,HPLUS,HHIN,DELF)
25900 %
26000 HP(K) = HPLUS
26100 HH(K) = HHIN
26200 1 CONTINUE
26300 %
26400 CALL ROOT (SN,LISA,LISH,IC,CYC)
26500 %
26600 DO 2 L = 1,IC
26700 JJ = LISH(L+1)
26800 N=LISA(L)
26900 IF ( K .LT. 0) GO TO 3
27000 II = 0(K)
27100 PI(JJ) = PI(II) + HP(K)
27200 GO TO 2
27300 %
27400 3 II = I(-K)
27500 PI(JJ) = PI(II) + HH(-K)
27600 2 CONTINUE
27700 %
27800 % CYCLE
27900 %
28000 CYC = 0
28100 %
28200 CALL PSHRTH ( CYC,J)
28300 %
28400 IF ( CYC .LE. 0) GO TO 4
28500 NISHA = .FALSE.
28600 JJ = J
28700 IC = 1

```

```

28800 Z
28900 Z PATH
29000 Z
29100     6 K = PB(JJ)
29200     LISA(IC) = K
29300     IF ( K .GT. 0 ) JJ = 0(K)
29400     IF ( K .LT. 0 ) JJ = T(-K)
29500     IF ( JJ .EQ. J ) GO TO 5
29600     IC = IC + 1
29700     GO TO 6
29800 Z
29900     5 CONTINUE
30000 Z
30100 Z FLOW
30200 Z
30300     ITERAC = ITERAC+1
30400 Z
30500     IF( IPRINT.EQ.2 ) PRINT 21, ITERAC, (K,F(K),HP(K),HM(K), K=1,H)
30600     21  FORMAT(1(/),10X,'ITERACION ',I3,2(/),10X,'ARCO',7X,'FLUJO',8X,
30700     * ' COSTO MARGINAL POR INCREMENT-',3X,'COSTO MARGINAL POR'
30800     * ' DECREMENT-', /,12X,'K',9X,'F(K)',9X,'TD DE FLUJO EN EL ARCO  K'
30900     * ',6X,'TD DE FLUJO EN EL ARCO  -K', 2(/),
31000     * (8X,I5,2X,F11.2,14X,F11.2,20X,F11.2) )
31100 Z
31200     CALL FLOCHG (LISA,IC,DELF)
31300 Z
31400     DO 7 L=1,IC
31500     K = LISA(L)
31600     IF ( K .GT. 0 ) GO TO 8
31700     K=-K
31800 Z
31900     8 CALL HFUNO7(K,HPLUS,HMIN,DELF)
32000 Z
32100     HP(K) = HPLUS
32200     HM(K) = HMIN
32300     7 CONTINUE
32400 Z
32500 Z DUAL
32600 Z
32700     PR(JJ) = KL
32800 Z
32900     CALL TREINT(N)
33000 Z
33100     CALL ROOT (J,LISA,LISN,IC,CYC)
33200 Z
33300     DO 9 L=1,IC
33400     JJ = LISN(L+1)
33500     K = LISA(L)
33600     IF ( K .LT. 0 ) GO TO 10
33700     II = 0(K)
33800     PI(JJ) = PI(II) + HP(K)
33900     GO TO 9
34000 Z
34100     10 II = T(-K)
34200     PI(JJ) = PI(II) + HM(-K)
34300     9 CONTINUE
34400 Z
34500     IF( IPRINT.NE.2 )GO TO 2
34600 Z
34700     PRINT 22, DELF
34800     22  FORMAT( 2(/),10X,'SOLUCION FACTIBLE ( CAMBIO DE FLUJO = ',F10.4,
34900     * ' ', /, 10X,5I(' ') )
35000 Z
35100     CALL CHVOUT
35200 Z
35300     GO TO 2

```



```
35400 %  
35500 % DELCHG  
35600 %  
35700 % 4 CONTINUE  
35800 %  
35900 % IF( IPRINT,LE.0 ) GO TO 14  
36000 % PRINT 23, DELF  
36100 % 23 FORMAT(1H1,3(/),10X,'SOLUCION OPTIMA PARA EL VALOR DEL CAMBIO DE'  
36200 % * ' FLUJO ',F8.2, '/',10X,5B('-' )  
36300 %  
36400 % IF( MISMA )GO TO 11  
36500 %  
36600 % CALL CHVOUT  
36700 %  
36800 % GO TO 14  
36900 %  
37000 % 11 DELFA = 2.*DELF  
37100 % PRINT 15, DELFA  
37200 % 15 FORMAT(2(/),10X,'LA MISMA QUE PARA EL CAMBIO DE FLUJO ',F8.2 )  
37300 %  
37400 % 14 DELF = DELF/2.  
37500 % IF ( DELF .LT. DELMIN) GO TO 999  
37600 %  
37700 % IF( IPRINT.EQ.2)PRINT 20, DELF  
37800 %  
37900 % ITERAC = 0  
38000 % MISMA = .TRUE.  
38100 % GO TO 12  
38200 %  
38300 % 999 CONTINUE  
38400 % RETURN  
38500 % END  
38600 %  
38700 %  
38800 %  
38900 %  
39000 %
```

```

39100 SUBROUTINE FSHRTH(CYC,J)
39200 Z
39300 COMMON /C/PB(120)/F/PI(120)/G/M,H/Z/EPNEG,EPPOS
39400 COMMON /JM3/KL,KE
39500 DIMENSION LISA(120),LISN(120)
39600 INTEGER PB,CYC
39700 Z
39800 Z *****
39900 Z * * * * *
40000 Z * PROPOSITO : *
40100 Z * ENCONTRAR UN CICLO NEGATIVO EN LA RED MARGI *
40200 Z * NAL. COMENZANDO CON UN ARBOL BASICO FACTIBLE Y CON PO- *
40300 Z * TENCIALES EN LOS NODOS QUE VIOLAN LAS CONDICIONES DEL *
40400 Z * FACTIBILIDAD DEL DUAL PARA ALGUNOS ARCOS NO BASICOS, *
40500 Z * ESTE ALGORITMO MODIFICA, ITERATIVAMENTE, EL ARBOL RA- *
40600 Z * SICO Y LOS POTENCIALES PARA OBTENER UNA SOLUCION OPTI- *
40700 Z * MA. SI SE ENCUENTRA UN CICLO NEGATIVO SE REGRESA CYC=1 *
40800 Z * * * * *
40900 Z *****
41000 Z
41100 Z
41200 Z SELECT
41300 Z
41400 Z 20 CALL SHNSPH(I,J,KE,DEL)
41500 Z
41600 Z IF ( ABS(DEL) .LT.EPPOS) GO TO 40
41700 Z
41800 Z PIVOT
41900 Z
42000 Z KL=PB(J)
42100 Z IF(KL.EQ.0)GO TO 100
42200 Z
42300 Z CALL DELTRE(KL)
42400 Z
42500 Z CALL AUDTRE(KE)
42600 Z
42700 Z DUAL
42800 Z
42900 Z CALL ROOT(J,LISA,LISN,IC,CYC)
43000 Z
43100 Z IF( CYC.NE.0 )GO TO 40
43200 Z
43300 Z DO 30 L=1,IC+1
43400 Z II=LISN(L)
43500 Z FI(II)=PI(II)+DEL
43600 Z 30 CONTINUE
43700 Z GO TO 20
43800 Z
43900 Z 100 CYC = 1
44000 Z PB(J) = KE
44100 Z 40 CONTINUE
44200 Z RETURN
44300 Z END
44400 Z
44500 Z
44600 Z
44700 Z
44800 Z

```

```

44900      SUBROUTINE SMNSPH(I,J,KE,DEL)
45000 %
45100      COMMON /A/D(120)/R/T(120)/G/H/H/L/H(120)/P/PI(120)
45200      COMMON /Z/EFNEG,EFFOS/JW1/HP(120)/JW2/HH(120)
45300      INTEGER O,T
45400 %
45500 % *****
45600 % *
45700 % * PROPOSITO :
45800 % * ENCONTRAR EL ARCO ADMISIBLE ( NO BASICO )
45900 % * QUE TENGA EL VALOR MAS NEGATIVO DE
46000 % *  $D = PI(I)HH(K) - PI(J)$  , ESTO ES EQUIVALENTE AL CRITERIO
46100 % * DE ENTRADA USUAL EN EL ALGORITMO SIMPLEX ( COEFICIEN-
46200 % * TES DE COSTO  $C(J) - Z(J)$  )
46300 % *
46400 % *****
46500 %
46600      DEL=9999
46700      KE=0
46800 %
46900 % TEST
47000 %
47100      DO 20 K=1,H
47200      I=D(K)
47300      J=T(K)
47400      D=PI(I)HH(K)-PI(J)
47500      IF ( ABS(D) .LT. EFFOS) GO TO 20
47600      IF ( D .GE. 0.) GO TO 1
47700      IF ( D .GE. DEL ) GO TO 20
47800      DEL=D
47900      KE=K
48000      GO TO 20
48100 %
48200 % MINUS
48300 %
48400      1 D = PI(I) - HH(K) - PI(J)
48500      IF(-D.GE,DEL)GO TO 20
48600      DEL=-D
48700      KE=-K
48800      20 CONTINUE
48900 %
49000 % FINISH
49100 %
49200      IF(KE.EQ.0)GO TO 55
49300      IF(KE.GT.0)GO TO 50
49400      I=T(-KE)
49500      J=D(-KE)
49600      GO TO 40
49700 %
49800      50 I=D(KE)
49900      J=T(KE)
50000      40 CONTINUE
50100      55 IF (DEL .GT. 0.) DEL = 0.
50200 %
50300      RETURN
50400      END
50500
50600
50700
50800
50900

```

```

51000      SUBROUTINE HFUN (K,HPLUS,HMIN,DELFF)
51100 %
51200      COMMON /L/H(120)/O/F(120)/K/C(120)/N/CL(120)
51300 %
51400 % *****
51500 % *
51600 % * PROPOSITO :
51700 % *          CALCULAR LOS COSTOS MARGINALES EN LOS ARCOS *
51800 % * HACIA ADELANTE Y REFLEJADO PARA UN CAMBIO DE FLUJO DE *
51900 % * DELF .
52000 % * EL ANALISTA DEBE PROPORCIONAR ESTA SUBROUTINA. LA QUE *
52100 % * SE PRESENTA A CONTINUACION CALCULA LOS COSTOS PARA UNA *
52200 % * FUNCION CUADRATICA CON TERMINO CONSTANTE H(K), ES DE - *
52300 % * CIR, H = H(K)*F(K)**2 ,
52400 % *
52500 % *****
52600 %
52700      IF ((F(K) +DELFF) .LE. C(K)) GO TO 1
52800      HPLUS = 9999.
52900      GO TO 2
53000 %
53100      1 HPLUS=(H(K)*(F(K)+DELFF)**2-H(K)*F(K)**2)/DELFF
53200      2 IF ( (F(K) - DELFF) .GE. CL(K)) GO TO 3
53300      HMIN = 9999.
53400      GO TO 999
53500 %
53600      3 HMIN= (H(K)*(F(K)-DELFF)**2-H(K)*F(K)**2)/DELFF
53700 %
53800      999 CONTINUE
53900      RETURN
54000      END
54100
54200
54300
54400
54500
54600      SUBROUTINE COSTOF ( K,COSTOK )
54700 %
54800      COMMON /L/H(100)/O/F(100)/K/C(100)/N/CL(100)
54900 %
55000 % *****
55100 % *
55200 % * PROPOSITO :
55300 % *          CALCULAR EL TERMINO DE LA FUNCION OBJETIVO *
55400 % * CORRESPONDIENTE AL ARCO K PARA UNA RED EN LA QUE TODOS *
55500 % * SUS ARCOS TIENEN UNA FUNCION DE COSTO CUADRATICA ,
55600 % *
55700 % *****
55800 %
55900      COSTOK = H(K)*F(K)**2
56000 %
56100      RETURN
56200      END
56300
56400
56500
56600
56700
56800

```

```

56900      SUBROUTINE ORIGS (I,J,LOWER,UPPER,COST)
57000 Z
57100      COMMON /A/O(120)/B/T(120)/J/PO(120)/K/C(120)/H/CL(120)
57200      COMMON /L/H(120)/H/B(120)/G/H,H/O/F(120)
57300      INTEGER PO,O,T
57400      REAL LOWER
57500 Z
57600 Z      *****
57700 Z      *
57800 Z      *   PROPOSITO :
57900 Z      *           CAPTAR LOS DATOS CORRESPONDIENTES A UN ARCO *
58000 Z      *   CON FUNCION DE COSTO CONVEXA Y LOS ALMACENA EN UNA *
58100 Z      *   LISTA ORDENADA CON ENUMERACION CRECIENTE POR NODO ORI- *
58200 Z      *   GEN ,
58300 Z      *
58400 Z      *****
58500 Z
58600      NPLUS1=N+1
58700 Z
58800 Z      INITIAL
58900 Z
59000      IF( N.NE.0 )GO TO 10
59100      DO 5 II = 1,NPLUS1
59200      5 PO(II) = 1
59300 Z
59400 Z      MOVE
59500 Z
59600      10 H = M + 1
59700      IPLUS1 = I + 1
59800      DO 15 II = IPLUS1,NPLUS1
59900      15 PO(II) = PO(II) + 1
60000      IF( PO(I+1).GT.H )GO TO 25
60100      NP01 = H - PO(I+1) + 1
60200 Z
60300      DO 20 L=1,NP01
60400      K=H-L
60500      O(K+1)=O(K)
60600      T(K+1)=T(K)
60700      CL(K+1)=CL(K)
60800      C(K+1)=C(K)
60900      20 H(K+1)=H(K)
61000 Z
61100 Z      ARC
61200 Z
61300      25 K=PO(I+1) -1
61400      O(K) = I
61500      T(K)=J
61600      CL(K) = LOWER
61700      C(K) = UPPER
61800      H(K) = COST
61900 Z
62000      RETURN
62100      END
62200
62300
62400
62500
62600
62700

```

```

62800      SUBROUTINE FLOCHG(LISA,IC,MF)
62900 %
63000      COMMON /O/F(120)/G/H,N
63100      COMMON /REPORT/IPRINT
63200      DIMENSION LISA(120)
63300      REAL MF
63400 %
63500 %
63600 % *****
63700 % *
63800 % * PROPOSITO :
63900 % * CAMBIAR LOS FLUJOS EN UN CICLO O TRAYECTO - *
64000 % * RIA ( LISA ) .
64100 % *
64200 % *****
64300 %
64400      IF( IPRINT.EQ.2) PRINT 1
64500 %
64600      1  FORMAT(2(/,10X,'ARCO DEL CICLO',12X,'FLUJO DESPUES DEL CAMBIO',
64700      * /,14X,'K 0 -K',27X,'F(K)', /)
64800 %
64900      DO 100 L=1,IC
65000      K=LISA(L)
65100      IF ( K .GT. 0 ) GO TO 98
65200      F(-K)=F(-K)-MF
65300 %
65400      IF( IPRINT.EQ.2) PRINT 2, K,F(-K)
65500 %
65600      2  FORMAT(14X,I6,20X,F11.2)
65700      GO TO 100
65800 %
65900      98 CONTINUE
66000      F(K)=F(K)+MF
66100 %
66200      IF( IPRINT.EQ.2 ) PRINT 2, K,F(K)
66300 %
66400      100 CONTINUE
66500 %
66600      RETURN
66700      END
66800
66900
67000
67100
67200

```

```

67300      SUBROUTINE CNVOUT
67400 Z
67500      COMMON /G/H,N/P/PI(120)/K/C(120)/L/H(120)/N/CL(120)
67600      COMMON /O/F(120)/C/PB(120)/A/O(120)/B/T(120)
67700      INTEGER O,T,PB
67800 Z
67900 Z *****
68000 Z *
68100 Z * PROFESITO :
68200 Z * IMPRIMIR UNA SOLUCION ( FACTIBLE O OPTIMA ).*
68300 Z * EN ESTA SUBROUTINA SE HACE EL LLAMADO DE LA SUBROUTINA *
68400 Z * COSTOF , LA CUAL DEBE SER PROPORCIONADA POR EL ANALIS- *
68500 Z * TA .
68600 Z *
68700 Z *****
68800 Z
68900 Z
69000      PRINT 101, (I,PI(I),PB(I), I=1,N )
69100 Z
69200      101 FORMAT(2(/),10X,'NODO',5X,'POTENCIAL',5X,'PUNTAJOR HACIA ATRAS',
69300      * /, 12X,'I',8X,'PI(I)',15X,'PB(I)',2(/),(8X,15,2X,F11.2,15X,15) )
69400 Z
69500      PRINT 103
69600      103 FORMAT(//,10X,'ARCO',4X,'N.ORIGEN',4X,'N.TERMINAL',5X,'COTA INF.',
69700      * ,5X,'COTA SUP.',6X,'COSTO',8X,'FLUJO',4X,'COSTO DEL FLUJO',/,
69800      * 12X,'K',7X,'O(K)',9X,'T(K)',9X,'CL(K)',9X,'C(K)',10X,'H(K)',9X,
69900      * 'F(K)', / )
70000 Z
70100      COSTOT = 0.
70200      DO 110 K=1,N
70300 Z
70400      CALL COSOF(K,COSTOK)
70500 Z
70600      PRINT 106, K,O(K),T(K),CL(K),C(K),H(K),F(K),COSTOK
70700      106 FORMAT(8X,15,6X,15,8X,15,3X,F11.2,2X,F11.2,3X,F11.2,2X,F11.2,2X,
70800      * F15.2 )
70900 Z
71000      COSTOT = COSTOT+COSTOK
71100      110 CONTINUE
71200 Z
71300      PRINT 107, COSTOT
71400      107 FORMAT( 1(/),71X,'COSTO TOTAL DEL FLUJO ',F15.2 )
71500 Z
71600      RETURN
71700      END
71800
71900
72000
72100
72200
72300

```

```

72400      SUBROUTINE ROOT(IROOT,LISA,LISH,IC,CYC)
72500 Z
72600      COMMON /A/O(120)/B/T(120)/C/FR(120)/D/PF(120)/F/PR(120)
72700      DIMENSION LISA(120),LISH(120)
72800      INTEGER O,T,PR,PF,FR,CYC
72900 Z
73000 Z      *****
73100 Z      *
73200 Z      *   PROPOSITO :
73300 Z      *   ENCONTRAR LA LISTA DE ARCOS (LISA) Y LA LIS *
73400 Z      *   TA DE NODOS (LISH) QUE ESTAN EN EL SUBARROL (ARBOL) *
73500 Z      *   DIRIGIDO CON RAIZ EN EL NODO IRROOT .
73600 Z      *   LA SECUENCIA DE NODOS ( ARCOS ) EN LA LISTA CORRESPON- *
73700 Z      *   A UN RECORRIDO " PREORDEN " EN EL ARBOL, SI LOS APUN- *
73800 Z      *   TABORES DETECTAN UN CICLO, SE HACE CYC=1 .
73900 Z      *
74000 Z      *****
74100 Z
74200 Z
74300 Z      INITIAL
74400 Z
74500      II=IRROOT
74600      IC=0
74700      LISH(1)=IRROOT
74800      CYC=0
74900 Z
75000 Z      FORWARD
75100 Z
75200      90 JJ=PF(II)
75300      IF(JJ.EQ.0)GO TO 10
75400 Z
75500 Z      ADLST
75600 Z
75700      20 IC=IC+1
75800      LISA(IC)=PB(JJ)
75900      LISH(IC+1)=JJ
76000      II=JJ
76100      IF(II.EQ.IRROOT)GO TO 80
76200      GO TO 90
76300 Z
76400      10 IF( II.EQ.IRROOT)GO TO 40
76500 Z
76600 Z      RIGHT
76700 Z
76800      170 JJ=PR(II)
76900      IF(JJ.EQ. 0 ) GO TO 110
77000      GO TO 20
77100 Z
77200 Z      BACK
77300 Z
77400      110 K=PR(II)
77500      IF(K.GT.0)GO TO 150
77600      II=1(-K)
77700      GO TO 160
77800 Z
77900      150 II=0(K)
78000      160 CONTINUE
78100      IF(II.EQ.IRROOT)GO TO 40
78200      GO TO 170
78300 Z
78400      80 CYC=1
78500 Z
78600      40 CONTINUE
78700      RETURN
78800      END
78900
79000
79100
79200
79300

```



```

79400      SUBROUTINE DELTRE (KL)
79500 %
79600      COMMON /A/O(120)/B/T(120)/C/PB(120)/D/PF(120)/F/PR(120)/G/H/N
79700      INTEGER O,T,PB,PF,PR
79800 %
79900 % *****
80000 % *
80100 % * PROPOSITO :
80200 % * QUITAR UN ARCO DEL ARBOL Y ACTUALIZAR SU
80300 % * REPRESENTACION EN TERMINOS DE LOS APUNTAORES DE
80400 % * * TRES ETIQUETAS *
80500 % *
80600 % *****
80700 %
80800      IF( KL.LE.0 )GO TO 20
80900 %
81000 % FORWARD
81100 %
81200      10 IL=O(KL)
81300      JL=T(KL)
81400      GO TO 30
81500 %
81600      20 IL=T(-KL)
81700      JL=O(-KL)
81800      30 IF (PF(IL).NE.JL) GO TO 40
81900      PF(IL)=PR(JL)
82000      GO TO 80
82100 %
82200      40 L=PF(IL)
82300 %
82400 % RIGHT
82500 %
82600      50 CONTINUE
82700      IF (PR(L).NE.JL) GO TO 90
82800      PR(L)=PR(JL)
82900 %
83000 % DELETE
83100 %
83200      60 PR(JL)=0
83300      PR(JL)=0
83400      GO TO 60
83500 %
83600      90 L=PR(L)
83700      GO TO 50
83800 %
83900      60 RETURN
84000      END
84100
84200
84300
84400
84500

```

```

04600      SUBROUTINE ADOTRE(KE)
04700 Z
04800      COMMON /A/O(120)/P/T(120)/C/PB(120)/D/PF(120)/F/PR(120)
04900      COMMON /R/PB(120)/G/H/N
05000      DIMENSION LISA(120),LISN(120)
05100      INTEGER O,T,PB,PF,PR,PD,PDADJ
05200 Z
05300 Z      *****
05400 Z      *
05500 Z      *   PROPOSITO :
05600 Z      *   AGREGAR EL ARCO KE(IE,JE) A UN BOSQUE ,
05700 Z      *   LOS NODOS IE Y JE DEBERAN ESTAR EN DIFERENTES ARBOLES *
05800 Z      *   Y EL NODO JE DEBERA SER RAIZ DE UN ARBOL ,
05900 Z      *
06000 Z      *****
06100 Z
06200 Z
06300 Z      FORWARD
06400 Z
06500      IF(KE) 15,15,20
06600      15 IE=T(-KE)
06700      JE=O(-KE)
06800      GO TO 30
06900 Z
07000      20 IE=O(KE)
07100      JE=T(KE)
07200      30 IF(PF(IE),EQ.0) GO TO 40
07300      PR(JE) = PF(IE)
07400 Z
07500 Z      BACK
07600 Z
07700      40 PF(IE) = JE
07800      FB(JE) = KE
07900 Z
08000 Z      DEPTH
08100 Z
08200      PDADJ = PD(IE) - PD(JE) + 1
08300 Z
08400      CALL ROOT (JE,LISA,LISN,IC,CYC)
08500 Z
08600      DO 50 I = 1,IC-1
08700      PD(LISN(I)) = PD(LISN(I)) + PDADJ
08800 Z
08900      50 CONTINUE
09000      RETURN
09100      END
09200
09300
09400
09500
09600

```

```

09700      SUBROUTINE TRENTE(N)
09800 Z     COMMON /A/O(120)/B/T(120)/C/PB(120)/D/FF(120)/F/PR(120)
09900      COMMON /B/PB(120)
90100      DIMENSION LISA(100),LISH(100)
90200      INTEGER O,I,FB,FF,PR,PD
90300 Z
90400 Z     *****
90500 Z     *
90600 Z     *   PROPOSITO :
90700 Z     *   TENIENDO CONOCIMIENTO DE LOS APUNTADES
90800 Z     *   HACIA ATRAS, CONSTRUIR LA REPRESENTACION DE APUNTADES-
90900 Z     *   RES DE * TRES ETIQUETAS * .
91000 Z     *
91100 Z     *****
91200 Z
91300      DO 80 I = 1,N
91400      FF(I) = 0
91500      PR(I) = 0
91600      PB(I) = 0
91700      80 CONTINUE
91800 Z
91900      DO 10 I =1,N
92000      IF (PB(I) .EQ. 0) GO TO 10
92100 Z
92200      CALL ABOTRE(PB(I))
92300 Z
92400      10 CONTINUE
92500 Z
92600      RETURN
92700      END
92800
92900
93000

```

APENDICE B

ALGORITMO DE RESOLUCION PARA REDES DE FLUJO CON COSTOS CONCAVOS

B.1 ALGORITMO DE DESVIACIONES PARA RESOLVER REDES DE FLUJO CON COSTOS LINEALES.

En esta sección se presenta un método para resolver problemas de redes de flujo con costo mínimo, al cual se le conoce en español como algoritmo de "desviaciones", debido a que el proceso disminuye las desviaciones que existen entre un flujo actual dado y el flujo óptimo.

Este algoritmo no requiere factibilidad inicial primal ni dual, consecuentemente, es un algoritmo primal-dual que puede comenzar con una solución que solo debe satisfacer la conservación de flujo. Esto proporciona flexibilidad para comenzar la solución en contraste con los algoritmos primal o dual.

Es posible usar un árbol base para hacer más eficiente el algoritmo. Esta idea se sigue en el algoritmo presentado aquí.

B.1.1 Red Modelo

La red modelo que usa el algoritmo de desviaciones es una red circular, es decir, todos los flujos externos de nodo se deben representar por arcos. En la figura B.1a se presenta una red

con sus parámetros de arco y nodo y en la figura B.1b se muestra la red circulatoria equivalente.

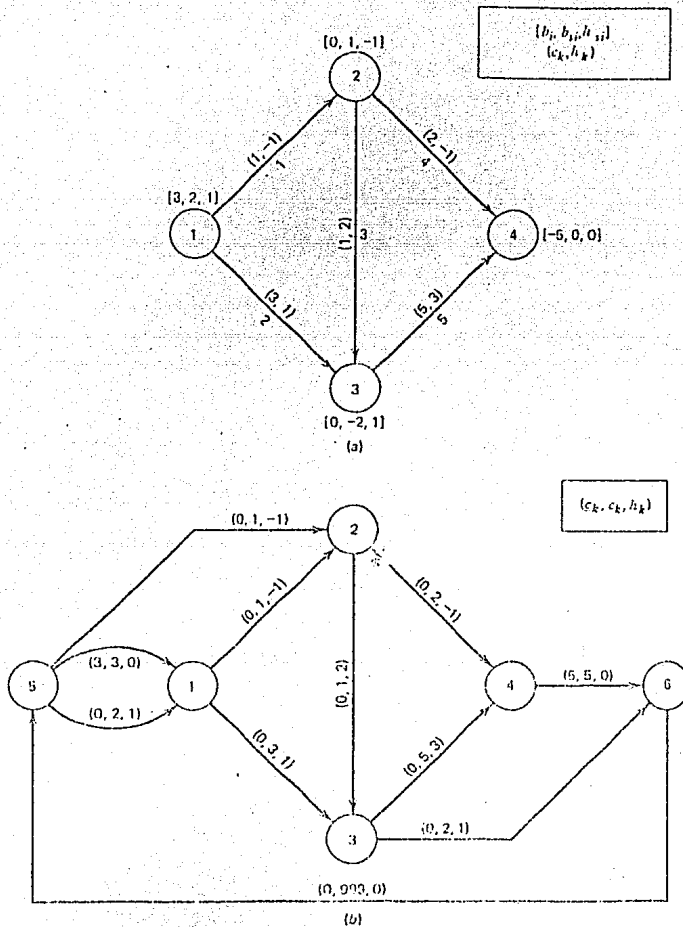


Figura B.1

Ejemplo de Red Modelo para el Algoritmo de Desviaciones
 (a) Red Original (b) Red Circulatoria Equivalente

La red circulatoria se obtiene definiendo un nodo fuente, nodo 5, y un nodo sumidero, nodo 6. Todos los flujos externos positivos de un nodo i generan un arco del nodo fuente al nodo i . Para el flujo externo fijo, las capacidades mínima y máxima del arco generado son iguales al valor del flujo fijo y el costo igual a cero. Para la holgura del flujo, la capacidad mínima es cero y la capacidad máxima es igual a la cota del flujo de holgura. El costo del arco se iguala al costo del flujo de holgura. De manera similar, los flujos externos negativos se representan con arcos que terminan en el nodo sumidero.

Por último, se genera un "arco de retorno" del nodo sumidero al fuente con capacidad inferior cero, capacidad máxima muy grande, y costo cero. La finalidad de este arco es proporcionar la conservación de flujo en los nodos fuente y sumidero. Puesto que se ha eliminado el flujo externo en la red, la cantidad de flujo que sale del nodo fuente debe ser igual a la que entra al nodo sumidero y la misma que pasa a través del arco de retorno.

La subrutina READOK lee los datos de la red tal como se describió en la sección 3.7 y convierte la red original al formato descrito arriba. Note que se generan tantos arcos artificiales como flujos externos haya en la red. Puesto que la implementación clásica del algoritmo de desviaciones no contempla una enumeración ordenada de los arcos, no se requiere la subrutina ORIGS.

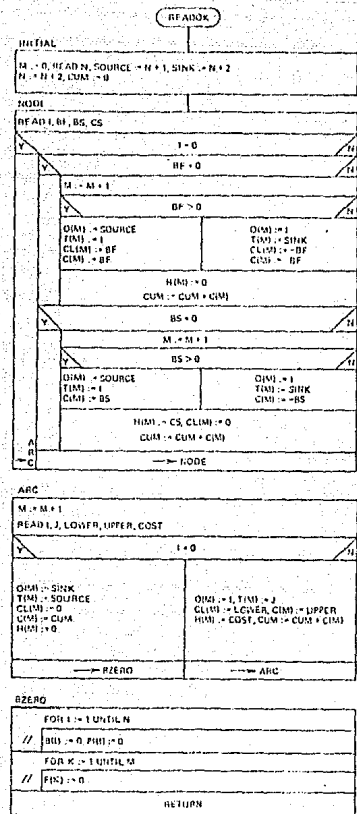
ALGORITMO READOK

PROPOSITO: Leer y guardar los datos correspondientes a nodos y arcos de la red y crear la representación (red circular) que usa el Algoritmo de Desviaciones (OUT-OF-KILTER ALGORITHM).

1. INITIAL. Se inicializa el número de arcos en cero, $M:=0$. Se lee el número de nodos (N) y se definen los nodos fuente (SOURCE) y sumidero (SINK).
2. NODO. Leer el conjunto de datos correspondientes al nodo I . Tales datos son: el flujo externo fijo BF , la cota superior del flujo externo de holgura BS , y el costo CS para el flujo externo de holgura. Si el conjunto de datos está en blanco ($I=0$), ir al paso 3. Si el flujo fijo es cero, ir a paso 2a.

Por el contrario determine el signo del flujo. Si es positivo, genere un arco del nodo fuente al nodo I con capacidades mínima y máxima iguales a BF y costo igual a cero. Si es negativo, genere un arco del nodo I al nodo sumidero con los mismos parámetros del caso positivo.

- 2a. Si el flujo de holgura es cero, repetir el paso 2. Por el contrario, determine su signo. Si es positivo, genere un arco del nodo fuente al nodo I , con capacidad mínima cero, capacidad máxima igual al flujo de holgura y costo igual al costo del flujo de holgura. Si es negativo, genere un -



arco del nodo al nodo sumidero con los mismos parámetros del caso positivo. Repetir el paso 2.

3. ARC. Leer el conjunto de datos correspondientes al arco (I,J). Tales datos son: capacidad mínima (LOWER), capacidad máxima (UPPER), y el costo por unidad de flujo (COST). Si el conjunto de datos está en blanco (I=0), se ha completado la lectura de datos; crear el arco de retorno (un arco del nodo sumidero al fuente) e ir al paso 4. En caso contrario, incrementar el número de arcos en uno y guardar la información del arco. Repetir el paso 3.

4. BZERO. Hacer cero todos los flujos externos fijos [R(I):=0], los potenciales de nodo [PI(I):=0], y todos los flujos [F(K):=0]. Regresar.

B.1.2 Modelo de Programación Lineal

Supuestos todos los flujos externos igual a cero y que se debe satisfacer la conservación del flujo en cada nodo, la formulación de desviaciones como modelo de programación Lineal tiene la forma siguiente:

$$\begin{aligned} \text{Min. } & \sum_{k=1}^m h_k f_k \quad \text{o} \quad \text{Max. } \sum_{k=1}^m -h_k f_k \\ \text{s. a. } & \sum_{k \in \text{MT}_i} f_k - \sum_{k \in \text{MO}_i} f_k = 0 \quad i = 1, \dots, n \\ & f_k \geq \underline{c}_k \quad k = 1, \dots, m \\ & f_k \leq \bar{c}_k \quad k = 1, \dots, m \end{aligned}$$

Se recuerda de la sección 1.6 que una de las restricciones de conservación de flujo es redundante. Note que en la formulación anterior no se restringen los signos de \underline{c}_k o \bar{c}_k , sin embargo se supondrá que $0 \leq \underline{c}_k \leq \bar{c}_k$.

Para formular el dual de este MPL se asocia Π_i con la restricción de conservación del flujo para el nodo i , ω_k con la restricción $f_k \geq \underline{c}_k$ (la cual se maneja como $-f_k \leq -\underline{c}_k$ para el propósito de tomar el dual), y δ_k con la restricción $f_k \leq \bar{c}_k$. Entonces, el dual de la formulación del algoritmo de desviaciones está dado por:

$$\begin{aligned} \text{Min. } & \sum_{k=1}^m \delta_k \bar{c}_k - \sum_{k=1}^m \omega_k \underline{c}_k \\ \text{s. a. } & \Pi_i - \Pi_j + \delta_k - \omega_k \geq -h_k \quad \text{para el arco } k(i,j) \in M \end{aligned}$$

$$\begin{aligned} \Pi_i & \text{ Libre } \quad i \in N \\ \omega_k & \geq 0 \quad k \in M \\ \delta_k & \geq 0 \quad k \in M \end{aligned}$$

Puesto que una de las restricciones del primal asociadas con Π es redundante, también una de las variables duales Π es redundante.

Las condiciones de optimalidad se derivan en forma similar al Capítulo 1. Por conveniencia se define

$$d_k = \Pi_i - \Pi_j + h_k$$

Dada una solución F al problema primal y una solución dual parcial Π ; entonces, se tiene que las dos soluciones son óptimas para sus respectivos problemas si se dan las siguientes condiciones.

1. Factibilidad primal.

(a) Conservación del flujo en cada nodo

(b) $f_k \geq \underline{c}_k$ para todo arco k

(c) $f_k \leq \bar{c}_k$ para todo arco k

2. Factibilidad Dual Restringida

(a) $\omega_k = \text{Max.}\{0, d_k\}$

(b) $\delta_k = \text{Max.}\{0, -d_k\}$

3. Holgura complementaria

(a) $d_k = 0$ para $0 < f_k < \bar{c}_k$

(b) $f_k = \underline{c}_k$ para $d_k > 0$

(c) $f_k = \bar{c}_k$ para $d_k < 0$

Antes se mencionó que el algoritmo de desviaciones requiere un flujo inicial que satisfaga la condición 1(a), pero no necesariamente la condición 1(b) o 1(c). Los potenciales de nodo (variables duales Π) iniciales son arbitrarios, de manera que la condición 3 difícilmente se satisface. La condición 2 no se usa en el algoritmo pero es muy útil para el análisis de sensibilidad posterior a la optimalidad, el cual, no se estudia aquí.

El algoritmo procede variando los flujos y los potenciales en forma tal que todos los arcos se muevan hacia las condicio-

nes 1(b), 1(c) y 3. Se garantiza una solución óptima en un número finito de iteraciones si existe una solución factible. Si no hay solución factible, el algoritmo detecta e indica esta situación.

B.1.3 Estados Conformables ("KILTER STATES")

Una vez que se hayan definido los potenciales para cada nodo i (π_i) y los flujos para cada arco $k(i,j)$, es posible determinar el valor de d_k para cada arco. En el Cuadro B.1 se definen los posibles estados en que un arco se puede encontrar. Note que un arco es conformable (sin desviación) si $\underline{c}_k \leq f_k \leq c_k$ y las condiciones de optimalidad 3. se satisfacen. Si el arco no satisface las condiciones de optimalidad se dice que está en un estado no conformable (con desviación). En la Figura B.2 se vuelve a resumir estos estados conformables, pero ahora esquemáticamente para facilitar su comprensión. Conforme se cambia el flujo en el arco este se mueve, hacia arriba o hacia abajo, sobre una columna específica dependiendo de que f aumente o disminuya. Conforme se cambian los π del arco (o equivalentemente d_k), el arco se mueve hacia la izquierda o hacia la derecha a lo largo de un renglón.

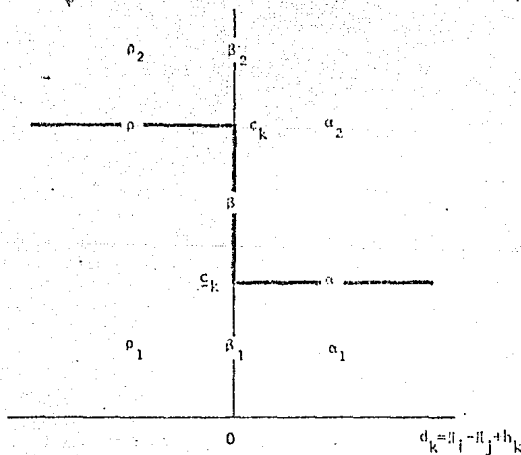
Cuadro B.1 ESTADOS CONFORMABLES

Estado	d	f	¿Conformable?	¿Porqué?
α	$d > 0$	$f = \underline{c}$	si	satisface 3(b)
β	$d = 0$	$\underline{c} \leq f \leq c$	si	satisface 3(a)
ρ	$d < 0$	$f = c$	si	satisface 3(c)
α_1	$d > 0$	$f < \underline{c}$	no	viola 1(b), 3(b)
β_1	$d = 0$	$f < \underline{c}$	no	viola 1(b)
ρ_1	$d < 0$	$f < c$	no	viola 3(c)
α_2	$d > 0$	$f > \underline{c}$	no	viola 3(b)
β_2	$d = 0$	$f > c$	no	viola 1(c)
ρ_2	$d < 0$	$f > c$	no	viola 1(c), 3(c)

Cada celda de la Figura B.2a corresponde a una subregión particular en la Figura B.2b

	$d < 0$	$d = 0$	$d > 0$
$f > c$	$\rho_1 \neq 1(c), 3(c)$	$\beta_2 \neq 1(c)$	$\alpha_2 \neq 3(b)$
$f = c$	ρ		
$\varepsilon < f < c$	$\rho_1 \neq 3(c)$	β_1	α
$f = \varepsilon$			
$f < \varepsilon$		$\beta_1 \neq 1(b)$	$\alpha_1 \neq 1(b), 3(b)$

(a)



(b)

Figura B.2

Representación de los Estados Conformables ("Kilter States")

(a) Pictórica (b) gráfica

Como ejemplo, considere las soluciones factibles de la Figura B.3. En cada solución además de los parámetros se indica el estado en que se encuentra el arco. El lector puede verificar cada estado fácilmente. La solución de la Figura B.3a no puede ser óptima debido a que los arcos 3 ($d_3 = -4$, $f_3 = 0 \Rightarrow \rho_1$) y 8 ($d_8 = 2$, $f_8 = 2 \Rightarrow \alpha_2$) son no conformables. La solución de la Figura B.3b es óptima puesto que todos los arcos son conformables.

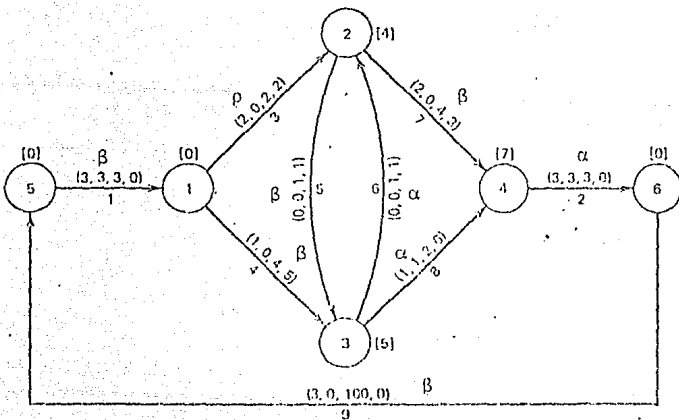
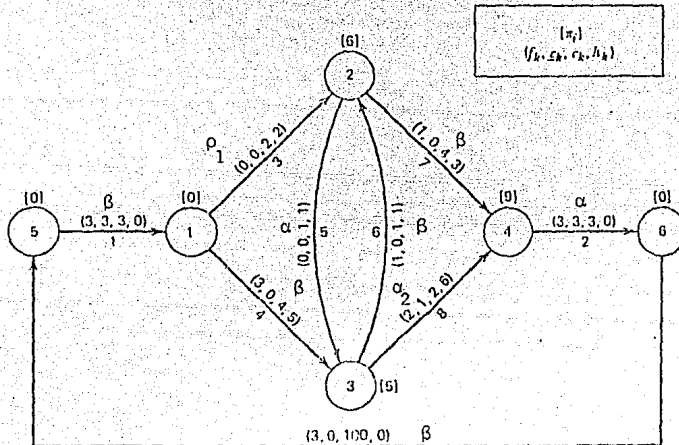


Figura B.3

Dos Soluciones para una Red Ejemplo

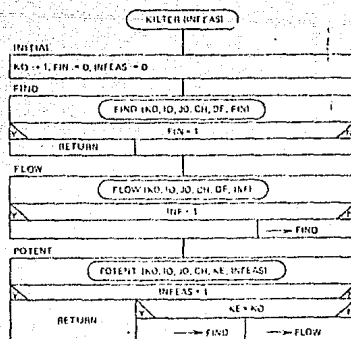
Ahora se describe una serie de procedimientos que producirán el óptimo o una indicación de no factibilidad. El algoritmo global es KILTER y tiene tres partes principales:

1. Encontrar un arco no conformable
2. Cambiar los flujos en la red (manteniendo constantes los potenciales) para mover al arco a un estado conformable.
3. Si el cambio de flujo no tiene éxito, se mantienen constantes los flujos y se cambian los potenciales de nodo para hacer conformable el arco o permitir más cambio de flujo.

ALGORITMO KILTER

PROPOSITO: Dado un flujo inicial que -
satisface la conservación de flujo y -
potenciales arbitrarios en los nodos:
este algoritmo modifica los flujos y -
potenciales en los nodos hasta obtener
una solución óptima. Si no hay solu-
ción factible, se regresa INFEAS = 1.

1. INITIAL. Asignar a k_0 el valor 1 y -
a los indicadores de solución óptima
(FIN) y no factibilidad un cero.
2. FIND. Determine si el arco k_0 es con-
formable. Si no, regresar los nodos-
origen (i_0) y terminal (j_0) de k_0 , -
la dirección del cambio de flujo (CH
= 1 so es un incremento, CH = -1 si
es dismunición), y la cantidad de -
flujo a cambiar necesaria para hacer
el arco k_0 conformable; ir al paso 3.
Si k_0 es conformable, encontrar un -
arco no conformable y regresar la in-
formación mencionada. Si todos los -
arcos son conformables (FIN=1), re-
gresar con la solución óptima.
3. FLOW. Manteniendo constantes los po-
tenciales de nodo, tratar de cambiar
los flujos en la red para mover al -
arco k_0 hacia un estado conformable.
Si esto es posible (INF=0) regresar
al paso 2. Si es imposible (INF=1),
ir al paso 4.
4. POTENT. Manteniendo constantes los -
flujos, tratar de cambiar los poten-
ciales en los nodos para mover al ar-
co k_0 hacia un estado conformable -



o permitir más cambios de flujo. Si se cambian los potenciales y el arco k_0 se convierte en conformable, regresar al paso 2. Por el contrario, regresar al paso 3. Si no se puede cambiar los potenciales, no hay solución factible al problema de flujo original (INFEAS = 1). Regresar.

B.1.4 Cambio de Flujo

Primero se estudia un procedimiento que intenta mover a un arco de un estado no conformable hacia otro conformable, cambiando los flujos en la red mientras se mantienen constantes los potenciales de nodo. El algoritmo de Desviaciones supone que hay conservación de flujo en cada nodo. Si la solución inicial satisface esta condición también lo harán las soluciones subsiguientes debido a que el flujo siempre se cambia en un ciclo.

Por ejemplo, considere el ciclo $M_c = \{-3, -6, -4\}$ de la Figura B.3a. Un cambio de flujo de una unidad en este ciclo produce los nuevos flujos $f_3 = 1$, $f_4 = 2$ y $f_6 = 0$, mientras que los demás flujos siguen igual. Verifique que esta nueva solución satisface la conservación de flujo.

Para efecto de identificar un ciclo en el cual se pueda cambiar el flujo se tienen en cuenta las dos reglas que siguen. Estas reglas aseguran la convergencia finita del algoritmo.

- (a) El cambio de flujo no hace que un arco conformable se convierta en uno no conformable.
- (b) El cambio de flujo no hace que un arco se desvie más.

Para ejemplificar el último caso, no es deseable incrementar el flujo en un arco con $f_k > c_k$ puesto que el arco se desviará aún más.

Las reglas anteriores conducen a los cambios de flujo especificados en el Cuadro B.2, los cuales se verifican fácilmente usando como referencia la Figura B.2. Obsérvese que solo los estados α y ρ no permiten cambio. En el estado β , el cambio de flujo puede ser positivo o negativo dependiendo del flujo en el arco.

El cuadro B.2 conduce a una nueva función de admisibilidad que especifica cuando un arco se puede incluir en un ciclo sobre el que se cambiará el flujo.

Cuadro B.2 CAMBIOS DE FLUJO PERMITIDOS EN ARCOS

Estado	d^*	f	Cambio de Flujo Permitido	Máxima Cantidad de Cambio	Nuevo Estado
α	$d > 0$	$f = \underline{c}$	ninguno	0	α
β	$d = 0$	$f < \underline{c}$	incremento	$c-f$	β
		$f > \underline{c}$	decremento	$f-\underline{c}$	β
ρ	$d < 0$	$f = \underline{c}$	ninguno	0	ρ
α_1	$d > 0$	$f < \underline{c}$	incremento	$\underline{c}-f$	α
β_1	$d = 0$	$f < \underline{c}$	incremento	$c-f$	β
ρ_1	$d < 0$	$f < \underline{c}$	incremento	$c-f$	ρ
α_2	$d > 0$	$f > \underline{c}$	decremento	$f-\underline{c}$	α
β_2	$d = 0$	$f > \underline{c}$	decremento	$f-\underline{c}$	β
ρ_2	$d < 0$	$f > \underline{c}$	decremento	$f-\underline{c}$	ρ

* $d = \pi_i - \pi_j + h_k$ para el arco $k(i,j)$

Un arco hacia adelante k es admisible si es posible incrementar su flujo

$$A_d(k) = 1 \text{ si } \begin{cases} f_k < \underline{c}_k & \text{o} \\ d_k \leq 0 \text{ y } f_k < c_k \end{cases}$$

$$A_d(k) = 0 \quad \text{en otro caso.}$$

Un arco reflejado $-k$ es admisible si es posible disminuir su flujo

$$A_d(-k) = 1 \text{ si } \begin{cases} f_k > c_k & \text{o} \\ d_k \geq 0 \text{ y } f_k > \underline{c}_k \end{cases}$$

$$A_d(-k) = 0 \quad \text{en otro caso.}$$

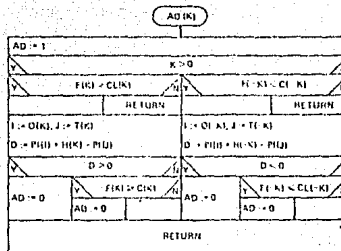
Esta nueva función de admisibilidad se implementa en la función (tipo FORTRAN) ADOKA.

FUNCION ADOKA

PROPOSITO: Determinar la admisibilidad de un arco para el Algoritmo de Desviaciones.

Si el flujo es menor que la capacidad mínima, el arco hacia adelante k es admisible. Si el flujo es mayor que la capacidad, el arco reflejado $-k$ es admisible. Si ninguna de las condiciones anteriores se cumple, -- calcular el valor de d_k para el arco. Para un arco hacia adelante k si $d_k \leq 0$ y $f_k < c_k$ el arco es admisible. -- Por el contrario, no es admisible. -- Para un arco reflejado $-k$, si $d_k \geq 0$ y $f_k > c_k$, el arco es admisible. Por el contrario, no es admisible.

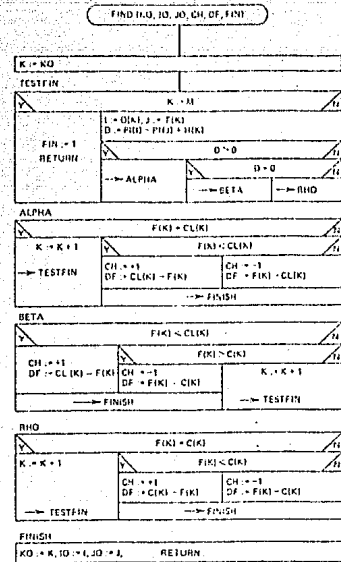
Para encontrar un ciclo sobre el que cambiará el flujo -- se debe encontrar un arco con desviación. El procedimiento implementado en el algoritmo FIND comienza con el arco 1 y sigue arco por arco (en orden creciente) hasta encontrar un arco con desviación o chequear todos. Si no se encuentra un arco con -- desviación, los flujos actuales son óptimos.



ALGORITMO FIND

PROPOSITO: Encontrar un arco K_0 no - formable (con desviación), su nodo - origen i_0 , y su nodo terminal j_0 . La dirección del cambio de flujo se indica por CH (+1 para incremento, -1 para decremento). La cantidad de flujo a circular para que k_0 sea conformable es DF.

1. TESTFIN. Determine si todos los arcos han sido chequeados. Si es así, regrese con FIN=1. Si no, encuentre los nodos origen y terminal -- del siguiente arco y calcule d_k . - Si $d_k > 0$, ir al paso 2. Si $d_k = 0$, ir al paso 3. Si $d_k < 0$, ir al paso 4.
2. ALPHA. El arco está en el estado α . Si $f_k = c_k$, el arco es conformable; - ir al paso 1. Si $f_k < c_k$, el arco está en el estado α_1 . Si $f_k > c_k$ el arco está en α_2 . Indicar el cambio - de flujo apropiado de acuerdo al - estado encontrado y regresar.
3. BETA. El arco está en el estado β . Si $f_k \leq c_k$ el arco está en β_1 . Si $f_k > c_k$ el arco está en β_2 . Indicar el cambio de flujo apropiado y regresar. En caso contrario, el arco es conformable; ir al paso 1.
4. RHO. El arco está en el estado ρ . - Si $f_k = c_k$ el arco es conformable. - Ir al paso 1. Si $f_k < c_k$ el arco está en ρ_1 . Si $f_k > c_k$ el arco está en ρ_2 . Indicar el cambio de flujo -- apropiado y regresar.



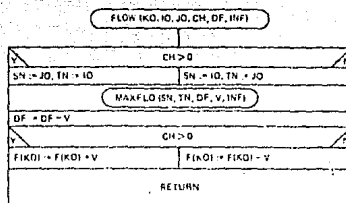
Una vez que se describe un arco $k_0(i_0, j_0)$ no conformable lo primero que se intenta es cambiar los flujos en la red para convertirlo en arco conformable. Esto se hace en el algoritmo FLOW. Si el flujo se ha de incrementar por una cantidad DF, se debe tratar de pasar el flujo DF del nodo j_0 al i_0 a través de la red usando solo arcos admisibles. Si esto es posible, el arco k_0 se puede convertir en conformable sin producir la desviación de otros o aumentarla. La principal herramienta de FLOW es el algoritmo MAXFLO, en el cual se ejecutan todas las operaciones necesarias para encontrar trayectorias de arcos admisibles y circular flujos.

ALGORITMO FLOW

PROPOSITO: Cambiar los flujos en la red de manera que el flujo en el arco $k_0(i_0, j_0)$ pueda incrementarse o disminuirse, y a la vez mantener la conservación de flujo en cada nodo.

Si el flujo se ha de incrementar en el arco k_0 por una cantidad DF, el algoritmo llama a MAXFLO para intentar pasar el flujo DF del nodo j_0 al i_0 a través de la red usando solo arcos admisibles. Si el flujo máximo que se puede pasar V es menor que DF se indica haciendo INF:=1 en MAXFLO. Se incrementa el flujo en el arco k_0 por V.

Si el flujo se ha de disminuir en el arco k_0 por DF, se llama MAXFLO para intentar pasar el flujo DF del nodo i_0 al j_0 a través de red usando solo arcos admisibles. Si el flujo-

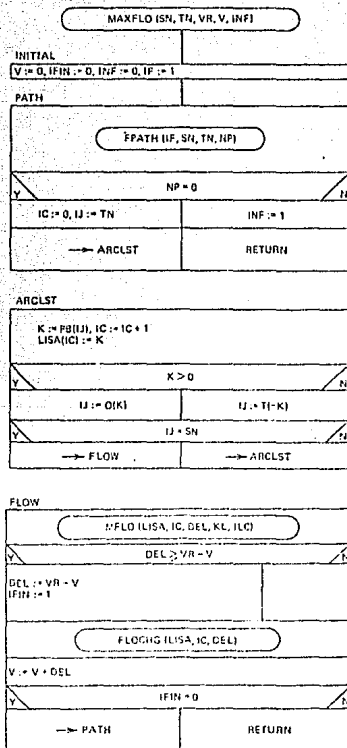


máximo que se puede pasar V es menor que DF , se indica haciendo $INF:=1$. - Se disminuye el flujo en el arco k por V .

ALGORITMO MAXFLO

PROPOSITO: Determina la red que consiste de arcos admisibles para la cual se obtiene un flujo dado (DF) del nodo fuente (SN) al nodo sumidero (TN). Si no se puede obtener el flujo dado se proporciona el flujo máximo (V) de ST a TN . Este procedimiento supone que se tiene de antemano un flujo factible inicial.

1. INITIAL. Inicializar el flujo en cero, $V:=0$.
2. PATH. Encuentra una trayectoria del nodo SN a TN que consiste de arcos admisibles, es decir, arcos hacia adelante con flujo menor que su capacidad, o arcos reflejados con flujo mayor que su cota inferior sobre su correspondiente arco hacia adelante. Si no se encuentra ninguna trayectoria se termina con el flujo máximo flujo de SN a TN . En caso contrario ir al paso 3.
3. ARCLST. Forma la lista de arcos de la trayectoria de SN a TN usando los apuntadores hacia atrás.
4. FLOW. Encuentra el flujo máximo (DEL) que puede ser enviado a través de la trayectoria. Si DEL excede a $VR-V$, incrementa el flujo en



VR-V y termina. Por el contrario, incrementa el flujo en la trayectoria y regresa al paso 2.

La principal subrutina de MAXFLO es FPATH3 la cual es usada para encontrar una trayectoria de solo arcos admisibles del nodo s al nodo t. Esta subrutina usa una técnica de etiquetación para encontrar la trayectoria deseada y no depende de un orden específico en la numeración de los arcos. Observe que si el flujo en el arco (i_0, j_0) se va a incrementar

$$s = j_0 \quad \text{y} \quad t = i_0$$

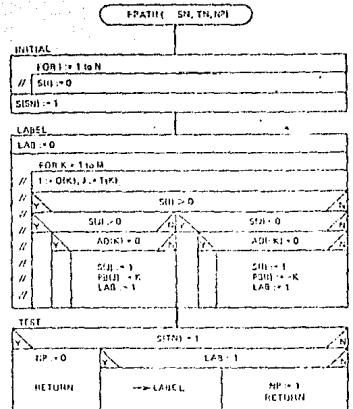
Si el flujo en el arco (i_0, j_0) se va a disminuir

$$s = i_0 \quad \text{y} \quad t = j_0$$

ALGORITMO FPATH3

PROPOSITO: Encontrar una trayectoria empleando un procedimiento de etiquetación. Las etiquetas se indican en la lista de nodos S.

1. INITIAL. Inicializar en cero todas las etiquetas excepto la del nodo s, $S(SN):=1$.
2. LABEL. Ir a través de todos los arcos de la red. Si se encuentra un arco $k(i,j)$ admisible tal que el nodo i está etiquetado pero el nodo j no, etiquetar el nodo j. Asignar al apuntador hacia atrás del nodo j el arco k.



Si el arco reflejado $-k(j,i)$ es admisible y el nodo j está etiquetado pero no el nodo i , etiquetar el nodo i . Asignar al apuntador hacia atrás del nodo i el arco $-k$.

3. TEST. Si el nodo t ha sido etiquetado, entonces se ha encontrado -- una trayectoria, la cual se puede trazar con apuntadores hacia atrás. Si el nodo t no ha sido etiquetado pero algún nodo se etiquetó en el paso 2, repetir el paso 2. Si el -- nodo t no ha sido etiquetado y nin -- gún nodo fue etiquetado en el paso 2, no hay trayectoria de arcos adm -- isibles de s a t . Los nodos eti -- quetados son todos aquellos que -- pueden ser alcanzados desde el no -- do s a través de arcos admisibles.

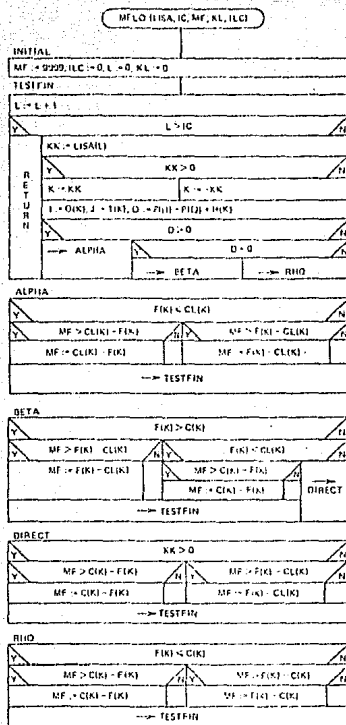
Ahora se proporciona el algoritmo MFLO que determina el máximo cambio de flujo en la trayectoria determinada en -- FPATH3. Este algoritmo hace uso de los resultados del Cuadro -- B.2 para obtener el máximo cambio de flujo en cada arco de la -- trayectoria.

Cuando FPATH3 no encuentra una trayectoria, significa -- que se ha encontrado el máximo flujo circulable entre los nodos s y t y que este flujo no es suficiente para conducir a k_0 a un estado conformable. El conjunto de nodos etiquetados $[s(I)=1]$ -- juega un importante papel en la parte que sigue del algoritmo.

ALGORITMO MFLO

PROPOSITO: Determinar el máximo cambio de flujo posible en la trayectoria encontrada en FPATH3.

1. INITIAL. Asignar al máximo flujo inicial un número grande y al índice de arcos el valor cero.
2. TESTFIN. Ir a través de la lista de arcos LISA. Para cada arco, determine su estado conformable e ir al paso asociado del logaritmo.
3. ALPHA. $d_k > 0$. Si $f_k \leq c_k$, determine si $MF > c_k - f_k$. Si es así, hacer $MF := c_k - f_k$. Si $f_k > c_k$, determine si $MF > f_k - c_k$. Si es así, hacer $MF := f_k - c_k$. Ir al paso 2.
4. BETA. $d_k = 0$. Si $f_k > c_k$, determine si $MF > f_k - c_k$. Si es así, hacer $MF := f_k - c_k$, e ir al paso 2. Si $f_k \leq c_k$, determine si $f_k < c_k$. Si es así, determine si $MF > c_k - f_k$. Si es así, hacer $MF := c_k - f_k$ e ir al paso 2. Si $c_k \geq f_k \geq c_k$, ir al paso 5.
5. DIRECT. Si el arco es hacia adelante, determine si $MF > c_k - f_k$. Si es así, hacer $MF := c_k - f_k$ e ir al paso 2. Si el arco es reflejado, determine si $MF > f_k - c_k$. Si es así, hacer $MF := f_k - c_k$ e ir al paso 2.
6. RHO. $d_k < 0$. Si $f_k \leq c_k$, determine si $MF > c_k - f_k$. Si es así, hacer $MF := c_k - f_k$ e ir al paso 2.



B.1.5 Cambio De Potencial

Quando la cantidad de flujo cambiada por FLOW no es suficiente para conducir al arco $k_0(i_0, j_0)$ a un estado conformable, se procede a cambiar los potenciales de nodo de manera que el arco k_0 se convierta en conformable o se produzcan más arcos admisibles. Al fallar FLOW se identifican dos conjuntos de nodos: N_1 es el conjunto de nodos etiquetados en la búsqueda de un ciclo, y N_2 es el conjunto de nodos no etiquetados. Esta información la proporciona el algoritmo MAXFLOW por medio de la lista S, en donde

$$i \in N_1 \quad \text{si } S(I) = 1$$

$$i \in N_2 \quad \text{si } S(I) = 2$$

Debe suceder que s pertenezca a N_1 y t a N_2 . Se recuerda que si el flujo se incrementa en el arco k_0 , entonces $s=j_0$ y $t=i_0$. Alternativamente, si el flujo se disminuye en k_0 , $s=i_0$ y $t=j_0$. La Figura B.4 ilustra esquemáticamente el caso en que el flujo se incrementa en el arco k_0 .

Al fallar FLOW se conocen ciertas características en los arcos que se originan en N_1 y terminan en N_2 ó, viceversa. Si el arco $k_1(i_1, j_1)$ se origina en N_1 y termina en N_2 , se sabe que el arco no puede estar en un estado conformable que permitiera un incremento de flujo pues, por el contrario, k_1 sería admisible y, por tanto, j_1 etiquetado y miembro de N_1 . Con esto en mente y referenciando al Cuadro B.2 o Figura B.2 se obtiene que el arco k_1 debe estar en uno de los estados listados en el Cuadro B.3. Observe en este cuadro que si el flujo es menor que la cota superior del arco entonces es posible hacer a k_1 admisible incrementando π_j ó, equivalentemente, disminuyendo d.

Si el arco $k_2(i_2, j_2)$ se origina en N_2 y termina en N_1 , se sabe que el arco no puede estar en un estado conformable que permitiera una disminución de flujo pues, por el contrario, $-k_2$ sería admisible y i_2 etiquetado. Así, k_2 debe estar en uno de los estados del Cuadro B.4. Para los estados de este cuadro si el flujo excede la cota inferior del arco entonces es posible hacer al arco k_2 admisible mediante el incremento de π_i , lo cual incrementa d.

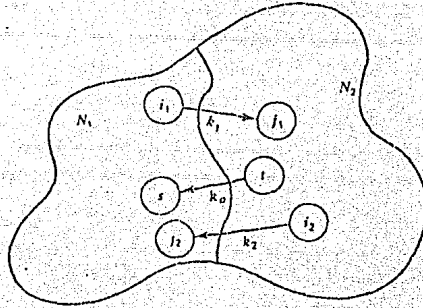


Figura B.4
Situación en que el Algoritmo FLOW Falla.
El Flujo se Incrementa en el Arco k_0 .

Para cambiar los potenciales de nodo se agrega la cantidad Δ_p a todos los potenciales de los nodos en N_2 . La cantidad Δ_p se escoge de tal manera que el arco k_0 se haga conformable o al menos un arco, como k_1 o k_2 , se convierta en admisible. Además, no se permiten cambios que conduzcan un arco de un estado conformable a otro no conformable. Los casos en que es posible cambiar los potenciales de nodo para conducir a k_0 a un estado conformable se presentan en el Cuadro B.5. Si el flujo en k_0 no está en el rango $[c, c]$ no hay cambio posible que haga a k_0 conformable. Δ_p se escoge como el menor de los valores necesarios para hacer conformable a k_0 o convertir un arco en admisible. La violación de esta regla permite que un arco se mueva de un estado conformable a otro no conformable.

El algoritmo POTENT determina el valor de Δ_p y lo agrega a los potenciales de los nodos en N_2 . Si no hay manera de conducir a k_0 a un estado conformable o crear otro arco admisible, no hay solución al problema de flujo a costo mínimo.

Cuadro B.3 POSIBLES ESTADOS DEL ARCO $k(i,j)$ CUANDO $i \in N_1$ y $j \in N_2$

Estado	d^*	f	Incremento en π_j para Hacer a k Admisible	Nuevo Estado
α	$d > 0$	$f = c$	$\Delta_p = d$	β
ρ	$d = 0$	$f = c$	-	ρ
α_2	$d < 0$	$f = c$	-	ρ
	$d > 0$	$c < f < c$	$\Delta_p = d$	β
		$f \geq c$		α_2, β_2 o ρ_2
β_2	$d = 0$	$f > c$		ρ_2
β_2	$d < 0$	$f > c$		ρ_2

$$*d = \pi_i - \pi_j + h_k$$

Cuadro B.4 POSIBLES ESTADOS DEL ARCO $k(i,j)$ CUANDO $i \in N_2$ y $j \in N_1$

Estado	d^*	f	Incremento en π_j para Hacer $-k$ Admisible	Nuevo Estado
α	$d > 0$	$f = c$	-	α
β	$d = 0$	$f = c$	-	α
ρ	$d < 0$	$f = c$	$\Delta_p = -d$	β
α_1	$d > 0$	$f < c$	-	α_1
β_1	$d = 0$	$f < c$	-	α_1
ρ_1	$d < 0$	$f \leq c$	-	ρ_1, β_1 o α_1
		$c < f < c$	$\Delta_p = -d$	β_1

$$*d = \pi_i - \pi_j + h_k$$

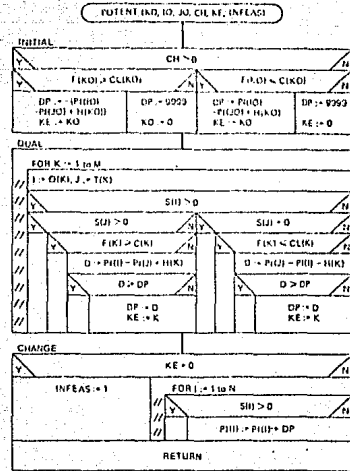
Cuadro B.5 CAMBIOS DE POTENCIAL PARA CONDUCIR AL ARCO k_0 A UN ESTADO CONFORMABLE

Estado	d_0	f_0	Cambio Requerido
α_1	$d = 0$	$f < c$	-
β_1	$d = 0$	$f < c$	-
ρ_1	$d < 0$	$f < c$	-
		$c \leq f < c$	Incrementar π_i por $-d$
α_2	$d > 0$	$c < f \leq c$	Incrementar π_j por d
		$f < c$	
β_2	$d = 0$	$f > c$	-
ρ_2	$d < 0$	$f > c$	-

ALGORITMO POTENT

PROPOSITO: Determinar el valor del incremento en los potenciales de los nodos que pertenecen a N_2 , para conducir al arco k_0 a un estado conformable ("in-Kilter state) o convertir admisible uno de los arcos que se origina en N_1 y termina en N_2 o, viceversa.

1. INITIAL. Determinar si un cambio de potencial puede hacer conformable el arco k_0 . Si es así, asignar a Δ_p el valor correspondiente y hacer $k_E := k_0$. Si no, asigne a Δ_p un valor grande y hacer $k_E := 0$.
2. DUAL. Ir a través de la lista de arcos. Para cada arco determine el arco tiene un nodo terminal en N_1 y otro en N_2 . Si es así, determine el cambio de potencial que podría hacer admisible al arco. Si este cambio es menor que Δ_p , actualice Δ_p y k_E . Cuando todos los arcos se han considerado, ir al paso 3.
3. CHANGE. Si k_0 no se puede hacer conformable y ningún arco se puede hacer admisible mediante un cambio de potencial ($KE=0$), indique no factibilidad ($INFEAS:=1$) y regrese. Si se ha determinado un cambio de potencial, incrementar por Δ_p los potenciales de los nodos en N_2 y regresar.



B.2 ALGORITMO NTCNCV

NTCNCV es el nombre del algoritmo global que resuelve el problema de redes de flujo con costo mínimo cóncavo. Está constituido por un programa principal y 17 subrutinas, las cuales se relacionan según se ilustra en la Figura B.5. Todas las lecturas, la impresión de la representación de la red, de la solución factible inicial y de la solución óptima se efectúan en el programa principal. La principal herramienta de NTCNCV es la subrutina ENUMER. En esta se implementa el procedimiento de enumeración implícita del Capítulo 3. Además de la subrutina READOK que lee la información para construir la representación de la red, se requiere la subrutina READCC que lee los datos para el problema cóncavo.

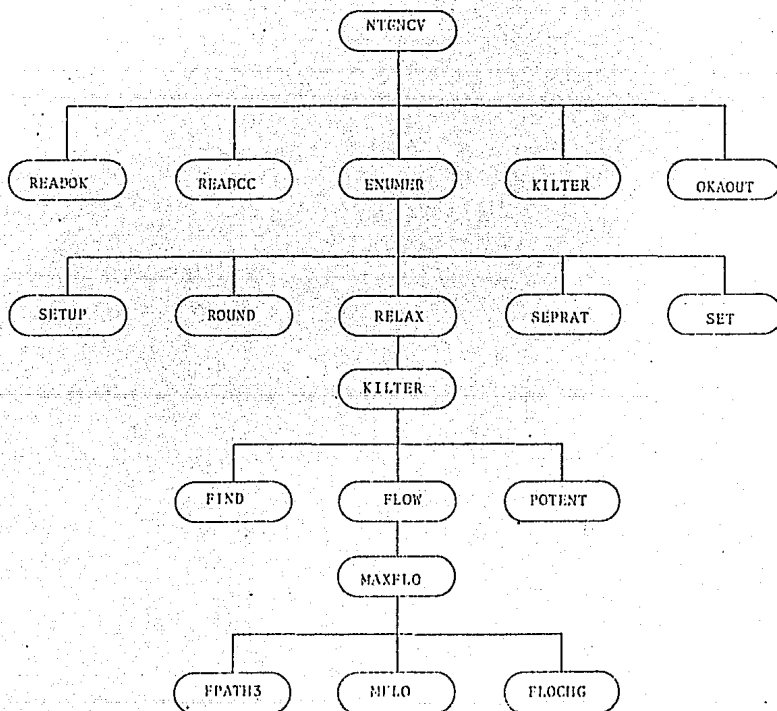


Figura B.5

Relación entre el Algoritmo Global y las Subrutinas Necesarias para Resolver Redes de Flujo con Costo Mínimo Cóncavo

ALGORITMO NTCNCV

PROPOSITO: Controlar las operaciones del procedimiento que resuelve el -- problema de redes de flujo con costo mínimo con función de costo cóncava- (carga fijo) en los arcos.

1. LECTURA. Leer y guardar los datos-- correspondientes a nodos y arcos -- creando al mismo tiempo la repre-- sentación (red circulatoria) que usa -- el Algoritmo de Desviaciones. Leer los flujos factibles iniciales y -- los parámetros asociados con el -- problema de flujo a costo mínimo -- cóncavo. Por último, leer el va-- lor de las variables que controlan las opciones de impresión, ISOLUC- y IPRINT. Si ISOLUC es distinto de cero no se determina la solución.- IPRINT puede asumir los valores 0 (sólo se imprime la solución ópti- ma) y 1 (se imprime además la solu- ción del problema relajado en cada vértice y la solución redondeada - asociada). Inicializar en cero los potenciales de nodo, PI. (El va-- lor inicial de los potenciales es- arbitrario).

2. IMPRESION. Imprimir el número de -- nodos, el número de arcos y los da- tos transformados de cada nodo -- [flujo externo fijo $B(I)$, poten- cial inicial $PI(I)$] y arco [nodo -- origen $O(k)$, nodo terminal $T(k)$, -- cota inferior $CL(k)$, cota superior

$C(k)$, costo unitario o lineal $H(k)$, flujo factible inicial $F(k)$].

3. SOLUCION. Si ISOLUC es distinto de cero, no se determina la solución; terminar. Por el contrario llamar a ENUMER para encontrar la solución óptima XB, es decir, se determina que arcos con cargo fijo tendrán flujo mayor que cero (arco abierto) en la solución óptima.

Si $XB(I)=1$, significa que el arco $k:=KC(I)$ con cargo fijo estará abierto en la solución óptima; entonces se asigna a su costo variable el valor correspondiente $[H(K):=HV(I)]$. Por el contrario $XB(I)=0$ y el arco $K:=KC(I)$ deberá tener flujo cero en la solución óptima. Para garantizar esto, se asigna un costo muy grande a su costo variable $[H(k):=BIG]$.

Llamar a KILTER para recuperar la solución óptima. Esto es necesario debido a que ENUMER no guarda los flujos ni los potenciales óptimos. Imprimir los datos de cada arco y su flujo óptimo.

3. OBJETIVO. Calcular el valor de la función objetivo. Para cada arco con cargo fijo: si el arco está abierto, se agrega su cargo fijo a la función objetivo. Por último, agregar el costo lineal de todos los arcos a la función objetivo. Terminar.

ALGORITMO READCC

PROPOSITO: Leer los parámetros asociados con el problema de flujo a costo mínimo cóncavo.

1. INITIAL. Leer los siguientes parámetros:

EPS: número pequeño usado para parar con cero.

BIG: número grande usado para parar con ∞ .

PER: porcentaje de desviación permitido con respecto al valor óptimo.

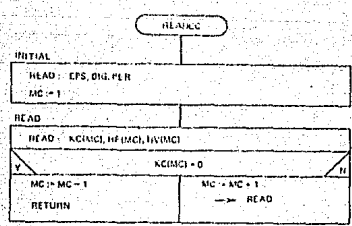
2. Leer los parámetros de los arcos con cargo fijo (costo cóncavo).

KC : es el índice del arco en la red circular creada por el algoritmo READOK. Se recuerda que en ese algoritmo se generan tantos arcos artificiales (excluyendo el arco de retorno) como flujos externos fijos o flujos externos de holgura haya en la red. Al primer arco artificial generado se le asigna el índice 1, al segundo el índice 2, etc. Para esto los nodos se consideran en orden creciente comenzando con el nodo 1.

HF : cargo fijo con el arco.

HV : costo o cargo variable asociado con el arco.

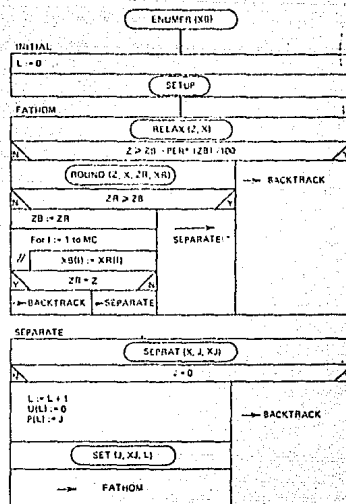
Este paso se repite hasta encontrar un "blanco". Regresar.



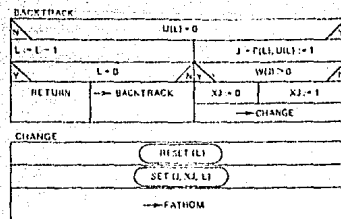
ALGORITMO ENUMER

PROPOSITO: Controlar las operaciones del procedimiento de enumeración implícita que resuelve el problema de redes de flujo con costo mínimo cóncavo (carga fijo).

1. INITIAL. Inicializar en cero el indicador de nivel y los vectores de secuencia P, de valores asignados a las variables W, y el de variables subrayadas U. LLamar a SETUP para crear las aproximaciones lineales en todos los arcos con carga fijo.
2. FATHOM. Usando el algoritmo de desviaciones (KILTER) resolver el problema relajado (es decir, con aproximaciones lineales) asociado con el vértice del árbol de soluciones. Si el valor de la función-objetivo del problema relajado, Z, es mayor o igual a la mejor solución actual, $ZB - PER * |ZB| / 100$, ir al paso 4. Si la cota inferior es menor que la mejor solución actual, encontrar la solución redondeada y su valor objetivo asociado ZR. Si ZR es mayor o igual que ZB, ir al paso 3. Si es menor, actualizar la mejor solución actual haciendo $ZB := ZR$. Si el objetivo redondeado es igual al objetivo relajado, no se podrá obtener una mejora; ir al paso 4. En caso contrario, ir al paso 3.



3. SEPARATE. Encontrar la variable libre a separar mediante el llamado a SEPRAT. Si no hay tal variable, ir al paso 4. Si hay variable separable, avanzar al siguiente nivel en el árbol. Agregar la variable a la secuencia. Hacer cero la subraya para el nuevo nivel. Asignar a la variable su valor al nivel avanzado y regresar al paso 2.
4. BACKTRACK. Si la subraya para el nivel es igual a 0, encontrar la variable $J[J=P(L)]$ del nivel, asignar un 1 a la subraya de este nivel, encontrar el nuevo valor para J , ir al paso 5. Si la subraya es igual a 1, subir un nivel en el árbol [$L:=L-1$]. Si se encuentra el nivel 0, esto es, si el nuevo valor de L es igual a 0, terminar; se ha completado el proceso. Si no se encuentra el nivel 0, repetir el paso 4.
5. CHANGE. Liberar todas las variables fijadas en un nivel igual o mayor que L . Asignar a la variable J el valor determinado en el paso 4. Regresar al paso 2.



La secuencia en el árbol se define mediante el apuntador P el cual tiene un valor para cada nivel. Por ejemplo, $P(L)=I$ indica que la variable J se considera en el nivel L . El vector W se usa para rastrear los valores asignados a las variables fijadas. El vector U contiene la información de las variables subrayadas. Si $U(L)=1$, el correspondiente elemento en $P(L)$ se-

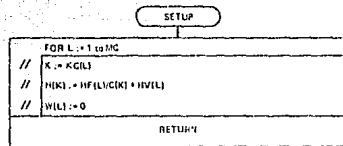
subraya en la secuencia. Si $U(L)=0$ no se subraya. Estos vectores ya se explicaron ampliamente en la sección 3.6.

Otras variables usadas son

- Z : valor de la función objetivo para el problema relajado.
- X : representa la solución al problema relajado. Las variables libres pueden ser fracciones.
- ZB : valor de la función objetivo correspondiente a la mejor solución actual.
- XB : vector binario que representa la mejor solución actual.
- ZR : valor de la función objetivo para la solución redondeada.
- XR : vector binario que representa la solución redondeada.
- J : variable separable.
- XJ : el valor de J, 0 o 1.

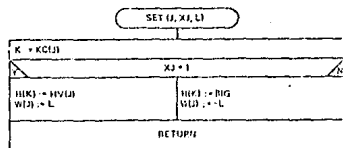
ALGORITMO SETUP

PROPOSITO: Crear las aproximaciones-lineales iniciales para el problema-relajado y hacer todas las variables libres.



ALGORITMO SET

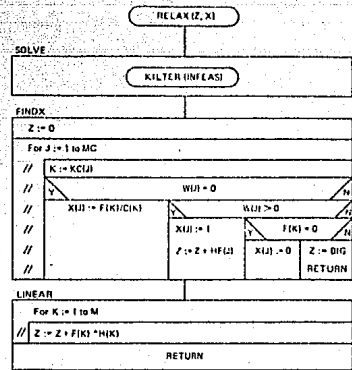
PROPOSITO: Fijar la variable J con el valor XJ al nivel L. Encontrar el arco k asociado con la variable J. Si $XJ=1$, hacer el costo lineal del arco k igual al costo variable de J. Asigne a $W(J)$ el valor L. Esto indica que la variable J se ha fijado en 1 en el nivel L. Si $XJ=0$, hacer el costo lineal del arco k igual a un valor -- grande. Asignar a $W(J)$ el valor -L. Esto indica que la variable J se ha fijado en 0 en el nivel L.



ALGORITMO RELAX

PROPOSITO: Resolver el problema rela-
jado. Calcula los valores de X y el-
valor de la función objetivo Z, para
el problema relajado.

1. SOLVE. Encuentre la solución ópti-
ma al problema relajado usando el
algoritmo de desviaciones (OUT-OF-
KILTER). El algoritmo comienza con
la solución obtenida en la itera-
ción anterior.
2. FINDX. Para cada variable X(J), en-
cuentre el correspondiente arco en
la red original. Si la variable es
libre, calcule el valor fraccional
de X(J). Si la variable J está fi-
ja en 1 hacer X(J)=1 y agregar el
cargo fijo a la función objetivo.-
Si la variable J está fija en 0 y
el arco correspondiente tiene un
flujo distinto de cero, indique la
no factibilidad asignando a Z un
valor grande y regrese. Si la va-
riable J está fija en 0 y el arco
correspondiente tiene flujo cero -
hacer X(J)=0.
3. LINEAR. Agregar a Z el costo li-
neal de todos los flujos y regre-
sar.

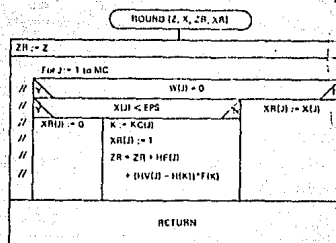


ALGORITMO ROUND

PROPOSITO: Redondear a números enteros (0 ó 1) las variables fraccionales en X para obtener XR, y calcular el valor de ZR correspondiente.

1. Asignar a ZR el valor de Z.
2. Para cada variable X(J):
 - Si J está fija, hacer $XR(J) := X(J)$.
 - Si J es libre y X(J) es igual a 0, hacer $XR(J) := 0$.
 - Si J es libre pero X(J) es distinto de 0, hacer $XR(J) := 1$ e incrementar la función objetivo por la diferencia entre el costo cóncavo y la aproximación lineal subestimada del arco correspondiente a J.

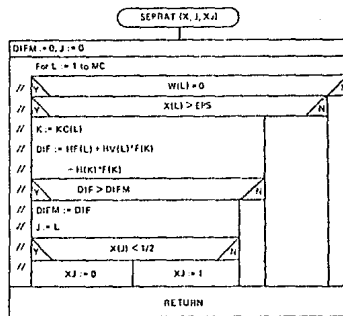
Regresar.



ALGORITMO SEPRAT

PROPOSITO: Dada una solución relajado X, encontrar la variable libre J que proporciona la mayor diferencia entre el costo cóncavo y el costo linealizado. Esta es la variable separable. También determina la dirección de la ramificación ($XJ=0$ ó 1).

1. Haga $DIFM := 0$ y $J := 0$.
2. Para cada variable hacer lo siguiente:
 - Si la variable está fija, tomar la siguiente variable.
 - Si la variable es cero ($X(J) < EPS$),



tomar la siguiente variable. En caso contrario, calcular la diferencia entre la función con costo cóncavo lineal usada para la relajación.

Si esta diferencia es mayor que el valor previamente determinado --- (DIFM), X(J) es la variable separable.

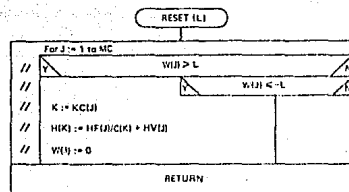
Si X(J) es menor que $\frac{1}{2}$, se ramifica primero en la dirección 0. Por el contrario, en la dirección 1. - Tomar la siguiente variable a menos que esta sea la última.

ALGORITMO RESET

PROPOSITO: Hacer libres todas las variables fijadas al nivel L o mayor.

Para cada variable J:

Si J está fijada en 1 ó 0 al nivel L o mayor, es decir, si el valor absoluto de W(J) es mayor que L, hacer la variable libre asignando a H(K) el valor de la aproximación lineal de la función cóncava y haciendo W(J):=0.



B.3 CODIFICACION FORTRAN IV DEL PROGRAMA NTCNCV

```

10 %
20 % PROGRAMA PRINCIPAL NTCNCV
30 %
40 COMMON /A/O(100)/B/T(100)/G/M,H
50 COMMON /C/PR(100)/F/PI(100)
60 COMMON /K/C(100)/L/H(100)/H/B(100)/N/CL(100)/O/F(100)
70 COMMON /CNCV/EPS,BIG,PER,HC/Z1/HF(100)/Z2/HV(100)/Z3/KC(100)
80 COMMON /Z4/P(100),W(100),U(100)/Z5/ZB
90 COMMON /REPORT/PRINT/Z6/HVERTI,INFEAS
100 DIMENSION LISA(100),LISH(100),XB(100)
110 INTEGER W,XB
120 INTEGER CYC,AD
130 INTEGER O,T,FB,C,D,CL,F,PI
140 INTEGER SC,SK,VR,SH,TH,V
150 %
160 % *****
170 % *
180 % *   ESTA ES LA CODIFICACION FORTRAN IV DEL ALGORITMO :   *
190 % *
200 % *
210 % *
220 % *
230 % *
240 % *   COMO SE PRESENTO EN EL APENDICE B (SECCION B,2) DE LA   *
250 % *
260 % *
270 % *
280 % *   REALIZADA POR EL ING. MARIO PARRA UGALDE PARA ODTE-   *
290 % *   NER EL GRADO DE MAESTRO EN
300 % *
310 % *
320 % *
330 % *   DIVISION DE ESTUDIOS DE POSGRADO FACULTAD DE INGENIERIA   *
340 % *   ( U.E.P.F.I. )
350 % *
360 % *   UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO ( U.N.A.M. )
370 % *   AGUSTO DE 1982
380 % *
390 % *   PROPOSITO :
400 % *
410 % *   NTCNCV SE DISENO PARA IMPLEMENTAR EL METO-
420 % *   DO DE "ENUMERACION IMPLICITA" DE LA SECCION 3.5 QUE
430 % *   RESUELVE EL PROBLEMA DE REDES DE FLUJO CON COSTO HI-
440 % *   NO DONDE LOS ARCOS PUEBEN TENER UNA FUNCION DE COSTO
450 % *   CONCAVA ( CARGO FIJO ) ,
460 % *
470 % *   LAS SUBROUTINAS QUE COMPRENDE ESTE PROGRAMA SON :
480 % *
490 % *
500 % *
510 % *
520 % *   1. LA RED ,
530 % *   2. UNA SOLUCION FACTIBLE INICIAL ,
540 % *   3. UN VALOR PARA COMPARAR CON CERO (POR EJEMPLO 0.001) ,
550 % *   4. UN VALOR PARA COMPARAR CON INFINITO ,
560 % *   5. EL PORCENTAJE DE DESVIACION CON RESPECTO AL OPTIMO,
570 % *   6. LOS PARAMETROS DE CADA ARCO CON COSTO CONCAVO, ES
580 % *   DECIR, EL INDICE CORRESPONDIENTE EN LA RED CIRCULA-
590 % *   TORIA, EL CARGO FIJO Y EL COSTO LINEAL O "VARIABLE"
600 % *
610 % *
620 % *
630 % *****

```

```

640 %
650 % LECTURA
660 %
670 %     CALL READOK
680 %
690 % LECTURA DEL FLUJO FACTIBLE EN CADA ARCO DE LA R; CIRCULATORIA
700 %
710 %     READ 11,(F(K),K=1,N)
720 %     11 FORMAT (BF10.0)
730 %
740 % LECTURA DE LOS DATOS ASOCIADOS CON EL PROBLEMA CONCAVO
750 %
760 %     CALL READCC
770 %
780 % LECTURA DE LAS OPCIONES DE REPORTES
790 %
800 %     READ 12, ISOLUC, IPRINT
810 %     12 FORMAT(2I1)
820 %
830 % IMPRESION
840 %
850 %     PRINT 1, N,M
860 %
870 %     1 FORMAT(1H1,3(/),10X,'RED DE FLUJO CON COSTO CONCAVO',/,10X,
880 % *30('--'),3(/), 10X,'REPRESENTACION DE LA RED CIRCULATORIA',
890 % *3(/),10X,'NUMERO DE NODOS = ',I4,10X,'NUMERO DE ARCOS = ',I4)
900 %
910 %     PRINT 2
920 %     2 FORMAT(2(/),10X,'LOS FLUJOS EXTERNOS Y EL POTENCIAL INICIAL'
930 % * ' DE CADA NODO ES 0.' )
940 %
950 %     PRINT 3, (K,D(K),T(K),CL(K),C(K),H(K),F(K), K=1,M )
960 %
970 %     3 FORMAT(2(/),10X,'ARCO',4X,'N.ORIGEN',4X,'N.TERMINAL',5X,'COTA INF'
980 % * ',',5X,'COTA SUP.',6X,'COSTO',4X,'FLUJO FACTIBLE INICIAL',/,/,
990 % * 12X, 'K',7X,
1000 % * 'D(K)',9X,'T(K)',9X,'CL(K)',9X,'C(K)',11X,'H(K)',11X,'F(K)', /,
1010 % * (8X,I5,6X,I5,8X,I5,3X,I11,2X,I11,4X,F11.2,4X,I11) )
1020 %
1030 %     PRINT 5, ISOLUC, IPRINT
1040 %     5 FORMAT(2(/),10X,'OPCIONES DE REPORTES : ISOLUC = ',I1,10X,
1050 % * ' IPRINT = ',I1 )
1060 %
1070 %     IF( ISOLUC.NE.0 ) STOP
1080 %
1090 %     DO 13 I = 1,N
1100 %     13 PI(I) = 0
1110 %     ZB = BIG
1120 %
1130 % SOLUCION
1140 %
1150 %     CALL ENUMER(XB)
1160 %
1170 %     DO 15 I = 1,NC
1180 %     K = KC(I)
1190 %     H(K) = BTG
1200 %     IF ( XB(I) .NE. 0) H(K) = HV(I)
1210 %     15 CONTINUE

```



```
1220 %  
1230 % RECUPERACION DE LOS FLUJOS OPTIMOS  
1240 %  
1250     CALL KILTER (INFEAS)  
1260 %  
1270 % CALCULO DEL VALOR DE LA F. OBJETIVO PARA LA SOLUCION OPTIMA  
1280 %  
1290     Z=0.  
1300 %  
1310     DO 16 I = 1,MC  
1320     IF (XB(I) .NE. 0) Z = Z + HF(I)  
1330 16 CONTINUE  
1340 %  
1350     DO 17 K = 1,M  
1360     Z = Z+F(K) * H(K)  
1370 17 CONTINUE  
1380 %  
1390     PRINT 21  
1400 21 FORMAT( 1H1,3(/),10X,'SOLUCION OPTIMA',/,10X,15('-') )  
1410 %  
1420     PRINT 22, ( XB(I),I=1, MC )  
1430 %  
1440 22 FORMAT( 2(/),10X,'XB', 1(/), (10X,25I3) )  
1450 %  
1460     CALL OKAOUT  
1470 %  
1480     PRINT 20,Z  
1490 20 FORMAT( 1(/),71X,'COSTO TOTAL DEL FLUJO ',F15.2 )  
1500 %  
1510     PRINT 23, NVERTI  
1520 23 FORMAT( 2(/),10X,'NUMERO DE PROBLEMAS RELAJADOS RESUELTOS =',I4 )  
1530 %  
1540     STOP  
1550     END  
1560  
1570  
1580  
1590  
1600
```

```

1610      SUBROUTINE READOK
1620 %
1630      COMMON /A/D(100)/B/T(100)/N/CL(100)/K/C(100)/L/H(100)/G/M;N
1640      * /H/B(100)
1650      COMMON /P/PI(100)/D/F(100)
1660      INTEGER O,T,CL,C,CUM,SOURCE,SINK,PI,F
1670      REAL LOWER
1680 %
1690 % *****
1700 % * * * * *
1710 % * PROPOSITO :
1720 % * LEER Y ALMACENAR LOS DATOS CORRESPONDIENTES *
1730 % * A NODOS Y ARCOS CREANDO, AL MISMO TIEMPO, LA REPRESENTACION *
1740 % * TACION ( RED CIRCULATORIA ) QUE REQUIERE EL ALGORITMO *
1750 % * DE DESVIACIONES ( 'OUT-OF-KILTER ALGORITHM' ), *
1760 % * * * * *
1770 % *****
1780 %
1790      N=0
1800      CUM = 0
1810 %
1820      READ 1, N
1830      1 FORMAT(15)
1840 %
1850      SOURCE =N+1
1860      SINK = N+2
1870      N=N+2
1880 %
1890 % NODE
1900 %
1910      52 READ 0,I,BF,BS,CS
1920      0 FORMAT (15,3F10.0)
1930 %
1940      IF ( I .EQ. 0 ) GO TO 51
1950      IF ( BF .EQ. 0. ) GO TO 4
1960      M=M+1
1970      IF ( BF .LT. 0. ) GO TO 3
1980      O(M) = SOURCE
1990      T(M) = I
2000      CL(M) = BF
2010      C(M) = BF
2020      GO TO 2
2030 %
2040      3 O(M) = I
2050      T(M) = SINK
2060      CL(M) = -BF
2070      C(M) = -BF
2080      2 H(M) = 0
2090      CUM = CUM + C(M)
2100      4 IF ( BS .EQ. 0. ) GO TO 52
2110      M =M+1
2120      IF ( BS .LT. 0. ) GO TO 5
2130      O(M) = SOURCE
2140      T(M) = I
2150      C(M) = BS
2160      GO TO 4
2170 %
2180      5 O(M) = I
2190      T(M) = SINK
2200      C(M) = -BS
2210      6 H(M) = CS
2220      CL(M) = 0
2230      CUM = CUM + C(M)
2240      GO TO 52

```

```
2250 %  
2260 % ARC  
2270 %  
2280 51 CONTINUE  
2290 M= M+1  
2300 %  
2310 READ 9,I,J,LOWER,UPPER,COST  
2320 9 FORMAT (2I5,3F10.0)  
2330 %  
2340 IF ( I .NE. 0) GO TO 7  
2350 O(M) = SINK  
2360 T(M) = SOURCE  
2370 CL(M) = 0  
2380 C(M) = CUM  
2390 H(M) = 0  
2400 GO TO 999  
2410 %  
2420 7 O(M) = I  
2430 T(M) = J  
2440 CL(M) = LOWER  
2450 C(M) = UPPER  
2460 H(M) = COST  
2470 CUM = CUM +C(M)  
2480 GO TO 51  
2490 %  
2500 % BZERO  
2510 %  
2520 999 CONTINUE  
2530 DO 55 I = 1,N  
2540 P(I) = 0  
2550 55 B(I) = 0  
2560 %  
2570 DO 56 K = 1,M  
2580 56 F(K) = 0  
2590 %  
2600 RETURN  
2610 END  
2620  
2630  
2640  
2650  
2660
```

```

2670      SUBROUTINE READCC
2680 %
2690      COMMON /CNCV,EPS,BIG,PER,HC,Z1/HF(100)/Z2/HV(100)/Z3/KC(100)
2700 %
2710 % *****
2720 % *
2730 % *   PROPOSITO :
2740 % *           LER LOS PARAMETROS ASOCIADOS CON EL PRO-
2750 % *   BLENA : REDES DE FLUJO CON COSTO MINIMO CONCAVO .
2760 % *
2770 % *****
2780 %
2790 %
2800 % INITIAL
2810 %
2820      READ 1,EPS,BIG,PER
2830      1 FORMAT(2F10.0,F5.0)
2840 %
2850      HC = 1
2860 %
2870 % READ
2880 %
2890      2 READ 4,KC(HC),HF(HC),HV(HC)
2900      4 FORMAT ( I5,2F10.0)
2910 %
2920      IF ( KC(HC) .EQ. 0) GO TO 3
2930      HC = HC + 1
2940      GO TO 2
2950 %
2960      3 HC = HC-1
2970      RETURN
2980      END
2990
3000
3010
3020
3030

```

```

3040      SUBROUTINE ENUMER (XB)
3050 Z
3060      COMMON /CNCV/EPS,BIG,PER,MC,Z4/P(100),W(100),U(100)/Z5/ZD
3070      COMMON /Z1/HF(100)/Z2/HV(100)/Z3/KC(100)/L/H(100)
3080      COMMON /REPORT/IFPRINT/Z6/NVERTI,INFEAS
3090 Z
3100      DIMENSION XB(100),X(100),XR(100)
3110      INTEGER XB,XR,P,W,U,XJ
3120 Z
3130 Z *****
3140 Z *
3150 Z * PROPOSITO :
3160 Z * CONTROLAR LAS OPERACIONES DEL PROCEDIMIENT
3170 Z * TO DE ENUMERACION IMPLICITA QUE RESUELVE EL PROBLEMA
3180 Z * DE REDES DE FLUJO CON COSTO MINIMO CONCAVO ( CARGO
3190 Z * FIJO ) .
3200 Z *
3210 Z *****
3220 Z
3230 Z
3240 Z INITIAL
3250 Z
3260      L=0
3270      DO 290 I =1,MC
3280      P(I)=0
3290      W(I)=0
3300      U(I)=0
3310      290 CONTINUE
3320 Z
3330      CALL SETUP(MC)
3340 Z
3350      PRINT 4, (I,KC(I),HF(I),HV(I),H( KC(I) )), I=1,MC)
3360 Z
3370      4 FORMAT( 3(/),10X,'DATOS ASOCIADOS CON EL PROBLEMA CONCAVO',3(/),
3380      * 10X,'INDICE DEL ARCO EN',4X,'INDICE DEL ARCO EN',5X,'CARGO FIJO',
3390      * 5X,'COSTO LINEAL',5X,'APROXIMACION LINEAL PARA', /,
3400      * 12X,'LA RED ORIGINAL',5X,'LA RED CIRCULATORIA',36X,'EL PROBLEMA'
3410      * ,' RELAJADO', /,19X,'I',17X,'K = KC(I)',12X,'HF(I)',11X,'HV(I)',
3420      * 18X,'H(K)',2(/), (17X,I3,20X,I3,9X,F11.2,5X,F11.2,11X,F11.3) )
3430 Z
3440      PRINT 6,EPS,BIG,PER
3450      6 FORMAT( 2(/),10X,'VALOR PARA COMPARAR CON CERO ',F10.5,/,10X,
3460      * 'VALOR PARA COMPARAR CON INFINITO ',F10.0,2(/),10X,
3470      * 'PORCENTAJE DE DESVIACION CON RESPECTO AL VALOR OPTIMO ',F5.1 )
3480 Z
3490      NVERTI = 0
3500 Z
3510 Z FATHON
3520 Z
3530      53 CONTINUE
3540      NVERTI = NVERTI+1
3550 Z
3560      CALL RELAX (Z,X)
3570 Z
3580      ZA = ZB - PER * ABS(ZB)/100
3590 Z
3600      IF( IPRINT,EQ,0 ) GO TO 111
3610 Z
3620      PRINT 118, ZB,ZA
3630      118 FORMAT( 1(/)10X,'MEJOR SOLUCION ACTUAL = ZB = ',F15.2,2(/),
3640      * 10X,'ZA = ZB - PER/100 * ZB/ = ',F15.2 )
3650 Z

```

```

3660 111 CONTINUE
3670 %
3680 IF ( Z .GE. ZA) GO TO 51
3690 %
3700 CALL ROUND (Z,X,ZR,XR)
3710 %
3720 IF( IPRINT.EQ.0 )GO TO 10
3730 %
3740 PRINT 119, ( XR(I),I=1,MC )
3750 119 FORMAT( 1(/),10X,'SOLUCION REDONDEADA', 1(/),10X,'XR',
3760 * /, (10X,25I3) )
3770 %
3780 PRINT 120, ZR
3790 120 FORMAT( 1(/),10X,'COSTO TOTAL DEL FLUJO = ZR = ',F15.2 )
3800 %
3810 10 CONTINUE
3820 %
3830 IF (ZR .GE. ZR) GO TO 52
3840 ZR = ZR
3850 DO 1 I = 1,MC
3860 1 XB(I) = XR(I)
3870 IF ( ZR .EQ. Z) GO TO 51
3880 %
3890 % SEPARATE
3900 %
3910 52 CONTINUE
3920 %
3930 CALL SEPRAT ( X,J,XJ)
3940 %
3950 IF ( J .EQ. 0) GO TO 51
3960 L = L+1
3970 U(L) = 0
3980 P(L) = J
3990 %
4000 CALL SET(J,XJ,L)
4010 %
4020 GO TO 53
4030 %
4040 % BACKTRACK
4050 %
4060 51 CONTINUE
4070 IF ( U(L) .EQ. 0) GO TO 54
4080 P(L) = 0
4090 U(L) = 0
4100 L= L-1
4110 IF ( L .EQ. 0) GO TO 999
4120 GO TO 51
4130 %
4140 54 J = P(L)
4150 U(L) = 1
4160 XJ = 1
4170 IF ( W(J) .GT. 0) XJ = 0
4180 %
4190 % CHANGE
4200 %
4210 CALL RESET (L)
4220 %
4230 CALL SET ( J,XJ,L)
4240 %
4250 GO TO 53
4260 %
4270 999 CONTINUE
4280 RETURN
4290 END
4300
4310
4320
4330
4340

```

```

4350      SUBROUTINE SETUP(MC)
4360      COMMON /Z3/KC(100)/Z1/HF(100)/Z2/HV(100)/Z4/P(100),W(100),U(100)
4370      COMMON /L/H(100)/K/C(100)
4380      INTEGER W,C
4390 Z
4400 Z      *****
4410 Z      *
4420 Z      * PROPOSITO :
4430 Z      *          CREAR LAS APROXIMACIONES LINEALES INICIA-
4440 Z      *          LES PARA EL PROBLEMA RELAJADO Y HACER TODAS LA VARIA-
4450 Z      *          BLES LIBRES ,
4460 Z      *
4470 Z      *****
4480 Z
4490      DO 1 L = 1,MC
4500      K = KC(L)
4510      H(K) = HF(L)/C(K) + HV(L)
4520      W(L) = 0
4530      1 CONTINUE
4540      RETURN
4550      END
4560 Z
4570
4580
4590
4600
4610      SUBROUTINE SET ( J,XJ,L)
4620      COMMON /L/H(100)/Z2/HV(100)/Z4/P(100),W(100),U(100)
4630      COMMON /CHCV/EFS,BIG,PER,MC/Z3/KC(100)
4640      INTEGER W,XJ
4650 Z
4660 Z      *****
4670 Z      *
4680 Z      * PROPOSITO :
4690 Z      *          FIJAR LA VARIABLE J CON EL VALOR XJ AL NI-
4700 Z      *          VEL L ,
4710 Z      *
4720 Z      *****
4730 Z
4740      K= KC(J)
4750      IF ( XJ ,HE, 1) GO TO 51
4760      H(K) = HV(J)
4770      W(J) = L
4780      GO TO 999
4790 Z
4800      51 H(K) = BIG
4810      W(J) = -L
4820      999 CONTINUE
4830      RETURN
4840      END
4850
4860
4870
4880
4890

```

```

4900      SUBROUTINE RELAX (Z,X)
4910 Z
4920      COMMON /Z3/KC(100)/D/F(100)/K/C(100)/Z1/HF(100)/L/H(100)
4930      COMMON /CNCV/EPS,RIG,PER,HC/G/H,H/Z4/P(100),W(100),U(100)
4940      COMMON /REPORT/IPRINT/Z6/NVERTI,INFEAS
4950      DIMENSION X(100)
4960      INTEGER P,W,U,F,C
4970 Z
4980 Z
4990 Z      *****
5000 Z      * PROPOSITO : *
5010 Z      * RESOLVER EL PROBLEMA RELAJADO MEDIANTE EL *
5020 Z      * ALGORITMO DE DESVIACIONES. CALCULA LAS VARIABLES EN EL *
5030 Z      * VECTOR X Y EL VALOR DE LA FUNCION OBJETIVO, Z. *
5040 Z      * *
5050 Z      *****
5060 Z
5070 Z
5080 Z      SOLVE
5090 Z
5100      CALL KILTER(INFEAS)
5110 Z
5120      IF( INFEAS.EQ.0 )GO TO 61
5130 Z
5140      Z = RIG
5150 Z
5160      IF( IPRINT.EQ.0 )GO TO 61
5170 Z
5180      PRINT 62, NVERTI
5190      62 FORMAT( 1H1,4(/),10X,'EL PROBLEMA RELAJADO EN EL VERTICE ',I3,
5200      * ' NO TIENE SOLUCION ',1(/),10X,'LOS VECTORES P, W, U PARA EL',
5210      * ' VERTICE SON :' )
5220 Z
5230      PRINT 113, ( P(I),I=1,NC )
5240      PRINT 114, ( W(I),I=1,NC )
5250      PRINT 115, ( U(I),I=1,NC )
5260 Z
5270      PRINT 66
5280      66 FORMAT( 2(/),10X,'ESTADO FINAL DEL PROCESO DE SOLUCION AL',
5290      * ' DETERMINARSE LA NO FACTIBILIDAD' )
5300 Z
5310      CALL OKAOUT
5320 Z
5330      RETURN
5340 Z
5350      61 CONTINUE
5360 Z
5370 Z      FINDX
5380 Z
5390      Z =0,
5400      DO 1 J = 1,NC
5410      K = KC(J)
5420      IF ( W(J) .NE. 0) GO TO 51
5430      X(J) = FLOAT(F(K))/FLOAT(C(K))
5440      GO TO 1
5450 Z
5460      51 IF ( W(J) .LT. 0) GO TO 53
5470      X(J) = 1,
5480      Z = Z+HF(J)
5490      GO TO 1
5500 Z
5510      53 IF ( F(K) .EQ. 0) GO TO 54
5520      Z = RIG
5530 Z

```



```

5540      IF( IPRINT.EQ.0 )GO TO 67
5550 Z
5560      PRINT 73, NVERTI, J,K,F(K)
5570      73 FORMAT( 1H1,4(/),10X,'EL PROBLEMA RELAJADO EN EL VERTICE ',I3,
5580      * ' NO TIENE SOLUCION',1(/),10X,'LA VARIABLE X(',I3,') ESTA'
5590      * ' FIJADA EN 0 Y EL FLUJO EN EL ARCO CORRESPONDIENTE ES DISTINTO',
5600      * ' DE 0 , F(',I3,') = ',I9 )
5610 Z
5620      PRINT 113, ( P(I),I=1,MC )
5630      PRINT 114, ( U(I),I=1,MC )
5640      PRINT 115, ( U(I),I=1,MC )
5650 Z
5660      67 CONTINUE
5670      RETURN
5680 Z
5690      54 X(J) = 0.
5700      1 CONTINUE
5710 Z
5720 Z LINEAR
5730 Z
5740      DO 2 K = 1,N
5750      2 Z = Z + F(K)*H(K)
5760 Z
5770 Z
5780      IF( IPRINT.EQ.0 )GO TO 999
5790 Z
5800      PRINT 112, NVERTI
5810      112 FORMAT(1H1,3(/),10X,'PROBLEMA RELAJADO EN EL VERTICE ',I3,/,
5820      * 10X,35('-') )
5830 Z
5840      PRINT 113, ( P(I),I=1,MC )
5850      PRINT 114, ( U(I),I=1,MC )
5860      PRINT 115, ( U(I),I=1,MC )
5870      113 FORMAT( 1(/),10X,'P',/, (10X,20I4) )
5880      114 FORMAT( 1(/),10X,'U',/, (10X,20I4) )
5890      115 FORMAT( 1(/),10X,'U',/, (10X,20I4) )
5900 Z
5910      PRINT 116, ( X(I),I=1,MC )
5920      116 FORMAT( 1(/),10X,'X',/, (10X,20F6.3) )
5930 Z
5940      CALL OKADUI
5950 Z
5960      PRINT 117,Z
5970      117 FORMAT( 1(/),71X,'COSTO TOTAL DEL FLUJO ',F15.2 )
5980 Z
5990      999 CONTINUE
6000 Z
6010      RETURN
6020      END
6030
6040
6050
6060
6070

```

```

6080      SUBROUTINE ROUND (Z,X,ZR,XR)
6090 Z
6100      COMMON /Z4/P(100),W(100),U(100)/Z3/KC(100)/Z1/HF(100)/Z2/HV(100)
6110      COMMON /CHCV/EFS,BIG,FER,HC/O,F(100)/L/H(100)
6120      DIMENSION X(100),XR(100)
6130      INTEGER XR,W,F
6140 Z
6150 Z      *****
6160 Z      *
6170 Z      * PROPOSITO :
6180 Z      * REDONDEAR A NUMEROS ENTEROS ( 0 0 1 ) LAS
6190 Z      * VARIABLES FRACCIONALES EN X PARA OBTENER XB, Y CAL-
6200 Z      * CULAR EL VALOR DEL ZR CORRESPONDIENTE ,
6210 Z      *
6220 Z      *****
6230 Z
6240      ZR = Z
6250 Z
6260      DO 1 J = 1,MC
6270      IF ( W(J) ,NE. 0) GO TO 51
6280      IF ( X(J) ,LT. EPS) GO TO 52
6290      K = KC(J)
6300      XR(J) = 1
6310      ZR = ZR*HF(J) +(HV(J)-H(K))*F(K)
6320      GO TO 1
6330 Z
6340      51 XR(J) = X(J)
6350      GO TO 1
6360 Z
6370      52 XR(J) = 0
6380      1 CONTINUE
6390      RETURN
6400      END
6410
6420
6430
6440
6450

```

```

6460      SUBROUTINE SEPRAT (X,J,XJ)
6470 %
6480      COMMON /Z4/P(100),W(100),U(100)/Z3/KC(100)/Z2/HV(100)/Z1/HF(100)
6490 *      /L/H(100)/D/F(100)
6500      COMMON /CHCV/EPS,BIG,PER,MC
6510      DIMENSION X(100)
6520      INTEGER W,XJ,F
6530 %
6540 %
6550 %
6560 % * PROPOSITO !
6570 % * DADA UNA SOLUCION RELAJADA X, ENCONTRAR LA
6580 % * VARIABLE LIBRE J QUE PROPORCIONA LA MAYOR DIFERENCIA
6590 % * ENTRE LOS COSTOS CONCAVO Y LINEALIZADO, TAMBIEN DETER-
6600 % * MINA LA DIRECCION DE LA RAMIFICACION ( XJ=0 O 1 ),
6610 % *
6620 % *****
6630 %
6640      DIFM = 0
6650      J=0
6660      DO 1 L=1,MC
6670      IF (W(L) .NE. 0) GO TO 1
6680      IF ( X(L) .LE. EPS) GO TO 1
6690      K = KC(L)
6700      DIF = HF(L) +HV(L)*F(K)-H(K)*F(K)
6710      IF ( DIF .LE. DIFM) GO TO 1
6720      DIFM = DIF
6730      J = L
6740      XJ = 1
6750      IF ( X(J) .LT. .5) XJ = 0
6760 %
6770 1 CONTINUE
6780      RETURN
6790      END
6800
6810
6820
6830
6840
6850      SUBROUTINE RESET (L)
6860 %
6870      COMMON /Z4/P(100),W(100),U(100)/Z3/KC(100)/L/H(100)/Z1/HF(100)
6880 *      /Z2/HV(100)/K/C(100)
6890      COMMON /CHCV/EPS,BIG,PER,MC
6900      INTEGER W,C
6910 %
6920 % *****
6930 % *
6940 % * PROPOSITO !
6950 % * HACER LIBRES TODAS LAS VARIABLES FIJADAS AL
6960 % * NIVEL L O MAYOR .
6970 % *
6980 % *****
6990 %
7000      DO 1 J = 1,MC
7010      IF ( IABS(W(J)) .LT. L) GO TO 1
7020      K = KC(J)
7030      H(K) = HF(J)/C(K) + HV(J)
7040      W(J) = 0
7050 1 CONTINUE
7060      RETURN
7070      END
7080
7090
7100
7110
7120

```

```

7130      SUBROUTINE KILTER (INFEAS)
7140 Z
7150      INTEGER FIN,CH,DF
7160 Z
7170 Z      *****
7180 Z      *
7190 Z      * PROPOSITO ;
7200 Z      * DADO UN FLUJO INICIAL QUE SATISFACE LA CON-
7210 Z      * SERVACION DE FLUJO EN CADA NODO Y POTENCIALES ARBITRA-
7220 Z      * RIOS EN ESTOS ; MODIFICAR LOS FLUJOS Y LOS POTENCIALES
7230 Z      * HASTA OBTENER LA SOLUCION OPTIMA. SI NO HAY SOLUCION
7240 Z      * FACTIBLE, SE REGRESA INFEAS=1.
7250 Z      *
7260 Z      *****
7270 Z
7280 Z
7290 Z      INITIAL
7300 Z
7310      KO = 1
7320      FIN = 0
7330      INFEAS = 0
7340 Z
7350 Z      FIND
7360 Z
7370      1 CALL FIND(KO,IO,JO,CH,DF,FIN)
7380 Z
7390      IF ( FIN .EQ. 1) RETURN
7400 Z
7410 Z      FLOW
7420 Z
7430      2 CALL FLOW (KO,IO,JO,CH,DF,INF)
7440 Z
7450      IF ( INF .NE. 1) GO TO 1
7460 Z
7470 Z      POTENT
7480 Z
7490      CALL POTENT (KO,IO,JO,CH,KE,INFEAS)
7500 Z
7510      IF ( INFEAS .EQ. 1) RETURN
7520      IF ( KE .EQ. KO) GO TO 1
7530      GO TO 2
7540 Z
7550      RETURN
7560      END
7570
7580
7590
7600
7610

```

```

7620      SUBROUTINE FIND (KO,IO,JO,CH,DF,FIN)
7630 %
7640      COMMON /A/D(100)/B/T(100)/P/PI(100)/L/H(100)/O/F(100)/N/CL(100)
7650      * /G/M,N/K/C(100)
7660      INTEGER O,T,PI,F,CL,C,D,DF,CH,FIN
7670 %
7680 % *****
7690 % *
7700 % * PROPOSITO :
7710 % * ENCONTRAR UN ARCO KO NO CONFORMABLE (CON *
7720 % * DESVIACION), SU NODO ORIGEN IO Y SU NODO TERMINAL JO, *
7730 % * LA DIRECCION DEL CAMBIO DE FLUJO SE INDICA POR CH (+1 *
7740 % * PARA INCREMENTO, -1 PARA DECREMENTO), LA CANTIDAD DE *
7750 % * FLUJO A CIRCULAR (PASAR) PARA QUE KO SEA CONFORMABLE *
7760 % * ES DF .
7770 % *
7780 % *****
7790 %
7800      K=KO
7810 %
7820 % TESTFIN
7830 %
7840      10 CONTINUE
7850      IF ( K .GT. N) GO TO 998
7860      I = O(K)
7870      J=T(K)
7880      D=PI(I) -PI(J)+H(K)
7890      IF ( D .GT. 0) GO TO 51
7900      IF ( D .EQ. 0) GO TO 52
7910      GO TO 54
7920 %
7930 % ALPHA
7940 %
7950      51 CONTINUE
7960      IF (F(K) .EQ. CL(K)) GO TO 20
7970      IF (F(K) .GE. CL(K)) GO TO 30
7980      CH = 1
7990      DF = CL(K) - F(K)
8000      GO TO 53
8010 %
8020      30 CH = -1
8030      DF = F(K) - CL(K)
8040      GO TO 53
8050 %
8060      20 K = K+1
8070      GO TO 10
8080 %
8090 % BETA
8100 %
8110      52 CONTINUE
8120      IF ( F(K) .LT. CL(K)) GO TO 40
8130      IF ( F(K) .LE. C(K)) GO TO 50
8140      CH = -1
8150      DF = F(K) - C(K)
8160      GO TO 53
8170 %
8180      50 K=K+1
8190      GO TO 10
8200 %
8210      40 CH =1
8220      DF = CL(K) - F(K)
8230      GO TO 53

```

```
0240 %  
0250 % RHO  
0260 %  
0270 54 CONTINUE  
0280 IF (F(K) .EQ. C(K)) GO TO 60  
0290 IF ( F(K) .LT. C(K)) GO TO 70  
0300 CH = -1  
0310 DF = F(K) - C(K)  
0320 GO TO 53  
0330 %  
0340 60 K=K+1  
0350 GO TO 10  
0360 %  
0370 70 CH = 1  
0380 DF = C(K)-F(K)  
0390 %  
0400 % FINISH  
0410 %  
0420 53 CONTINUE  
0430 KD = K  
0440 IO=I  
0450 JO = J  
0460 GO TO 999  
0470 %  
0480 998 FIN=1  
0490 999 CONTINUE  
0500 RETURN  
0510 END  
0520  
0530  
0540  
0550  
0560
```

```

0570 SUBROUTINE FLOW (KO,IO,JO,CH,DF,INF)
0580 Z
0590 COMMON /O/F(100)
0600 INTEGER SN,TH,CH,V,F,DF
0610 Z
0620 Z *****
0630 Z *
0640 Z * PROFUSITO !
0650 Z * CAMBIAR LOS FLUJOS EN LA RED DE MANERA QUE
0660 Z * EL FLUJO EN EL ARCO KO(IO,JO) FUERA INCREMENTARSE O
0670 Z * DISMINUIRSE, Y A AL VEZ MANTENER LA CONSERVACION DE
0680 Z * FLUJO EN CADA NODO ,
0690 Z *
0700 Z *****
0710 Z
0720 IF ( CH .GT. 0) GO TO 10
0730 SN=IO
0740 TH=JO
0750 GO TO 20
0760 Z
0770 10 SN =JO
0780 TH=IO
0790 Z
0800 20 CALL MAXFLO(SN,TH,DF,V,INF)
0810 Z
0820 DF = DF -V
0830 IF ( CH .GT.0) GO TO 30
0840 F(KO) = F(KO) - V
0850 GO TO 999
0860 Z
0870 30 F(KO) = F(KO) + V
0880 999 CONTINUE
0890 RETURN
0900 END
0910
0920
0930
0940
0950

```

```

0760      SUBROUTINE POTENT (KO,IO,JO,CH,KE,INFEAS)
0770 %
0780      COMMON/MB/S(100)
0790      COMMON /O/F(100)/N/CL(100)/P/PI(100)/K/C(100)/G/M,N/L/H(100)
0800      COMMON /A/U(100)/B/T(100)
0810      INTEGER CH,S,F,CL,PI,C,O,T,D,DP
0820 %
0830 %
0840 %
0850 %
0860 %
0870 %
0880 %
0890 %
0900 %
0910 %
0920 %
0930 %
0940 %
0950 %
0960 %
0970 %
0980 %
0990 %
1000 %
1010 %
1020 %
1030 %
1040 %
1050 %
1060 %
1070 %
1080 %
1090 %
1100 %
1110 %
1120 %
1130 %
1140 %
1150 %
1160 %
1170 %
1180 %
1190 %
1200 %
1210 %
1220 %
1230 %
1240 %
1250 %
1260 %
1270 %
1280 %
1290 %
1300 %
1310 %
1320 %
1330 %
1340 %
1350 %
1360 %
1370 %
1380 %
1390 %
1400 %
1410 %
1420 %
1430 %
1440 %
1450 %
1460 %
1470 %
1480 %
1490 %
1500 %
1510 %
1520 %
1530 %
1540 %
1550 %
1560 %
1570 %
1580 %
1590 %
1600 %
1610 %
1620 %
1630 %
1640 %
1650 %
1660 %
1670 %

```

```

*****
*
* PROPOSITO :
* DETERMINAR EL VALOR DEL INCREMENTO EN LOS
* POTENCIALES DE LOS NODOS QUE PERTENECEN A N2, PARA
* CONDUCCION AL ARCO KO A UN ESTADO CONFORMABLE ( SIN DES-
* VIACION ) O CONVERTIR ADMISIBLE UNO DE LOS ARCOS QUE
* SE ORIGINA EN N1 Y TERMINA EN N2 O, VICEVERSA .
*
*****

```

INITIAL

```

IF (CH .LT. 0) GO TO 10
IF ( F(KO) .GE. CL(KO)) GO TO 20
40 DP = 9999
KE = 0
GO TO 30
20 DP = -(PI(IO) - PI(JO) + H(KO))
KE = KO
GO TO 30
10 IF ( F(KO) .LE. C(KO)) GO TO 40
DP = PI(IO) - PI(JO) + H(KO)
KE = KO

```

DUAL

```

30 CONTINUE
DO 50 K = 1,M
I = O(K)
J= T(K)
IF ( S(I) .EQ. 0) GO TO 60
IF ( S(J) .NE. 0) GO TO 50
IF ( F(K) .GE. C(K)) GO TO 50
D = PI(I)-PI(J)+H(K)
GO TO 90
60 IF ( S(J) .EQ. 0) GO TO 50
IF ( F(K) .LE. CL(K)) GO TO 50
D= PI(J) - PI(I) - H(K)
90 IF ( D .GE. DP ) GO TO 50
DP = D
KE = K
50 CONTINUE

```

CHANGE

```

IF ( KE .EQ. 0) GO TO 70
DO 80 I = 1,N
IF ( S(I) .EQ. 0) PI(I) = PI(I) + DP
80 CONTINUE
RETURN
70 INFEAS = 1
RETURN
END

```



```

9600      SUBROUTINE MAXFLO(SN,TH,DF,U,INF)
9690 %
9700      COMMON /A/O(100)/B/T(100)/C/PB(100)/F/FR(100)/G/H,N
9710      * /K/C(100)/L/H(100)/M/D(100)
9720      * /N/CL(100)/O/F(100)/B/FF(100)/P/PI(100)
9730      DIMENSION LISA(100),LISH(100)
9740      INTEGER U,F,PD, C,B,CL,F,PI ,SN,TH,DEL,U,DF,DWES
9750 %
9760 % *****
9770 % *
9780 % * PROPOSITO :
9790 % * DETERMINA LA RED CON SOLO ARCOS ADMISIBLES *
9800 % * PARA LA CUAL SE OBTIENE UN FLUJO DADO (DF) DEL NODO *
9810 % * FUENTE (SN) AL NODO SUMIDERO (TH), SI NO SE PUEDE OB- *
9820 % * TENER TAL FLUJO, SE PROPORCIONA EL FLUJO MAXIMO (U) DE *
9830 % * SN A TH . ESTE PROCEDIMIENTO SUPONE QUE SE TIENE A MA- *
9840 % * NO UN FLUJO FACTIBLE INICIAL ,
9850 % *
9860 % *****
9870 %
9880 %
9890 % INITIAL
9900 %
9910      U=0
9920      IFIN=0
9930      INF=0
9940      IF=1
9950 %
9960 % PATH
9970 %
9980      7 CALL FPATH(SN,TH,NP)
9990 %
10000      IF ( NP ,EQ, 0) GO TO 1
10010      INF = 1
10020      GO TO 3
10030 %
10040      1 IC = 0
10050      IJ = TN
10060 %
10070 % LSTARC
10080 %
10090      9 K=PB(IJ)
10100      IC = IC + 1
10110      LIS, (IC) = K
10120      IF (K ,LT, 0) GO TO 6
10130      IJ = O(K)
10140      10 IF ( IJ ,EQ, SN ) GO TO 8
10150      GO TO 9
10160 %
10170      6 IJ = T(-K)
10180      GO TO 10
10190      8 CONTINUE

```

```
10200 %  
10210 % FLOW  
10220 %  
10230     CALL MFLO(LISA,IC,DEL)  
10240 %  
10250     DWES = DEL + V - DF  
10260     IF ( DEL + V .LT. DF) GO TO 4  
10270     DEL=DF-V  
10280     IFIN =1  
10290 %  
10300     4 CALL FLOCHG(LISA,IC,DEL)  
10310 %  
10320     V = V + DEL  
10330     IF ( IFIN .EQ. 0) GO TO 7  
10340 %  
10350     3 CONTINUE  
10360     RETURN  
10370     END  
10380  
10390  
10400  
10410  
10420
```

```

10430      SUBROUTINE FPATH (SN,TN,NP)
10440 Z
10450      COMMON /WB/S(100)
10460      COMMON /A/B(100)/B/T(100)/C/PB(100)/G/H,N
10470      INTEGER O,T,AD,S,PB,SN,TN
10480 Z
10490 Z      *****
10500 Z      *
10510 Z      *   PROPOSITO :
10520 Z      *           ENCONTRAR UNA TRAYECTORIA DE SOLO ARCOS AD- *
10530 Z      *   MMISIBLES EMPLEANDO UN PROCEDIMIENTO DE ETIQUETACION. *
10540 Z      *   LAS ETIQUETAS SE INDICAN EN LA LISTA DE NODOS S , *
10550 Z      *
10560 Z      *****
10570 Z
10580 Z      INITIAL
10590 Z
10600      DO 10 I =1,N
10610      10 S(I) = 0
10620      S(SN) = 1
10630 Z
10640 Z      LABEL
10650 Z
10660      40 CONTINUE
10670      LAB = 0
10680      DO 20 K = 1,M
10690      I=D(K)
10700      J= T(K)
10710      IF ( S(I) ,GT. 0) GO TO 30
10720      IF ( S(J) ,EQ. 0) GO TO 20
10730      IF ( AD(-K) ,EQ. 0) GO TO 20
10740      S(I) = 1
10750      PB(I) =-K
10760      LAB = 1
10770      GO TO 20
10780 Z
10790      30 IF (S(J) ,GT. 0) GO TO 20
10800      IF ( AD(K) ,EQ. 0) GO TO 20
10810      S(J) = 1
10820      PB(J) = K
10830      LAB = 1
10840      20 CONTINUE
10850 Z
10860 Z      TEST
10870 Z
10880      IF ( S(TN) ,EQ. 1) GO TO 998
10890      IF ( LAB ,EQ. 1) GO TO 40
10900      NP = 1
10910      GO TO 999
10920 Z
10930      998 NP = 0
10940      999 CONTINUE
10950      RETURN
10960      END
10970
10980
10990
11000
11010

```

```

11020      INTEGER FUNCTION AD(K)
11030 %
11040      COMMON /D/F(100)/N/CL(100)/K/C(100)/A/D(100)/B/T(100)/P/PI(100)
11050      *      /L/H(100)
11060      INTEGER F,CL,C,O,T,D,PI
11070 %
11080 %      *****
11090 %      *
11100 %      * PROPOSITO :
11110 %      *      DETERMINAR LA ADMISIBILIDAD DE UN ARCO PA-
11120 %      *      RA EL ALGORITMO DE DESVIACIONES .
11130 %      *
11140 %      *****
11150 %
11160      AD = 1
11170      IF ( K .LT. 0) GO TO 1
11180      IF ( F(K) .GE. CL(K)) GO TO 2
11190      GO TO 999
11200 %
11210      1 IF ( F(-K) .GT. C(-K)) GO TO 999
11220      GO TO 4
11230 %
11240      2 I=0(K)
11250      J= T(K)
11260      D = PI(I) + H(K) -PI(J)
11270      IF ( D .GT. 0) GO TO 5
11280      IF ( F(K) .GE. C(K)) GO TO 5
11290      GO TO 999
11300 %
11310      4 I = 0(-K)
11320      J = T(-K)
11330      D= PI(I) + H(-K) - PI(J)
11340      IF ( D .LT. 0) GO TO 5
11350      IF ( F(-K) .GT. CL(-K) ) GO TO 999
11360      5 AD = 0
11370 %
11380      999 RETURN
11390      END
11400
11410
11420
11430
11440

```

```

11450 SUBROUTINE MFLO (LISA,IC,MF)
11460 X
11470 COMMON /A/D(100)/B/T(100)/P/PI(100)/L/H(100)/N/CL(100)/K/C(100)
11480 * /D/F(100)
11490 DIMENSION LISA(100)
11500 INTEGER D,T,PI,CL,C,F,D
11510 X
11520 X *****
11530 X * *
11540 X * PROPOSITO : *
11550 X * DETERMINAR EL MAXIMO CAMBIO DE FLUJO POSI- *
11560 X * EN LA TRAYECTORIA ENCONTRADA EN EL ALGORITMO FPATH3 . *
11570 X * *
11580 X *****
11590 X
11600 X
11610 X INITIAL
11620 X
11630 MF = 9999
11640 ILC = 0
11650 L=0
11660 KL=0
11670 X
11680 X TESTFIN
11690 X
11700 30 CONTINUE
11710 L=L+1
11720 IF (L .GT. IC) GO TO 999
11730 KK = LISA(L)
11740 IF ( KK .GT. 0) K=KK
11750 IF ( KK .LT. 0) K = -KK
11760 I=D(K)
11770 J=T(K)
11780 D=PI(I)-PI(J)+H(K)
11790 IF ( D .GT. 0) GO TO 51
11800 IF ( D .EQ. 0) GO TO 52
11810 GO TO 53
11820 X
11830 X ALPHA
11840 X
11850 51 CONTINUE
11860 IF ( F(K) .LE. CL(K) ) GO TO 10
11870 40 IF ( MF .GT. (F(K) -CL(K))) MF =F(K)-CL(K)
11880 GO TO 30
11890 X
11900 10 IF (MF .GT. (CL(K) -F(K))) MF=CL(K)-F(K)
11910 GO TO 30
11920 X
11930 X BETA
11940 X
11950 52 CONTINUE
11960 IF (F(K) .GT. C(K)) GO TO 40
11970 IF ( F(K) .GE. CL(K)) GO TO 50
11980 60 IF (MF .GT. (C(K)-F(K))) MF = C(K)- F(K)
11990 GO TO 30
12000 X
12010 X DIRECT
12020 X
12030 50 CONTINUE
12040 IF (KK .GT. 0) GO TO 60
12050 GO TO 40

```

```

12060 Z
12070 Z RHO
12080 Z
12090 Z 53 CONTINUE
12100 Z IF ( F(K) .LE. C(K)) GO TO 60
12110 Z IF ( MF .GT. (F(K) -C(K))) MF=F(K) -C(K)
12120 Z GO TO 30
12130 Z
12140 Z 999 CONTINUE
12150 Z RETURN
12160 Z END
12170 Z
12180 Z
12190 Z
12200 Z
12210 Z
12220 Z SUBROUTINE FLOCHG(LISA,IC,DEL)
12230 Z
12240 Z COMMON /O/F(100)
12250 Z DIMENSION LISA(100)
12260 Z INTEGER F
12270 Z
12280 Z *****
12290 Z *
12300 Z * PROPOSITO :
12310 Z * CAMBIAR LOS FLUJOS EN UNA TRAYECTORIA
12320 Z * ( LISA ) ,
12330 Z *
12340 Z *****
12350 Z
12360 Z DO 100 L=1,IC
12370 Z K=LISA(L)
12380 Z IF ( K .GT. 0 ) GO TO 98
12390 Z F(-K)=F(-K)-DEL
12400 Z GO TO 100
12410 Z
12420 Z 98 F(K) = F(K)+DEL
12430 Z 100 CONTINUE
12440 Z
12450 Z RETURN
12460 Z END
12470 Z
12480 Z
12490 Z
12500 Z

```

```

12510      SUBROUTINE OKAOUT
12520 %
12530      COMMON /G/H/N/P/PI(100)/K/C(100)/L/H(100)/N/CL(100)/D/F(100)
12540      COMMON /C/PB(100)/A/D(100)/B/T(100)
12550      INTEGER  Q,T,PB,CL,C,F,PI
12560 %
12570 % *****
12580 % * * * * *
12590 % * PROPOSITO : *
12600 % * IMPRINIR UNA SOLUCION OPTIMA . *
12610 % * * * * *
12620 % *****
12630 %
12640      PRINT 101, (I,PI(I),PB(I), I=1,N )
12650 %
12660 101 FORMAT(2(/),10X,'NODO',5X,'POTENCIAL',5X,'APUNTAOR HACIA ATRAS',
12670 * /, 12X,'I',6X,'PI(I)',15X,'PB(I)',2(/),(6X,I5,2X,I11,15X,I5) )
12680 %
12690      PRINT 103
12700 103 FORMAT(2(/),10X,'ARCO',4X,'N.ORIGEN',4X,'N.TERMINAL',5X,'COTA INF'
12710 * ',',5X,'COTA SUP.',6X,'COSTO',8X,'FLUJO',6X,'COSTO DEL FLUJO', /,
12720 * 12X,'K',7X,'O(K)',9X,'T(K)',9X,'CL(K)',9X,'C(K)',10X,'H(K)',9X,
12730 * 'F(K)', / )
12740 %
12750      DO 110 K=1,M
12760      COSTOK = F(K)*H(K)
12770      PRINT 106, K,O(K),T(K),CL(K),C(K),H(K),F(K),COSTOK
12780 106 FORMAT(8X,I5,6X,I5,8X,I5,3X,I11,2X,I11,3X,F11.2,2X,I11,2X,
12790 * F15.2 )
12800 110 CONTINUE
12810 %
12820      RETURN
12830      END
12840
12850
12860
12870
12880

```