

Tabla de Contenidos

CAPITULO I. 1

Grafos, Caminos y Circuitos

- I.1 Conceptos Básicos. 1
- I.2 Subgrafo de un Grafo. 2
- I.3 Matrices de Adyacencias y de Incidencia. 4
- I.4 Listas. 4
- I.5 Caminos, Cadenas, Circuitos y Ciclos. 5
- I.6 Caminos de Longitud Mínima. 6

CAPITULO II 11

Algoritmos de Rutas Mínimas

- II.1 Algoritmo de Bellman. 11
- II.2 Algoritmo de Dijkstra. 12
 - II.2.1 Justificación del Algoritmo de Dijkstra. 13
- II.3 Algoritmo General 14
- II.4 Algoritmo de Dantzig 15

CAPITULO III 16

Flujo Máximo

- III.1 Condiciones para hallar el Flujo Máximo en una Red. 16
- III.2 Fundamentos del Algoritmo de Ford & Fulkerson. 17
- III.3 Teorema del Corte Mínimo. 18
- III.4 mplementación del Algoritmo F & F para calcular Flujo Máximo. 20
- III.5 Justificación
- III.6 Extensión a Redes Canalizadas. 22

CAPITULO IV 23

Flujos Factibles

- IV.1 Condiciones de existencia de Flujos Factibles. 23
- IV.2 Teorema de Hoffman. 23
- IV.3 Algoritmo para encontrar un Flujo Factible. 24

CAPITULO V 26

Flujo de Costo Mínimo

- V.1 Problemas de Flujo de Costo Mínimo. 26
- V.2 Algoritmo para obtener el Flujo de Costo Mínimo. 29

CAPITULO VI 30

Flujo Máximo de Costo Mínimo

VI.1 Planteamiento. 30

VI.2 Primer Algoritmo (para el problema de Flujo Máximo de Costo Mínimo). 33

VI.3 Segundo Algoritmo (para el problema de Flujo Máximo de Costo Mínimo). 34

CAPITULO I

I.1 Conceptos Básicos.

Comenzaremos por definir formalmente un grafo, lo cual será necesario para poder definir una Red, tema central de este trabajo.

Definición: Un **Grafo** es un par (V, E) denotado $G(V,E)$, donde V es un conjunto finito, llamado conjunto de NODOS o VERTICES del grafo y E es una familia de pares $\{a,b\}$ de elementos de V , llamados LADOS del grafo. Siempre que los pares $\{a,b\}$ no sean ordenados se dice que el grafo es no-orientado o no dirigido y sus lados se llaman ARISTAS.

Cuando los pares $\{a,b\}$ son ordenados, se denotan (a,b) y se les llama ARCOS, en este caso se dice que el grafo es orientado o dirigido y se le llama también DIGRAFO.

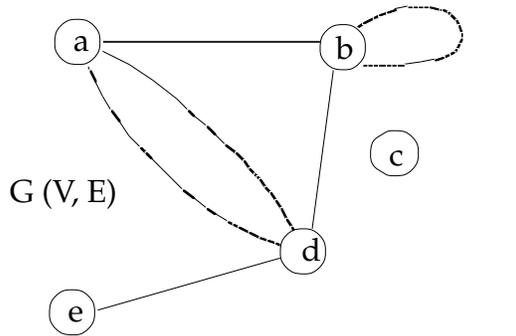
Cada uno de los nodos que conforman un lado del grafo se llaman EXTREMOS de ese lado. Si el grafo es dirigido u orientado entonces si (a, b) es un arco del grafo G , al nodo a se le llama su extremo INICIAL y al nodo b su extremo TERMINAL. Dos nodos que conforman un lado de un grafo se dice que son ADYACENTES, así mismo un lado $\{a,b\}$ es INCIDENTE a cada uno de sus extremos a y b . Igualmente dos lados que posean un extremo común se dice que son adyacentes.

Definición: Se llama VECINDAD o CONJUNTO DE ADYACENCIAS de un nodo x al conjunto de nodos adyacentes a x . Un nodo cuya vecindad es él mismo o el conjunto vacío se llama un NODO AISLADO. Un LAZO o BUCLE es un lado cuyos dos extremos son el mismo nodo.

Definición: Un grafo $G(V,E)$ que no posea lazos y tal que para ningún par de nodos existan lados iguales que los unan se llama GRAFO SIMPLE. Esto último nos lleva a pensar que pueden existir grafos con varios lados entre un mismo par de nodos, y en efecto, definimos E como una **familia** de pares, es decir que pueden haber varios lados iguales. Cuando esto ocurre al grafo se le llama también MULTIGRAFO. Se define entonces la MULTIPLICIDAD de un par (a,b) o $\{a,b\}$, según que el grafo sea o no dirigido, como el número de veces que ese par aparece en la familia E de lados del grafo. Un grafo simple sería entonces un grafo sin bucles y donde la multiplicidad de cada par es a lo sumo 1, o lo que es lo mismo, donde la familia E es un conjunto.

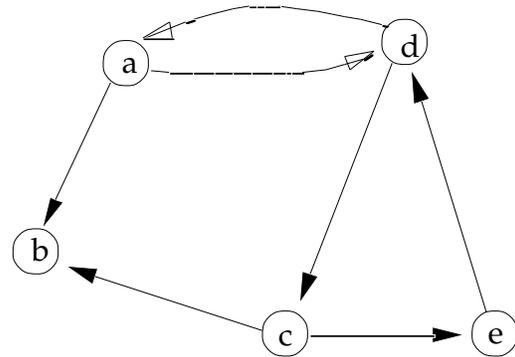
A cada grafo se le puede siempre asociar una figura o representación gráfica, la cual viene siendo la herramienta de mayor utilidad para estudiar propiedades del mismo. Esta figura es un dibujo en donde los nodos aparecen como puntos o círculos y cada lado se representa mediante una línea que une los dos puntos que corresponden a sus extremos. Cuando el grafo es dirigido los arcos se representan mediante una flecha que va del nodo inicial al nodo terminal (**Figura I.1**).

Figura I.1



$V = \{a, b, c, d, e\}$
 $E = \{\{a,b\}, \{a,d\}, \{a,d\}, \{d,e\}, \{b,d\}, \{b,b\}\}$

c : nodo aislado; $\{b,b\}$: lazo o bucle;
 multiplicidad de $\{a,d\} = 2$.

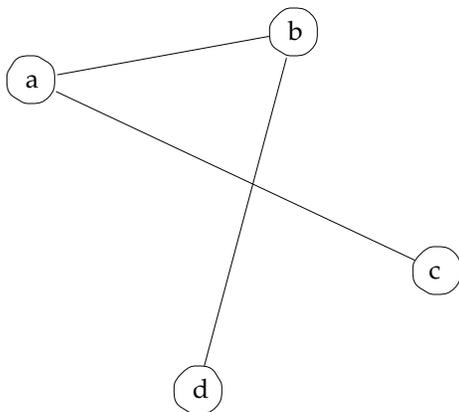


Digrafo Simple

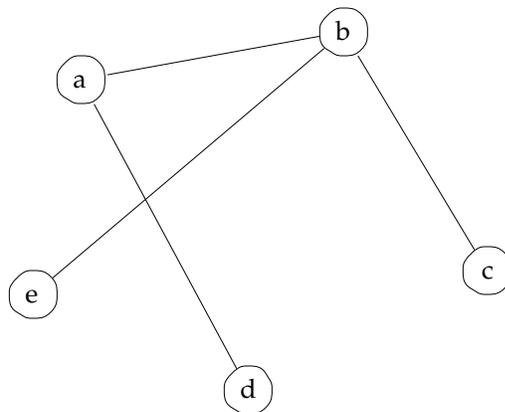
I.2 Subgrafo de un Grafo.

Definición: Un grafo $H(W,F)$ es un subgrafo de $G(V,E)$ si $V \dots W$, $E \dots F$ y los extremos de los lados en F están en W . Si $W=V$ se dice que el subgrafo H expande a G , y se le llama también GRAFO PARCIAL.

Figura I.2



$H(W,F)$ Subgrafo de $G(V,E)$



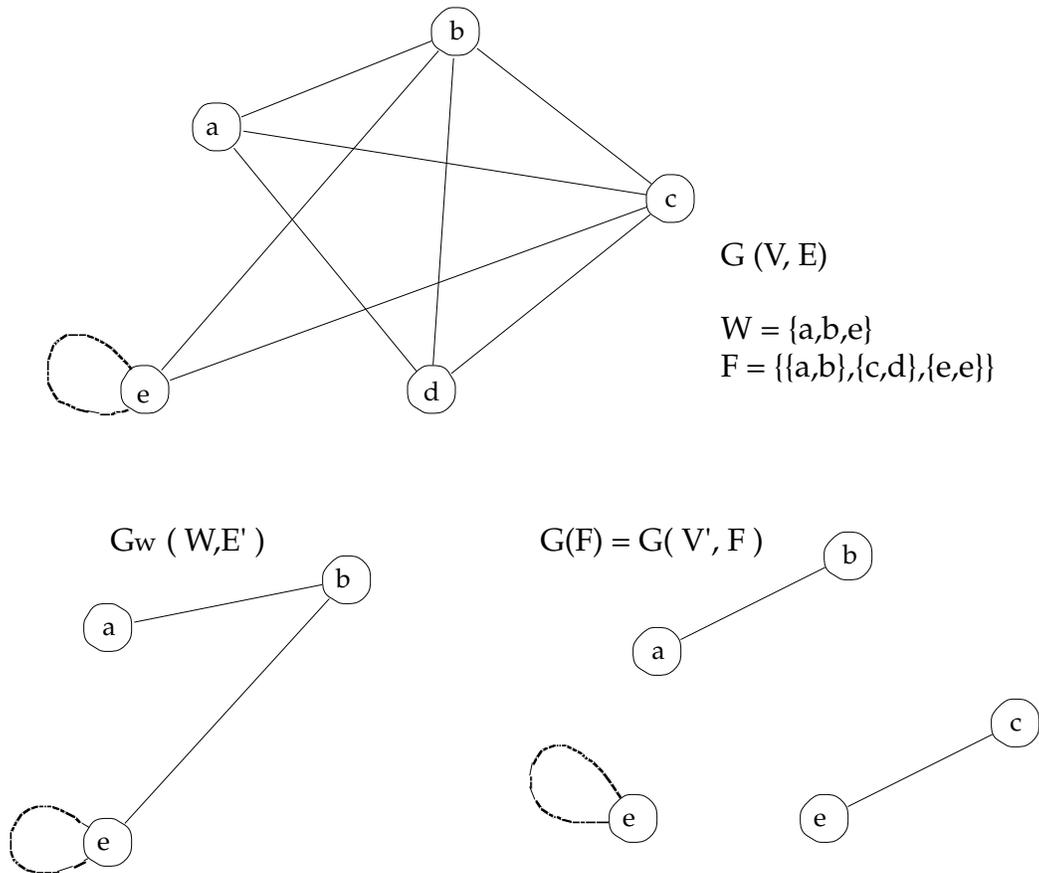
$H(V,F)$ Subgrafo Parcial de $G(V,E)$

Sean W y F subconjuntos de V y E respectivamente en $G(V,E)$, podemos definir también los subgrafos ENGENDRADOS o INDUCIDOS por los subconjuntos W y F de la siguiente manera:

Definición: El subgrafo engendrado por un subconjunto de nodos W de V es el subgrafo $G_W(W, E')$ de $G(V,E)$ donde E' es el subconjunto de lados tales que ambos extremos están en W .

Definición: El subgrafo engendrado por un subconjunto de lados F de E es el subgrafo $G(V',F)$ de $G(V,E)$ donde V' es el subconjunto de nodos que son extremos de los lados en F . Se denota también $G(F)$.

Figura I.3



I.3 Matrices de Adyacencias y de Incidencia.

Las matrices proporcionan una forma de representar grafos. Este tipo de representación es de gran utilidad sobre todo para efectos computacionales.

Definición: La **Matriz de Adyacencias** asociada a un grafo no orientado $G(V,E)$ es una matriz **MA** donde las filas y las columnas representan elementos de V . Una casilla de la matriz tiene el valor 1 si los elementos representados por la fila y la columna correspondientes a esa casilla son adyacentes en G , y tendrá el valor cero en caso contrario.

Toda matriz de adyacencias de un grafo no orientado es simétrica, y si el grafo no tiene bucles, todos los elementos de la diagonal principal son cero.

En el caso de grafos dirigidos u orientados existe otra forma matricial de representación asociada a la relación binaria E tomada como pares ordenados de $V \times V$. A esta representación se le da también el nombre de **Matriz de Adyacencias del Digrafo**; es importante sin embargo no confundirla con la matriz de adyacencias de grafos no dirigidos definida anteriormente. En esta representación, las filas y las columnas de nuevo corresponden a los elementos de V . Una entrada (i,j) de esta matriz tendrá el valor **1** (uno) si existe en E un arco cuyo extremo inicial sea el verice de la fila i y cuyo extremo terminal sea el verice de la columna j , y **0** (cero) en caso contrario. Esta matriz NO siempre será simétrica.

Definición: La **Matriz de Incidencia** es una matriz **MI** nodos - arcos, es decir, donde las filas representan los nodos del grafo y las columnas los lados del mismo. Si el grafo es no orientado, entonces las entradas (i,j) de la matriz tendrán el valor **1** (uno) cada vez que el nodo de la fila i sea extremo del lado de la columna j , y **0** (cero) en caso contrario.

Si el grafo es dirigido, entonces las entradas de la matriz serán **1, -1 y 0**; 1 si el nodo de la fila i es el extremo terminal del arco de la columna j ; -1 si el nodo de la fila i es inicial del arco de la columna j ; y 0 si el nodo de la fila i no es incidente al arco de la columna j .

En general, la forma de representación o almacenamiento de grafos depende de la naturaleza de los datos y de las operaciones que se van a llevar a cabo. Aparte de esto, la mejor representación también se ve afectada por otros factores, tales como el número de nodos, densidad o número de lados, si el grafo es o no dirigido, si será necesario agregar o eliminar otros nodos o lados, etc.

En general la representación en forma de matriz de adyacencias o incidencias, almacenadas como arreglos, presenta varias desventajas, entre otras:

- 1) Si el grafo tiene un número pequeño de lados ($|E| \ll n^2$) entonces la matriz será "esparcida", es decir, con un gran número de ceros, lo que equivale a desperdiciar memoria;
- 2) Esta forma de representación al usar arreglos requiere de una asignación estática de memoria, por lo tanto si se desconoce el tamaño inicial del grafo o si se requiere agregar nodos durante el proceso, es necesario declarar los arreglos con más espacio del realmente requerido para poder realizar estos cambios;

- 3) Si se debe almacenar información acerca de los nodos será necesario utilizar estructuras adicionales, igualmente si se tiene información sobre los arcos, esta quizá se podría sustituir en los unos y ceros de la matriz, pero entonces la matriz ya no podría ser booleana.

Definición: Las listas enlazadas ofrecen una forma alternativa para almacenar grafos. La más común de estas formas de almacenamientos es la que representa al grafo por sus **Listas de Adyacencias**. De esta forma tenemos una lista para cada nodo, en la cual aparecen todos los demás nodos que le son adyacentes. Así, para grafos dirigidos, el nodo cabeza de la lista será el nodo inicial de la secuencia de arcos donde los nodos terminales son los demás nodos de la misma lista.

Esta forma de representación permite llevar, utilizando la misma estructura, información sobre los arcos. Una desventaja para los grafos no-orientados es que cada arista aparecería repetida en las dos listas correspondientes a sus dos extremos. Si además se debe manejar información sobre los nodos, entonces generalmente se tiene una lista con los nodos y la información respectiva, y las listas de adyacencias se hacen con apuntadores a los nodos de la lista de nodos.

Definición: Otra forma alternativa, también utilizando listas enlazadas, es la representación como **Lista de Lados**. Esta consiste en una lista de pares $\{a,b\}$ de nodos, uno por cada lado del grafo. Si el grafo es dirigido entonces el primer nodo a representa el nodo inicial y el segundo nodo b , el nodo terminal. Si el grafo no es dirigido no es necesario imponer ningún orden sobre los pares.

Esta última forma de representación permite almacenar, utilizando la misma estructura, información adicional sobre los lados. Sin embargo, si fuera necesario manejar información sobre los nodos, se requerirían estructuras adicionales.

Veremos un pequeño ejemplo para ilustrar la importancia de escoger la estructura correcta. Se define **Orden** de un grafo como la cardinalidad del conjunto de nodos. Supongamos que se desea determinar si un grafo de orden n tiene al menos un lado. Si el grafo está representado por su matriz de adyacencias, es necesario revisar la matriz hasta encontrar un **1**, lo que puede llevar en el peor de los casos n^2 comparaciones. Si el grafo se tiene como lista de adyacencias, será necesario examinar cada nodo hasta encontrar una lista no-vacía, en el peor de los casos n comparaciones, una por cada nodo. Si por el contrario tenemos el grafo representado por la lista de arcos, basta hacer **1** comparación para determinar si la lista es no vacía.

Una de las aplicaciones de teoría de Grafos es la determinación de caminos o itinerarios en una red de transporte, distribución de productos, itinerarios de viaje, etc..

Por ejemplo, en el problema del agente viajero se desea visitar un grupo de ciudades sin duplicaciones, ¿Cuál sería el itinerario a seguir, dejando abierta la posibilidad de que existan ciudades que no se comunican? Podemos modelar la región que se desea visitar mediante un grafo donde los vértices son las ciudades y las aristas representan las vías de comunicación existentes.

Estos problemas y muchos otros se pueden plantear formalmente en terminos de cadenas, caminos, ciclos o circuitos de un grafo. Daremos a continuación las definiciones de cada uno de estos conceptos.

Definición: Una **Cadena** entre dos nodos X_0, X_n es una secuencia alternada de nodos y **lados** $(X_0, e_1, X_1, \dots,$

X_{n-1}, e_n, X_n) tal que para todo $i=1, \dots, n$ las extremidades de e_i son X_{i-1} y X_i .

Este concepto es válido tanto para grafos no orientados, como para grafos orientados, donde los lados serían los arcos tomados independientemente de su orientación.

Definición: Un **Camino** entre dos nodos X_0, X_n es una secuencia alternada de nodos y arcos $(X_0, e_1, X_1, \dots, X_{n-1}, e_n, X_n)$ tal que para todo $i=1, \dots, n$ el extremo inicial de e_i es X_{i-1} y el extremo terminal es X_i .

Cada una de las secuencias que definen caminos o cadenas podrían contener solo un nodo, es decir, que no incluyen ningún lado, en cuyo caso nos encontramos con caminos o cadenas triviales de un nodo a si mismo, donde los nodos inicial y terminal coinciden.

Definición: Una Cadena o Camino se llama **Simple** si ninguno de los lados que la conforman aparece más de una vez en la secuencia.

Definición: Una Cadena o Camino se llama **Elemental** si cada vértice aparece en la secuencia que lo define a lo sumo una vez.

Definición: Un **Ciclo** es un Cadena Simple donde los nodos inicial y final coinciden, es decir, $X_0 = X_n$. Un Ciclo es una Cadena Simple circular o cerrada.

Definición: Un **Circuito** es un Camino Simple donde los nodos inicial y final coinciden, i.e., $X_0 = X_n$. Un Circuito es un Camino Simple circular o cerrado. Un Ciclo o Circuito es Elemental si todos los vertices distintos de X_0 y X_n aparecen a lo sumo una vez en la secuencia que lo define.

Como vemos, los conceptos de Camino y Circuito dependen de la orientación de los lados o arcos y por lo tanto solo tienen sentido para digrafos. Sin embargo los conceptos de Cadena y Ciclo son válidos en ambos casos pues en digrafos podemos considerar los arcos sin tomar en cuenta el orden de sus elementos.

Definición: Otros conceptos que pueden ser de utilidad son los de **Pseudo-Ciclo** y **Pseudo-Circuito**, es decir, una cadena, o respectivamente camino, donde los extremos inicial y final coinciden. Al levantar la exigencia de que la cadena o camino sea simple, en estos casos se permiten lados o arcos repetidos en la secuencia, incluso consecutivamente.

En un grafo, la **Longitud** de un camino, cadena, ciclo o circuito se define como el número de lados que conforman la secuencia que lo conforma.

Con todos estos conceptos a la mano, definiremos finalmente lo que llamamos una **RED**.

Definición: Una **Red**, la cual denotaremos $R(V, E, f_1, \dots, f_m)$ es una tupla donde V y E son los nodos y lados de un grafo $G(V, E)$ llamado el "grafo subyacente" y donde f_1, \dots, f_m son funciones $f_i: E \rightarrow \mathbf{R}$ que pueden representar magnitudes tales como distancia o longitud, costo, capacidad, etc.

Consideremos una red $R(V, E, d)$ donde la función d la llamaremos distancia. Mientras no se especifique lo contrario, supondremos que el grafo subyacente $G(V, E)$ es un grafo orientado.

Dados dos nodos x, y en V , denotaremos por C_x^y un camino de el nodo x al nodo y . Definimos la **Longitud** de un camino (cadena) de x a y en una red $R(V,E,d)$ como la suma de las distancias de los arcos (aristas) que lo conforman. Si el camino (cadena) no es simple, cada arco (arista) será contado tantas veces como aparezca en la secuencia que conforma el camino (cadena).

Definición: Un camino (cadena) que no incluya ningún arco, es decir, definido por una secuencia vacía, tendrá longitud nula.

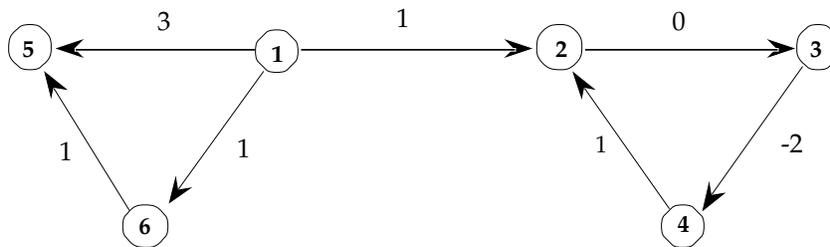
Definición: Diremos que un grafo o red es **Conexo** si, \forall par de nodos $x, y \in V, \exists$ una cadena que los une.

Definición: Un digrafo o red es **Fuertemente Conexa** si, \forall par de nodos $x, y \in V, \exists$ caminos C_x^y, C_y^x .

Tomemos ahora dos nodos cualesquiera x e y en V . Si estudiamos los caminos de x a y , vemos que pueden presentarse tres casos (**Fig. I.4**):

- i) No existe camino de x a y en $R(V,E,d)$.
- ii) Existen caminos de x a y pero no podemos encontrar un camino que tenga mínima longitud.
- iii) Existe un camino de longitud mínima de x a y .

Figura I.4



Definición: Dados dos nodos x, y en $R(V,E,d)$, si existe un camino más corto (ruta mínima) de x a y , entonces la longitud de ese camino se denomina "**Distancia mínima** de x a y ". Así, la distancia mínima de un nodo a sí mismo si existe es nula.

Plantaremos a continuación seis problemas relacionados con caminos en redes que serán motivo de nuestro estudio.

- A. Hallar un camino elemental de x a y en $R(V,E,d)$ de longitud mínima.
- B. Dado un nodo s en $R(V,E,d)$, hallar un camino elemental C_s^x para todo otro nodo x en V .
- C. Para todo par de nodos x, y en $R(V,E,d)$, hallar un camino elemental C_x^y de longitud mínima.

Si levantamos en cada uno de los anteriores la exigencia de que los caminos sean elementales, obtenemos los otros tres problemas A', B', C'.

Definición: Un circuito en $R(V,E,d)$ se dice **Absorbente** si su longitud es negativa. Como ejemplo, se puede considerar el circuito $C = ((2,3) (3,4) (4,2))$ de la Figura 3.

Será fácil observar que siempre que el problema A' tenga solución, su análogo A también la tendrá. Igualmente ocurre con B' , B y C' , C . Veremos a continuación tres teoremas que nos permitirán garantizar soluciones a A' , B' y C' .

Teorema I.1.A'.

Dados dos nodos x, y en $R(V, E, d)$, entonces existirá una ruta mínima de x a y en R ssi:

- i) El conjunto S de nodos que son a la vez antecesores de y y sucesores de x es no-vacío.
- ii) La sub-red $R_S(S, E', d)$ engendrada por el conjunto S no contiene ningún circuito absorbente.

Teorema I.1.B'.

Dado un nodo s en $R(V, E, d)$, existirá una ruta mínima C_s^x para todo $x \neq s$ en $R(V, E, d)$ ssi:

- i) El nodo s es una raíz de $R(V, E, d)$.
- ii) $R(V, E, d)$ no tiene circuitos absorbentes.

Teorema I.1.C'.

Para todo par de nodos x, y en $R(V, E, d)$ existirá ruta mínima de x a y ssi:

- i) $R(V, E, d)$ es fuertemente conexa.
- ii) $R(V, E, d)$ no contiene circuitos absorbentes.

Los tres teoremas anteriores son muy similares. La i-ésima condición garantiza la existencia de caminos entre los nodos indicados y la ii-ésima condición asegura que las longitudes de estos caminos estén acotadas inferiormente. Es interesante observar, que en el caso de que existan circuitos de longitud nula, los conjuntos de soluciones de A' , B' , C' pueden ser mayores que los de A , B y C . Sin embargo, en lo sucesivo, siempre que no se especifique lo contrario, los caminos de longitud mínima serán considerados elementales.

Demostraremos a continuación el teorema 1A'. Las demostraciones de 1B' y 1C' son similares y se dejan como ejercicio para el lector.

Demostración I.1.A':

\implies) Es evidente. La existencia de un camino mínimo C_x^y implica que el conjunto S es no-vacío, y si existiese algún circuito absorbente en R_S entonces este circuito tiene al menos un nodo z . Tomemos el camino C'' formado por la unión del circuito C con las subsecuencias de $C_x^y: C_x^z, C_z^y$, es decir $C'' = C_x^z \cup C \cup C_z^y$, entonces la longitud de C'' es menor que la de C_x^y , lo cual contradice la hipótesis de que C_x^y es una ruta mínima.

\Leftarrow) La condición i) implica que existen caminos de x a y , por lo tanto podemos encontrar un camino elemental mínimo C_x^y . Si este camino elemental no fuese la ruta mínima de x a y , entonces podemos encontrar otros caminos no-elementales pero de longitud inferior a la de C_x^y . Sea C' un camino tal con número mínimo de arcos. Así $l(C') < l(C_x^y)$.

Puesto que C' no es elemental, podemos encontrar un nodo p tal que C' pasa dos veces por él. Sea \forall la sub-

secuencia de C' entre dos "pasadas" por p . \forall es un circuito y entonces $C'' = C' - \forall$ es también un camino de x a y que tiene menos arcos que C' , por lo tanto, por la forma como se escogió C' , C'' debe ser elemental y además: $l(C'') \geq l(C) > l(C')$.

De esto último tenemos que $l(\forall) = l(C') - l(C'') < 0$ de donde \forall sería un circuito absorbente lo cual contradice la hipótesis ii). *lqqd.*

Siendo una red $R(V,E,d)$ finita, existirá un número finito de caminos elementales entre un par de nodos dado. Según esto, en principio, los problemas A, B, C son problemas de optimización combinatoria que no parecieran ser de gran dificultad, bastaría enumerar los caminos y escoger el más corto. Sin embargo, esto puede hacerse muy tedioso y largo a medida que aumenta el número de nodos del grafo, por ejemplo, para un grafo completo (K_{n+1}) con $n+1$ nodos el número de caminos que unen 2 (dos) nodos es del orden de $n!$.

Por otra parte, a pesar de que el número de caminos no-elementales es en general mucho mayor, existen algoritmos muy eficientes y sencillos para resolver A' , B' y C' . En cambio, cuando estos últimos problemas no tienen solución, es muy difícil resolver sus análogos A, B, C. Más aún, se afirma que no pueden existir algoritmos eficientes que los resuelvan, lo que se llama problemas NP-completos.

Conviene señalar además un fenómeno aparentemente paradójico: a pesar de que el problema B' (dado s hallar C_s^x mínimo para todo x en $R(V,E,d)$) se puede expresar como $n-1$ problemas del tipo A' (dados x , y en $R(V,E,d)$, hallar C_x^y mínimo), siempre que se desea resolver A' , se pasa para ello por la resolución de B' .

Teorema I.2.

Sea C un camino más corto de s a p en $R(V,E,d)$. Sean x , y dos nodos en el camino y C_x^y la porción de C situada entre ellos. Entonces C_x^y es la ruta mínima de x a y .

Demostración:

Supongamos que existe otro camino C'' de x a y tal que $l(C'') < l(C_x^y)$, entonces $C_s^x \cup C'' \cup C_x^p$ es otro camino de s a p y su longitud $l(C_s^x) + l(C'') + l(C_x^p) < l(C_s^x) + l(C_x^y) + l(C_x^p) = l(C)$, lo cual contradice que C sea la ruta mínima de s a p .

Definición:

Se llama **Raíz** de una red o grafo a un nodo s tal que existen caminos desde s hasta todos los demás nodos de la red.

Definamos a continuación dos funciones $I, T : E \rightarrow V$, las cuales asocian a cada arco sus nodos Inicial y Terminal respectivamente.

Sea además s una raíz de $R(V,E,d)$ sin circuitos absorbentes, definimos entonces otra función $\pi: V \rightarrow \mathbf{R}$ que asocia a cada nodo x la distancia mínima de s a x .

Teorema I.3.

Sea $R(V,E,d)$ sin circuitos y con una raíz s , entonces:

$$\forall e \in E: \pi(T(e)) - \pi(I(e)) \leq d(e). \quad (*)$$

Demostración:

Sea $e=(x,y)$, entonces $C = C_s^x \cup (x,y)$ donde C_s^x es una ruta mínima, es decir, un camino de longitud mínima de s a y , de donde:

$$l(C) = \pi(x) + d(e) \geq \pi(y). \quad lqgd.$$

Teorema I.4.

Sea $R(V,E,d)$ como antes, entonces la sub-red $R(E')$, donde

$$E' = \{e \in E / \pi(T(e)) - \pi(I(e)) = d(e)\}$$

también tiene raíz s , es decir, todas las rutas mínimas en $R(V,E,d)$ están en $R(E')$.

Recíprocamente, todo camino de s a x en $R(E')$ es una ruta mínima.

Demostración:

\implies) Sea C_s^z la ruta mínima de s a z , $e=(x,y)$ un arco de esta ruta mínima. Entonces las subsecuencias C_s^x , C_s^y son rutas mínimas a x e y respectivamente. Así: $l(C_s^y) = \pi(y) = l(C_s^x) + d(e) = \pi(x) + d(e)$, de donde vemos que todo arco perteneciente a un camino mínimo de s a cualquier otro nodo x está en E' .

\Leftarrow) Sea C un camino cualquiera de s a cualquier nodo x en $R(E')$, veamos que este es una ruta mínima.

$$l(C) = \sum_{\varepsilon \in X} \delta(\varepsilon) = \sum_{\varepsilon \in X} (\pi(T(\varepsilon)) - \pi(I(\varepsilon))) = \pi(x) - \pi(s) = \pi(x) - \pi(s)$$

luego siendo la longitud de C igual a la distancia mínima de s a x , C es una ruta mínima.

Cabe observar que luego de esta demostración podemos caracterizar todos los arcos sobre una ruta mínima verifican la desigualdad (*) del **Teorema 3** como una igualdad.

Este teorema muestra también que la búsqueda del conjunto de rutas mínimas que salen de un nodo s hacia los demás nodos en una red $R(V,E,d)$ puede remitirse a la búsqueda de las distancias más cortas. Luego, es muy fácil construir la sub-red $R(E')$ en la cual todo camino desde s es una ruta mínima.

Corolario. Todo circuito en $R(E')$ tiene longitud nula.

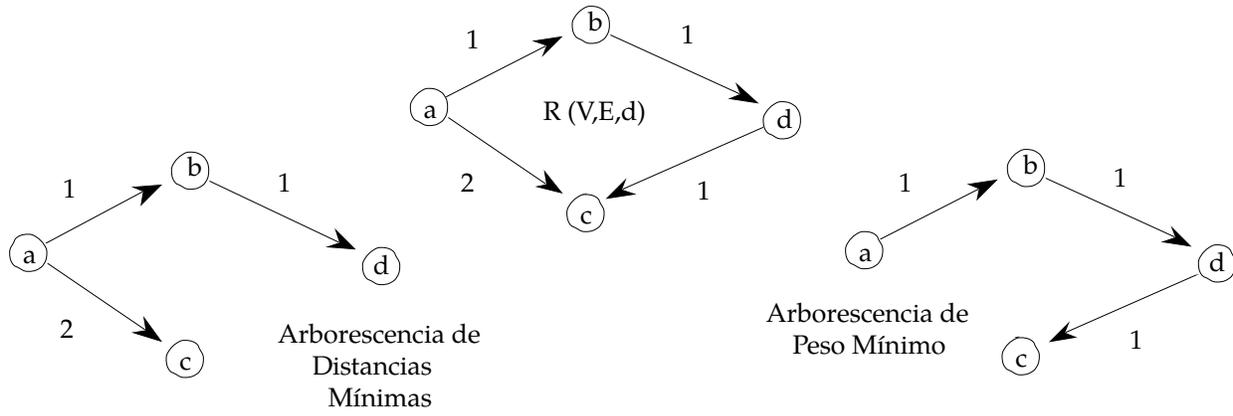
Definición: un **árbol** es un grafo conexo sin ciclos, y una **arborescencia** es un árbol enraizado, es decir, con una raíz.

Podemos afirmar entonces que el grafo subyacente de la red $R(E')$ contiene una arborescencia expandida de raíz s . Para obtenerla, basta eliminar ciertos arcos hasta que el número de arcos que llegan a cada nodo, excepto a s , sea igual a 1. Esta arborescencia permite encontrar una única ruta mínima de s a x y se le llama la "**Arborescencia de Distancias Mínimas**".

Es importante notar que la arborescencia de distancias mínimas **NO** es la misma que conocemos con el

nombre de "arborescencia de peso mínimo" definida como aquella donde la suma de las distancias de los arcos que la conforman es mínima.

Figura I.5



CAPITULO II

Algoritmos de Búsqueda de Caminos más Cortos.

Siendo el problema de búsqueda de Caminos más Cortos o Rutas Mínimas uno de los problemas de mayor aplicación en Optimización en Redes, existen actualmente más de 150 algoritmos para resolverlo, en particular ocurre así para el problema B' que mencionamos en el capítulo anterior, búsqueda de caminos más cortos desde un nodo hasta todos los demás de la red.

Veremos a continuación solo algunos de ellos escogidos por su eficiencia o por tener un mayor interés teórico. Además veremos que cada uno de los que estudiaremos se adapta a ciertas condiciones tales como:

- Redes sin circuitos.
- Redes con distancias positivas.
- Redes con posibles circuitos absorbentes.
- Se desea determinar la distancia mínima entre todo par de nodos.

Los algoritmos que veremos son tales que podrán luego ser utilizados en la resolución de problemas de Flujo en Redes.

II.1 Algoritmo de Bellman.

(Redes sin circuitos)

El principio de este algoritmo es buscar la distancia mínima desde un nodo s hasta los demás nodos de la red, en forma expansiva o exhaustiva, es decir, $\pi(x)$ (distancia mínima de s a x) no se conocerá hasta conocer $\pi(y)$ para todo y antecesor de x .

Algoritmo BELLMAN:

Entradas: V, E, I, T, d, s

Salidas: π, A, EsRaiz

- (* EsRaiz es una variable booleana que indica si el nodo s es raíz de $R(V,E,d)$
 A es un arreglo con el antecesor de cada nodo en el camino más corto.
 \hat{e} es al arco que corresponde al mínimo *)

Comienzo

$S \leftarrow \{s\};$

$\pi[s] \leftarrow 0;$

$A[s] \leftarrow -1;$

$\text{EsRaiz} \leftarrow \text{verdadero};$

mientras ($\exists x \notin S / \forall y$ antecesor de $x, y \in S$) **hacer**

$\pi[x] \leftarrow \text{Min}_{e/T(e)=x} \{ \pi[I(e)] + d(e) \} = \pi[I(\hat{e})] + d(\hat{e});$

$A[x] \leftarrow I(\hat{e});$

$S \leftarrow S \cup \{x\};$

fin mientras.

$\text{EsRaiz} \leftarrow (S = V);$

Fin.

Suponiendo que no hay circuitos, el algoritmo termina en un número finito de pasos. Si s no es raíz de G , $S \neq V$, de lo contrario el ciclo del **mientras** recorre todo V llegando a cada nodo de la red por la vía más corta.

El arreglo A nos proporciona además los arcos $(A[x],x)$ pertenecientes al árbol de raíz s , es decir, la arborescencia de distancias mínimas desde s . La ruta mínima desde s hasta cualquier nodo x se obtiene recorriendo el arreglo A en sentido inverso desde x hasta s .

II.2 Algoritmo de Dijkstra.

(Redes con distancias positivas)

Como en el algoritmo de Bellman, este algoritmo mantiene un conjunto de nodos S para los cuales ya se ha determinado la distancia más corta desde s . Este conjunto S aumenta en un nodo por iteración. La particularidad de este algoritmo es que los nodos se agregan a S en orden creciente de distancia al nodo s .

Se utilizará una variable **xpivot** para indicar el último nodo que entró al conjunto S .

Algoritmo DIJKSTRA:

Entradas: V, E, I, T, d, s

Salidas: π, A, EsRaiz

Comienzo

$S \leftarrow \{s\};$

$xpivot \leftarrow s;$

$\pi[s] \leftarrow 0;$

$A[s] \leftarrow -1;$

$\text{EsRaiz} \leftarrow \text{verdadero};$

para todo $\{x \in V / x \neq s\}$ **hacer**

$\pi[x] \leftarrow \infty;$

$A[x] \leftarrow -1;$

fin para;

mientras $(S \neq V) \wedge (\pi[xpivot] < \infty)$ **hacer**

para todo $\{y \notin S / (xpivot, y) \in E\}$ **hacer**

$\text{Suma} \leftarrow \pi[xpivot] + d(xpivot, y);$

si $(\pi[y] > \text{Suma})$ **entonces**

$\pi[y] \leftarrow \text{Suma};$

$A[y] \leftarrow xpivot;$

fin si;

fin para;

$xpivot \leftarrow x / \pi[x] = \text{Min}_{y \notin S} \{\pi[y]\};$

$S \leftarrow S \cup \{xpivot\};$

fin mientras;

$\text{EsRaiz} \leftarrow (\pi[xpivot] < \infty)$

Fin.

II.2.1 Justificación del Algoritmo de Dijkstra.

Esta justificación se hace necesaria pues sería natural pensar que si aún no conocemos la distancia más corta a todos los antecesores de x_{pivot} al momento que lo incluimos en S , podría ocurrir que existiera un camino más corto pasando por alguno de sus antecesores que no están aún en S . Sin embargo, siendo $d(e) > 0, \forall e \in E$, esto no ocurrirá jamás. Veremos a continuación por qué.

Sea C_S el conjunto de todos los caminos que tienen todos sus nodos en S excepto quizá el último. Estos caminos se llaman a veces "caminos especiales".

El algoritmo busca en cada iteración un camino en C_S para todos los nodos $x \notin S$ adyacentes a x_{pivot} , y luego toma aquel nodo cuyo camino (obviamente en C_S , pues sino $\pi(x) = \text{infinito}$) sea el más corto para incluirlo en S .

A menos que se demuestre lo contrario podemos suponer que existe otro nodo $w \notin S$ tq. la longitud de $C = C_S^w \cup C_{wv}^y = l(C)$ sea menor que $\pi(y)$ una vez que y ha sido incluido en S . Este camino pasando por w sería un camino no-especial.

Podemos suponer, sin pérdida de generalidad, que y es el último nodo que entró en S , i.e. x_{pivot} . Sea v el primer nodo en C_S^w fuera de S (v podría ser el mismo w), obviamente $\pi(v) \geq \pi(y)$ pues de lo contrario v hubiese entrado a S en lugar de y . Así, la longitud del camino C :

$$l(C) = l(C_S^v) + l(C_{vw}^w) + l(C_{wv}^y) \geq \pi(v) + l(C_{vw}^w) + l(C_{wv}^y) \geq \pi(y). \quad \ddagger$$

II.3 Algoritmo General

(Redes de cualquier tipo)

Este algoritmo más general permite detectar circuitos absorbentes si los hay, y en caso contrario encuentra la arborescencia de distancias mínimas de raíz s .

El algoritmo encuentra, utilizando el algoritmo de Dijkstra, una arborescencia inicial A de raíz s , no necesariamente de distancias mínimas, y luego trata de mejorar A hasta que sea óptima, o hasta detectar un circuito absorbente.

Para mejorar la arborescencia inicial, el algoritmo se basa en los teoremas 3 y 4 del capítulo I.

Algoritmo RUTAMINGENERAL:

Entradas: V, E, I, T, d, s
Salidas: $\pi, A, \text{EsRaiz}, \text{CircuitAbs}, C$

(* EsRaiz y CircuitAbs son variables booleanas que indican si s es o no raíz y si existe o no un circuito absorbente. En caso que ocurra esto último, C sería tal circuito *)

Comienzo

```

DIJKSTRA (V,E,I,T,d,s;  $\pi$ ,A,EsRaiz);
CircuitAbs  $\leftarrow$  falso;
si (EsRaiz) entonces
  E'  $\leftarrow$  {e $\in$ E / e = (A[x],x), x $\in$ V};
  mientras ( { $\exists \hat{e} \in E-E'$  / d( $\hat{e}$ ) <  $\pi$ [T( $\hat{e}$ )] -  $\pi$ [I( $\hat{e}$ )] } )  $\wedge$  ( $\neg$  CircuitAbs) hacer
    si (G(V,E'  $\cup$  { $\hat{e}$ }) contiene un circuito C ) entonces
      CircuitAbs  $\leftarrow$  verdadero
    sino
      x  $\leftarrow$  T( $\hat{e}$ );
      E'  $\leftarrow$  E'  $\cup$  { $\hat{e}$ } - { (A[x],x) };
      A[x]  $\leftarrow$  I( $\hat{e}$ );
       $\delta$   $\leftarrow$   $\pi$ [T( $\hat{e}$ )] -  $\pi$ [I( $\hat{e}$ )] - d( $\hat{e}$ );
       $\pi$ [x]  $\leftarrow$   $\pi$ [x] -  $\delta$ ;
      para todo (y sucesor de x en el árbol A) hacer
         $\pi$ [y]  $\leftarrow$   $\pi$ [y] -  $\delta$ ;
      fin para;
    fin si;
  fin mientras;
fin si;

```

Fin.

Se observa que a cada vuelta del algoritmo $\sum_{x \in V} \pi[x]$ decrece estrictamente, así que no se vuelve nunca a una arborescencia hallada anteriormente. Siendo finito el número de arborescencias de $R(V,E,d)$, el algoritmo es

finito.

Para determinar si existe circuito absorbente basta mirar si $T(\hat{e})$ es antecesor de $I(\hat{e})$ en A , donde \hat{e} es el arco que se está agregando en ese momento. Por ser A una arborescencia, esto último es muy sencillo pues existe un único camino de s a x en A y basta recorrerlo en sentido inverso.

II.4 Algoritmo de Dantzig

Este algoritmo calcula las distancias entre todo par x, y en V . Las arborescencias correspondientes a los caminos más cortos pueden deducirse fácilmente construyendo a partir de las distancias, las arborescencias de distancias mínimas desde cada uno de los nodos de la red. Se trabaja con una matriz para las distancias entre cada par de nodos x, y . Si $|V| = n$ es tal que una matriz $n \times n$ puede ser almacenada sin problemas, es preferible utilizar el algoritmo de Dantzig.

Para comenzar el algoritmo es necesario numerar los nodos, X_1, X_2, \dots, X_n . Una vez hecho esto, el algoritmo va construyendo las matrices D^k de dimensión $k \times k$ donde $D_{i,j}^k$ representa la distancia mínima de X_i a X_j en la subred construida sobre los primeros k nodos.

Algoritmo DANTZIG: (* Podemos suponer que $R(V,E,d)$ es simple y sin lazos *)

Entradas: V, E, d

Salidas: D^n

Comienzo

$D_{1,1}^1 \leftarrow d(X_1, X_1) = 0;$

$k \leftarrow 1;$

CircuitAbs \leftarrow falso;

mientras $(k < n) \wedge (\neg \text{CircuitAbs})$ **hacer**

para todo $\{ i=1..k \}$ **hacer**

$D_{k+1,i}^{k+1} \leftarrow \text{Min}_{j=1..k} [D_{j,i}^k + d(X_{k+1}, X_j)];$

$D_{i,k+1}^{k+1} \leftarrow \text{Min}_{j=1..k} [D_{i,j}^k + d(X_j, X_{k+1})];$

si $((D_{i,k+1}^{k+1} + D_{k+1,i}^{k+1}) < 0)$ **entonces**

 CircuitAbs \leftarrow verdadero

sino

$D_{k+1,k+1}^{k+1} \leftarrow 0;$

fin si;

fin para;

si $(\neg \text{CircuitAbs})$ **entonces**

para todo $\{ i=1..k \}$ **hacer**

para todo $\{ j=1..k \}$ **hacer**

$D_{i,j}^{k+1} \leftarrow \text{Min} [D_{i,j}^k, D_{i,k+1}^{k+1} + D_{k+1,i}^{k+1}]$

fin para;

fin para;

fin si;

$k \leftarrow k+1;$

fin mientras;

Fin.

El número de operaciones de este algoritmo puede calcularse fácilmente. En cada iteración se hacen:

$2k$ sumas (y comparaciones) para calcular $D_{i,k+1}$ y $D_{k+1,i}$ y $k(k-1)$ operaciones para calcular $D_{i,j}^{k+1}$

1 suma (y comparación) si se trata de detectar los circuitos absorbentes.

total $\sum_{k=1..|V|} k(k-1) + 3k = \sum_{k=1..|V|} k(k+2) = (2n^3 + 3n^2 - 17n + 12)/6$, lo cual resulta en $O(n^3)$.

CAPITULO III

Flujo Máximo

III.1 Condiciones para hallar el Flujo Máximo en una Red.

Prácticamente hablando, un problema de flujo máximo podría consistir en buscar la cantidad máxima de agua que puede fluir, en un determinado momento, desde una estación de bombeo hasta un reservorio, a través de un sistema de acueductos. Podría representar también la cantidad máxima de vehículos por unidad de tiempo que soporta una red vial entre dos ciudades.

Para definir formalmente el problema del Flujo Máximo, veremos primero otras dos definiciones.

En una red $R(V,E,c)$, un NODO FUENTE es un nodo s tal que no existe ningún arco $e \in E$, tal que $T(e) = s$. Se llama NODO DESTINO a un nodo p tal que no existe ningún arco $e \in E$ con $I(e) = p$.

Dada una red $R(V,E,c)$ con un nodo FUENTE s y un nodo DESTINO p , donde la función $c: E \rightarrow \mathbf{R}$ representa una cierta "capacidad" definida sobre los arcos, buscar el "Flujo Máximo" consiste en buscar un vector $f \in \mathbf{R}^m$, $m = |E|$, tal que:

- i) f sea un **flujo** sobre el grafo $G(V, \tilde{E})$, donde $\tilde{E} = E \cup \{(p,s)\} = E \cup \{e_r\}$
- ii) $0 \leq f(e) \leq c(e)$, $\forall e \in E$.
- iii) $f(e_r) = f((p,s))$ sea máximo bajo las condiciones i) e ii).

Definición: El arco $e_r = (p,s)$ se llama "Arco de Retorno". Este arco se agrega a la red para efectos de facilitar la resolución del problema, lo cual veremos a continuación.

Definición: La satisfacción de la condición i), es decir, que f sea un FLUJO, implica que deben cumplirse las leyes de "Conservación de Flujo" sobre todos los nodos, es decir que:

$$\forall x \in V: \sum_{e \in E, T(e)=x} f(e) - \sum_{e \in E, I(e)=x} f(e) = 0$$

"todo lo que entra a un nodo es lo mismo que debe salir".

Entre otras cosas, es por ello que es necesario agregar el arco de retorno, para que la ley de conservación pueda cumplirse incluso para los nodos s y p . Debido a esto mismo, el flujo por el arco de retorno, representa la cantidad total de flujo que "pasa" por la red en cualquier momento y es esta cantidad $f(e_r)$ la que se busca maximizar al resolver el problema.

Un flujo f en $R(V, \tilde{E}, c)$ que satisface las condiciones i) e ii) se llama un "flujo factible".

Otros problemas, tales como la presencia de varios nodos fuente (s_1, \dots, s_k) y/o varios nodos destino (p_1, \dots, p_q), pueden ser resueltos fácilmente llevándolos al caso expuesto anteriormente. Para ello basta con agregar un nodo

fFuente ficticio s_0 , o "super-fuente" y/o un nodo destino ficticio p_0 , o "super-destino", con arcos (s_0, s_i) , $i=1, \dots, k$, (p_j, p_0) , $j=1, \dots, k$ con capacidad máxima, además del arco de retorno (p_0, s_0) .

Igualmente, problemas tales como la presencia de restricciones de capacidad sobre los nodos, lo cual podría representar limitaciones de capacidad de intersecciones, pueden también ser fácilmente resueltos bajo

el esquema anterior, representando a cada nodo x como dos nuevos nodos x' e x'' unidos por un arco (x' , x'') con capacidad igual a la capacidad del nodo x original.

Este problema de Flujo Máximo puede ser formulado como un Problema de Programación Lineal (P.P.L.). Para ello usaremos a continuación lo que llamamos la Matriz de Incidencias de una red o digrafo.

La MATRIZ DE INCIDENCIA es una matriz nodos—arcos, es decir, donde las filas representan los nodos del grafo y las columnas los lados del mismo. En un grafo dirigido o en una red las entradas de la matriz serán 1, -1 y 0; 1 si el nodo de la fila i es el extremo terminal del arco de la columna j , -1 si el nodo de la fila i es inicial del arco de la columna j y 0 si el nodo de la fila i no es incidente al arco de la columna j .

Si denotamos \tilde{E} la Matriz de Incidencias de la red $R(V, \tilde{E}, c)$, el problema del flujo máximo se convierte en el siguiente P.P.L.

$$\begin{aligned}
 &\max \quad f(e_r) \\
 &\text{s.a. } \tilde{E}f = 0 \quad \text{donde: } f \text{ es el vector "flujo"} \\
 &\quad f \leq c \quad \quad \quad c \text{ es el vector "capacidad"} \\
 &\quad f \geq 0
 \end{aligned}$$

III.2 Fundamentos del Algoritmo de Ford & Fulkerson.

Una primera idea para este algoritmo es: dado un flujo f inicial factible, buscar un camino (elemental) C de s a p tal que ningún arco del camino esté saturado, es decir que $\forall e \in C: f(e) < c(e)$. De conseguirse este camino, el flujo puede ser aumentado en cada uno de los arcos en:

$$\varepsilon = \text{Miv}_{e \in X} \{c(e) - f(e) > 0\}$$

Sin embargo este procedimiento podría bloquearse, como se ve en la Figura III.1, sin que se haya encontrado el flujo máximo.

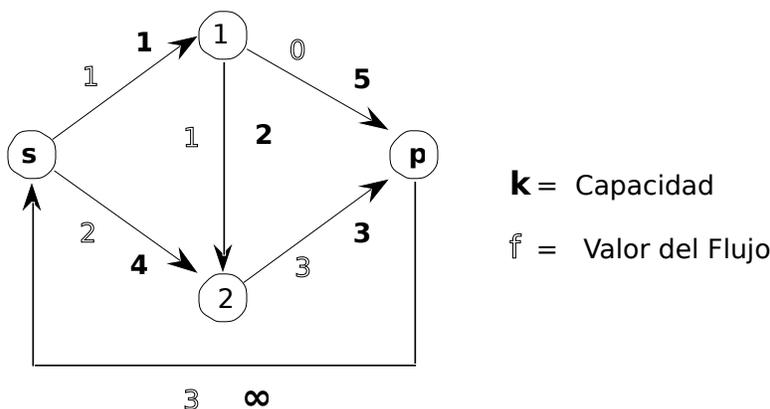


Figura III.1

Una segunda idea consiste en buscar una cadena (elemental) Γ tal que, si agregamos el arco de retorno e_r a esta cadena Γ , podemos dividir los arcos de Γ en:

$$\Gamma^+ = \{e \in \Gamma / e \text{ tiene orientación igual a } e_r \text{ en } \Gamma\}$$

$$\Gamma^- = \{e \in \Gamma / e \text{ tiene orientación contraria a } e_r \text{ en } \Gamma\}$$

Cada uno de los arcos en Γ debe ser tal que:

$$\text{Si } e \in \Gamma^+ : f(e) < c(e)$$

$$\text{Si } e \in \Gamma^- : f(e) > 0.$$

Una cadena como la descrita anteriormente se llama una CADENA AUMENTANTE. Una vez hallada esta cadena, el flujo a travez de la red puede aumentarse en:

$$e = \text{Min} \{ e_1, e_2 \} = \text{Min} \left\{ \underset{e \in \Gamma^+}{?} \text{Min} \{ \chi(e) - \phi(e) \}, \underset{e \in \Gamma^-}{?} \text{Min} \{ \phi(e) \} \right\} > 0$$

haciendo:

$$f(e) = f(e) + e \quad \text{si } e \in \Gamma^+ \cup \{e_r\}$$

$$f(e) = f(e) - e \quad \text{si } e \in \Gamma^-$$

Este será un nuevo flujo factible ya que:

- i) Las restricciones de conservación de flujo se mantiene sobre todos los nodos:
 Si Γ no pasa por $x \in V$, es evidente que el flujo a través de x se mantienen constante.
 Si x es un nodo tal que Γ pasa por él, habrá exactamente dos arcos de Γ incidentes a x . En esta situación se pueden presentar cuatro casos. Llamemos e_1 y e_2 los dos arcos incidentes a x , entonces:

$$\begin{aligned} T(e_1) = T(e_2) = x &\text{ ó} \\ I(e_1) = I(e_2) = x &\text{ ó} \\ T(e_1) = I(e_2) = x &\quad \text{ó} \\ I(e_2) = T(e_1) = x. & \end{aligned}$$

En todos estos casos las modificaciones sobre el arco e_1 compensan las modificaciones en el arco e_2 de forma que las leyes de conservación de flujo se mantienen.

- ii) El nuevo valor del flujo f verifica de nuevo: $0 \leq f(e) \leq c(e), \forall e \in E$, por la forma como se escoge e .

III.3 Teorema del Corte Mínimo.

Definición: Sea Y un subconjunto de nodos de $R(V,E,c)$. El conjunto de arcos:

$$G(Y) = \{e \in E / I(e) \in Y, T(e) \notin Y \cup \{ T(e) \in Y, I(e) \notin Y \}$$

se llama un **COCICLO INDUCIDO** por el conjunto Y . Los arcos de $\Gamma(Y)$ los dividimos en dos subconjuntos Γ^+, Γ^- , donde

$$\Gamma^+(Y) = \{ e \in E / I(e) \in Y, T(e) \notin Y \}$$

$$\Gamma^-(Y) = \{ e \in E / T(e) \in Y, I(e) \notin Y \}$$

Definición: Se llama un "Corte que separa s de p " a un subconjunto K de arcos de $R(V,E,c)$ tal que, existe un subconjunto de nodos Y con $s \in Y, p \notin Y$ con:

$$K = \Gamma^+(Y).$$

Es interesante notar que si K es un corte que separa s de p , entonces todo camino de s a p en $R(V,E,c)$ tiene un arco en K .

Definición: Se define la "Capacidad de un Corte K " en $R(V,E,c)$ como la suma de las capacidades de los arcos en K , es decir:

$$c(K) = \sum_{\epsilon \in K} \chi(\epsilon)$$

Teorema III.1:

Para todo flujo factible f en $R(V, \tilde{E}, c)$ y para todo corte K que separa s de p :

$$f(e_r) \leq c(K).$$

Demostración:

Sea Y un subconjunto de nodos tal que $K = \Gamma^+(Y)$. Si sumamos las restricciones de conservación de flujo sobre todos los nodos en Y obtenemos:

$$\sum_{e \in \Gamma^+(\Psi)} f(e) - \sum_{\epsilon \in \Gamma^-(\Psi)} \phi(\epsilon) = 0$$

Puesto que $s \in Y, p \notin Y$, entonces $e_r = (p,s) \in \Gamma^-(Y)$, de donde despejando se obtiene

$$f(e_r) = \sum_{e \in \Gamma^+(\Psi)} f(e) - \sum_{\epsilon \in \Gamma^-(\Psi) - \{e_r\}} \phi(\epsilon)$$

pero $0 \leq f(e) \leq c(e), \forall e \in E \Rightarrow \sum_{e \in \Gamma^+(\Psi) - \{e_r\}} f(e) \geq 0$, y

$$\sum_{\epsilon \in \Gamma^-(Y)} [f(e)] = \sum_{\epsilon \in K} [f(e)] \leq \sum_{\epsilon \in K} [c(e)]$$

luego $f(e_r) \leq c(K)$. ‡

III.4 Implementación del Algoritmo F & F para calcular Flujo Máximo.

Este algoritmo requiere que el flujo f inicial sea un flujo factible en $R(V,E,b,c)$. Normalmente se toma $f(e)=0, \forall e \in E$. MODIF es una variable booleana que vale **verdadero** si el flujo f es modificado por el algoritmo. ExisteFM, también booleana, indica si existe un Corte de capacidad finita, en cuyo caso ExisteFM = **verdadero**.

Algoritmo FLUJOMAX: (* Ford & Fulkerson *)

Entradas: $V, E, I, T, e_r, s, p, b, c, f;$

Salidas: $f, Y, \text{MODIF}, \text{ExisteFM};$

procedimiento BUSCARCADENAAUMENTANTE ($V, E, I, T, e_r, s, p, b, c, f; Y, A, \text{epsilon};$)

(* $\partial(s)$ es la cantidad máxima en que se desea incrementar el flujo del arco e_r *)

comienzo procedimiento (* BUSCARCADENAAUMENTANTE *)

para todo $\{x \in V / x \neq s\}$ **hacer** $\partial(x) \leftarrow 0;$ **fin para;**

$Y \leftarrow \{s\}; \text{ExisteC} \leftarrow \text{verdadero}; \partial(s) \leftarrow c(e_r) - f(e_r); \text{epsilon} := 0;$

si $(\partial(s) > 0)$ **entonces**

mientras $(p \notin Y \text{ y } \text{ExisteC})$ **hacer**

si $(\exists e=(x,y) / x \in Y, y \notin Y, f(e) < c(e))$

entonces

$Y \leftarrow Y \cup \{y\}; A(y) \leftarrow e;$

$\partial(y) \leftarrow \text{Min} [\partial(x), c(e) - f(e)]$

sino

si $(\exists e=(x,y) / y \in Y, x \notin Y, f(e) > b(e))$ (* $b(e) = 0$ en $R(V,E,c)$ *)

entonces

$Y \leftarrow Y \cup \{x\}; A(x) \leftarrow e;$

$\partial(x) \leftarrow \text{Min} [\partial(y), f(e) - b(e)]$

sino $\text{ExisteC} \leftarrow \text{falso};$

fin si;

fin si;

fin mientras;

fin si;

si (ExisteC) **entonces** $\text{epsilon} \leftarrow \partial(p)$ **fin si;**

fin procedimiento; (* BUSCARCADENAAUMENTANTE *)

Comienzo (* Flujo Max *)

$\text{MODIF} \leftarrow \text{falso}; \text{ExisteFM} \leftarrow \text{verdadero};$

repetir

BUSCARCADENAAUMENTANTE ($V, E, I, T, e_r, s, p, b, c, f; Y, A, \text{epsilon};$)

si $(p \in Y)$ y $(\text{epsilon} \neq \text{infinito})$ **entonces**

$x \leftarrow p; \Gamma^+ \leftarrow \{e_r\}; \Gamma^- \leftarrow \{\}; \text{MODIF} := \text{verdadero};$

mientras $(x \neq s)$ **hacer**

$e \leftarrow A(x);$

```
si  $(x = T(u))$  entonces  $[\Gamma^+ \leftarrow \Gamma^+ \cup \{e\}; x \leftarrow I(e)]$   
    sino  $[\Gamma^- \leftarrow \Gamma^- \cup \{e\}; x \leftarrow T(e)]$ ;  
fin si;  
fin mientras;  
para todo  $e \in G^+$  hacer  $f(e) \leftarrow f(e) + \text{epsilon}$  fin para;  
para todo  $e \in G^-$  hacer  $f(e) \leftarrow f(e) - \text{epsilon}$  fin para;
```

```
    fin si
    hasta que (p≠Y ó epsilon = infinito);
si (epsilon = infinito) entonces ExisteFM:= falso fin si;
Fin. (* Flujo Max *)
```

III.5 Justificación:

- i) Sea $E = \{ e \in E \mid \exists y \in Y \text{ con } A[y] = e \}$ (A e Y son salidas de BuscarCadenaAumentante). El grafo $G(V, E)$ es claramente un árbol. La cadena que une s a p en $G(V, E)$ unido con e_r es un ciclo G donde G^+ son los arcos orientados como e_r y G^- son los arcos orientados en sentido contrario a e_r .
 - ii) Por la forma como se calcula e , al modificar los valores de $f(e)$ se mantiene la factibilidad.
- i) + ii) => el flujo que se obtiene al final de cada iteración es siempre factible y estrictamente mejor que el anterior, pues $f(e_r) := f(e_r) + e$, ($e > 0$).

Una vez que termina el algoritmo con $p \in Y$, se puede ver que hemos encontrado un corte K de s a p con $c(K) = f(e_r)$, lo cual según el Teorema III.1 implica que f no puede aumentar más (y además no podremos conseguir otro corte de capacidad menor).

Veamos a continuación que efectivamente se ha detectado tal corte:

Sea Y el último conjunto de nodos marcados. Por construcción $p \in Y$.

Sea $e = (x, y) \in G^+(Y) \Rightarrow f(e) = c(e)$ (sino $y = T(e)$ sería marcado en la marcación directa).

Sea $e = (y, x) \in G^-(Y) \Rightarrow f(e) = 0$ (sino $y = I(e)$ sería marcado en la marcación inversa).

Tomando $K = G^+(Y)$ tenemos $f(e_r) = \sum_{e \in K} [c(e)] = c(K)$. \ddagger

Teorema III.2. (Flujo Max - Corte Min)

El valor máximo de $f(e_r)$ para un flujo factible f en $R(V, E, c)$ es igual a la capacidad de un corte de s a p de capacidad mínima. En particular $f(e_r)$ será no acotado si y sólo si no existe en $R(V, E, c)$ un corte de capacidad finita que separe s de p .

Demostración: Podemos remitirnos al hecho de que el problema de flujo máximo es un P.P.L. que tiene al menos una solución: $f(e) = 0, \forall e \in E$. Además si existe algún corte de capacidad finita, el problema es acotado y por lo tanto tiene solución óptima finita.

En cuanto al algoritmo de Ford & Fulkerson (F&F), siempre que las capacidades de los arcos sean números racionales, podemos considerar que todas son múltiplos enteros de algún número δ en \mathbf{R} . Así, si existe corte finito, al comenzar con $f(e)=0, \forall e \in E$, e será siempre múltiplo de δ , luego los valores del vector f para cada solución serán también múltiplos de δ , y $f(e_r)$ aumenta en cada iteración en un múltiplo de δ estrictamente mayor que cero. Este crecimiento al ser el problema acotado se detiene en un número finito de pasos.

Si las capacidades de los arcos son números irracionales, es posible que el algoritmo se torne infinito, o peor aún, que converja a soluciones lejanas al óptimo. Esto sin embargo, en los casos reales no ocurre puesto que siempre se pueden tomar aproximaciones racionales lo suficientemente buenas ya que los irracionales no tienen

una representación exacta en la computadora.

En el caso que no exista corte finito, esto implica que existe al menos un camino C_s^p de s a p en el cual todos los arcos tienen capacidad infinita. Siendo finito el número de caminos elementales y estando siempre

este camino C_s^P no saturado, es seguro que en la primera etapa de la marcación (marcación directa) este camino será encontrado y se detectará al ser $e = \mathbf{infinito}$ para tal camino.

III.6 Extensión a Redes Canalizadas.

Llamamos "Red Canalizada" a una red $R(V,E,b,c)$ en la cual además de la función "capacidad", que acota superiormente el flujo, tenemos otra función, $b: E \rightarrow \mathbf{R}$ que corresponde a una cota inferior para el flujo por los arcos; es por esto que decimos que el flujo esta canalizado:

$$b(e) \leq f(e) \leq c(e), \forall e \in E.$$

En este caso la solución inicial trivial $f(e) = 0, \forall e \in E$, puede no ser factible por lo que habrá que disponer de un mecanismo para encontrar al menos una solución inicial factible al problema. Esto se verá en el próximo capítulo.

Una vez que se tenga una solución inicial factible, la única modificación necesaria al algoritmo de F&F para redes canalizadas son las siguientes:

- i) Para la marcación inversa se tomarán aquellos arcos con $f(e) > b(e)$ en lugar de tomar los arcos con $f(e) > 0$.

CAPITULO IV

Flujos Factibles

IV.1 Condiciones de existencia de Flujos Factibles.

Consideremos una red $R(V,E,b,c)$ con $b: E \rightarrow \mathbf{R}$, $c: E \rightarrow \mathbf{R}$ y $b(e) \leq c(e)$, $e \in E$.

Se desea encontrar un vector f en \mathbf{R}^m ($m=|E|$), que llamaremos un "flujo factible" en $R(V,E,b,c)$, el cual debe satisfacer las siguientes condiciones:

- i) f es un flujo en $G(V,E)$
- ii) $b(e) \leq f(e) \leq c(e)$, $e \in E$.

Veremos en lo que sigue un método que nos permitirá encontrar un flujo factible, siempre que este exista, y una condición necesaria y suficiente para su existencia.

IV.2 Teorema de Hoffman.

Una condición necesaria y suficiente para que exista un flujo factible en $R(V,E,b,c)$ es que:

$$\text{" cociclo } G(Y) \in G(V,E): \sum_{e \in G^-(Y)} [b(e)] \leq \sum_{e \in G^+(Y)} [c(e)] \quad (*)$$

Demostración.

Sea f un flujo factible en $R(V,E,b,c)$ y $G(Y)$ un cociclo, entonces, sumando todas las ecuaciones de conservación de flujo sobre los nodos en Y obtenemos:

$$\sum_{e \in G^-(Y)} [f(e)] = \sum_{e \in G^+(Y)} [f(e)], \text{ pero}$$

$$\sum_{e \in G^-(Y)} [b(e)] \leq \sum_{e \in G^-(Y)} [f(e)] \quad \text{y} \quad \sum_{e \in G^+(Y)} [f(e)] \leq \sum_{e \in G^+(Y)} [c(e)]$$

de donde se obtiene la condición (*) buscada.

El siguiente es un algoritmo constructivo, el cual cuando no puede encontrar un flujo factible pone en evidencia la existencia de un cociclo que no cumple la condición de Hoffman, lo cual demuestra la suficiencia de

la misma.

IV.3 Algoritmo para encontrar un Flujo Factible.

Algoritmo FLUJOFACT:

Entradas: V, E, I, T, b, c, fi;

Salidas: f, Y, ExistFF;

Comienzo

ExistFF:= verdadero; f := fi (* Flujo Inicial, generalmente = 0 "uCEE *)

repetir

INDIC := $\sum_{u/f(u)>c(u)} [f(u)-c(u)] + \sum_{u/f(u)<b(u)} [b(u)-f(u)];$

si (INDIC \neq 0 y ExistFF)

entonces

si ($\hat{e} = (x,y) \in E / f(\hat{e}) > c(\hat{e})$)

entonces

$I(u_p) := p := y;$

$T(u_p) := s := x;$

$c\sim(u_p) := f(\hat{e}) - c(\hat{e})$

a := 1;

sino

si ($\hat{e} = (x,y) \in E / f(\hat{e}) < b(\hat{e})$)

entonces

$I(u_p) := p := x;$

$T(u_p) := s := y;$

$c\sim(u_p) := b(\hat{e}) - f(\hat{e});$

a := -1;

fin si;

fin si;

para todo uCEE **hacer**

$b\sim(u) := \mathbf{Min} [b(u), f(u)]$

$c\sim(u) := \mathbf{Max} [f(u), c(u)]$

$f\sim(u) := f(u);$

fin para;

$f\sim(u_p) := 0; b\sim(u_p) := 0;$

FLUJOMAX (V,E,I,T,u_p,s,p,b \sim ,c \sim ,f \sim ; f \sim , Y, MODIF, ExistFM);

si ($f\sim(u_p) = c\sim(u_p)$) (* El flujo se pudo cambiar *)

entonces

para todo (u $\in E / u \neq \hat{e}$) **hacer**

$f(u) := f\sim(u);$

fin para;

$f(\hat{e}) := f\sim(\hat{e}) - a * f\sim(u_p);$

```
        sino
            ExistFF:= Falso;    (* El flujo no cumple condición de Hoffman *)
        fin si;
    fin si;
    hasta que (INDIC = 0 ó no ExistFF).
Fin.
```

Observación: $INDIC = 0 \Rightarrow f$ es un flujo factible. Las nuevas cotas \tilde{b}, \tilde{c} se definen de forma tal que $f = \tilde{f}$ sea factible en $R(V,E \setminus \{u_r\}, \tilde{b}, \tilde{c})$ para poder llamar a FLUJOMAX que requiere un flujo inicial factible.

El algoritmo anterior se detiene cuando $INDIC = 0$, es decir cuando se tiene un flujo f factible, o cuando $ExistFF$ es falso.

En este segundo caso, sea Y el último conjunto de nodos "marcados" que devuelve FLUJOMAX :

a) Si $a = 1$ entonces:

$$\begin{aligned} " u \in G^+(Y) - \{\hat{e}\}: & \quad f(u) = \tilde{f}(u) = \tilde{c}(u) \geq c(u) \\ " u \in G^-(Y): & \quad f(u) = \tilde{f}(u) = \tilde{b}(u) \leq b(u) \\ f(\hat{e}) = \tilde{f}(\hat{e}) - a * \tilde{f}(u_r) & < c(\hat{e}) \end{aligned}$$

b) Si $a = -1$ entonces:

$$\begin{aligned} " u \in G^+(Y): & \quad f(u) = \tilde{f}(u) = \tilde{c}(u) \geq c(u) \\ " u \in G^-(Y) - \{\hat{e}\}: & f(u) = \tilde{f}(u) = \tilde{b}(u) \leq b(u) \\ f(\hat{e}) = \tilde{f}(\hat{e}) - a * \tilde{f}(u_r) & > b(\hat{e}). \end{aligned}$$

En ambos casos:

$$0 = \sum_{u \in G^+(Y)} [f(u)] - \sum_{u \in G^-(Y)} [f(u)] > \sum_{u \in G^+(Y)} [c(u)] - \sum_{u \in G^-(Y)} [b(u)]$$

de donde podemos concluir que f no factible $\Rightarrow \exists G(Y)$ que no cumple la condición de Hoffman.

CAPITULO V

Flujo de Costo Mínimo

V.1 Problemas de Flujo de Costo Mínimo.

Planteamiento.

Sea $R(V,E,a,b,c)$ con $a: E \rightarrow \mathbf{R}$, $b: E \rightarrow \mathbf{R}$, $c: E \rightarrow \mathbf{R}$, $b(e) \leq c(e)$, " $e \in E$.".

Se desea encontrar un vector f tal que:

- i) f sea un flujo en $G(V,E)$
- ii) $b(e) \leq f(e) \leq c(e)$, " $e \in E$."
- iii) $\sum_{e \in E} [a(e) * f(e)]$ sea mínimo.

Así como el problema de Flujo Máximo, el problema de Flujo de Costo Mínimo, también puede formularse como un P.P.L.:

$$\begin{aligned} \min \quad & \mathbf{a}^t * \mathbf{f} \\ \text{s.a.} \quad & \mathbf{E} * \mathbf{f} = \mathbf{0} \\ & \mathbf{f} \geq \mathbf{b} \\ & -\mathbf{f} \geq -\mathbf{c} \end{aligned}$$

Antes de ver como resolver este problema, veremos como otros problemas conocidos, pueden plantearse como problemas de flujo de costo mínimo.

a) **Ruta Mínima de s a p en $R(V,E,d)$.**

En este caso el planteamiento lo hacemos construyendo una nueva red $R(V, E \gg \{u_r\}, a,b,c)$, donde

$$\begin{aligned} u_r = (p,s), \quad & a(u_r) = 0, \quad b(u_r) = 1, \quad c(u_r) = \infty \\ " e \in E : & a(e) = d(e), \quad b(e) = 0, c(e) = \infty. \end{aligned}$$

b) **Flujo Máximo de s a p en $R(V, E \gg \{u_r\}, b, c)$.**

Construiremos la nueva red $R'(V, E \gg \{u_r\}, a,b,c)$, donde

$$\begin{aligned} u_r = (p,s) \quad & a(u_r) = -1, \quad b(u_r) = 0, \quad c(u_r) = \infty \\ " e \in E : & a(e) = 0, b(e) = b(e), \quad c(e) = c(e). \end{aligned}$$

c) **Problema de Transporte.**

Supongamos N fuentes con capacidades a_i , M destinos con demandas b_j , y costos unitarios de transporte c_{ij} de la fuente i al destino j .

La formulación como P.P.L. de este problema es:

$$\begin{aligned} \text{Min} \quad & \sum_i \sum_j c_{ij} X_{ij} \\ \text{s.a.} \quad & \sum_j x_{ij} \leq a_i, \quad "i \end{aligned}$$

$$\begin{aligned}\sum_j x_{ij} &\geq b_j, \quad "j \\ x_{ij} &\geq 0, \quad "i,j.\end{aligned}$$

Construiremos entonces la siguiente red $R(F \gg D \gg \{(p,s)\}, E1 \gg E2 \gg E3, a,b,c)$ donde

$F = \{ \text{cjto. de nodos correspondientes a cada fuente} \}$

$D = \{ \text{cjto. de nodos correspondientes a cada destino} \}$

$s = \text{super-fuente}, \quad p = \text{super-destino}.$

$E1 = \{(s, F_i) / F_i \text{ es una fuente}\}$

$E2 = \{(D_j, p) / D_j \text{ es un destino}\}$

$E3 = \{(F_i, D_j) / F_i \text{ es una fuente, } D_j \text{ es un destino}\}$

$$\begin{aligned}
 & \text{" } e \in E1: a(e) = 0, b(e) = 0, c(e) = a_i \\
 & \text{" } e \in E2: a(e) = 0, b(e) = b_j, \quad c(e) = \infty \\
 & \text{" } e \in E3: a(e) = c_{ij}, \quad b(e) = 0, c(e) = \infty, \quad \text{para } e = (F_i, D_j) \\
 & u_r = (p,s): \quad a(u_r) = 0, \quad b(u_r) = \sum b_j, \quad c(u_r) = \sum a_i
 \end{aligned}$$

Definición. Dado un flujo f factible en una red $R(V,E,a,b,c)$, podemos construir una red:

$$R^{\wedge}(V, E^{\wedge}, d) \text{ donde } E^{\wedge} = E^{\wedge+} \cup E^{\wedge-}$$

construidos como sigue:

$$\begin{aligned}
 \text{Si } f(e) < c(e) \text{ entonces } & e' \in E^{\wedge+} & \text{con } I(e') = I(e), T(e') = T(e), & d(e') = a(e). \\
 \text{Si } f(e) > b(e) \text{ entonces } & e'' \in E^{\wedge-} & \text{con } I(e'') = T(e), T(e'') = I(e), & d(e'') = -a(e).
 \end{aligned}$$

Teorema V.1.

Un flujo factible f en $R(V,E,a,b,c)$ es de costo mínimo si y sólo si la red $R^{\wedge}(f)$ no tiene circuitos absorbentes.

Demostración.

==>) Supongamos que $R^{\wedge}(f)$ si tiene un circuito absorbente C , llamemos:

$$C^+ = C \ll E^{\wedge+}, \quad C^- = C \ll E^{\wedge-}$$

Llamemos $G^+ =$ arcos en E asociados a arcos en C^+ , $G^- =$ arcos en E asociados a arcos en C^- .

Por ser C un circuito absorbente: $\sum_{e \in C} d(e) < 0$.

$$\text{Pero } \sum_{e \in C} d(e) = \sum_{e' \in C^+} d(e') + \sum_{e'' \in C^-} d(e'') = \sum_{e \in G^+} a(e) - \sum_{e \in G^-} a(e).$$

$$\text{Sea } e = \text{Min} \{ \text{Min}_{e \in G^+} [c(e) - f(e)], \text{Min}_{e \in G^-} [f(e) - b(e)] \} > 0.$$

Podemos entonces construir un nuevo flujo g como:

$$\begin{aligned}
 g(e) &= f(e) + e, \quad \text{si } e \in G^+ \\
 g(e) &= f(e) - e, \quad \text{si } e \in G^- \\
 g(e) &= f(e), \quad \text{si } e \in (G^+ \cup G^-)^c = G.
 \end{aligned}$$

El costo de este flujo es:

$$\begin{aligned}
 a * g &= \sum_{e \in E} a(e) * g(e) = \sum_{e \in G^+} a(e) * [f(e) + e] + \sum_{e \in G^-} a(e) * [f(e) - e] \\
 &+ \sum_{e \in G} a(e) * f(e) =
 \end{aligned}$$

$$= \sum_{e \in E} a(e) \cdot f(e) + e * \left[\sum_{e \in E^+} a(e) - \sum_{e \in E^-} a(e) \right] < a * f$$

de donde se observa que **g** tiene menor costo que **f**.

\Leftarrow Si $R^{\wedge}(f)$ no tiene circuitos absorbentes, entonces, agregando los arcos necesarios para que un cierto nodo **s**, escogido arbitrariamente, sea una raíz de $R^{\wedge}(f)$ y dando a estos arcos (E^{\wedge}) una distancia $M > 0$, podemos asegurar que existen rutas mínimas desde **s** hasta cada nodo **x** de $R^{\wedge}(f)$.

De esta manera, podemos asociar los valores

$\pi(x)$ = distancia mínima de s a x , $x \in V$,
 tales que:
 $\forall e \in E: \pi(T(e)) - \pi(I(e)) \leq d(e)$.

Plantearemos a continuación el dual del problema de flujo de costo mínimo:

$$\begin{aligned} \max \quad & 0x + bY - cZ \\ \text{s.a.} \quad & xE + Y - Z = a \\ & Y, Z \geq 0 \end{aligned}$$

Definamos:

$$\begin{aligned} t(e) &= \pi(T(e)) - \pi(I(e)) \\ a(e) &= \text{Max} [0, a(e) - t(e)] \\ b(e) &= \text{Max} [0, t(e) - a(e)] \end{aligned}$$

Así,

$$\begin{aligned} \text{si } e \in E: \quad & \pi(T(e)) - \pi(I(e)) = \pi(T(e)) - \pi(I(e)) \leq d(e) = a(e) \implies t(e) \leq a(e) \\ \text{si } e \in E: \quad & \pi(T(e)) - \pi(I(e)) = \pi(I(e)) - \pi(T(e)) \leq d(e) = -a(e) \implies t(e) \geq a(e) \end{aligned}$$

Veamos que si las variables duales (X, Y, Z) toman los valores (π, a, b) , esto es una solución factible.

$$\begin{aligned} a(e) \geq 0, \forall e \in E, \text{ pues si } a(e) - t(e) < 0 \text{ entonces } a(e) = 0. \\ b(e) \geq 0, \forall e \in E, \text{ pues si } t(e) - a(e) < 0 \text{ entonces } b(e) = 0. \end{aligned}$$

La restricción dual se transforma en:

$$\pi(T(e)) - \pi(I(e)) + a(e) - b(e) = a(e)$$

y es fácil verificar que esta igualdad siempre se cumple.

Ahora sólo falta ver que las soluciones (π, a, b) , para el dual, y f para el primal, cumplen las condiciones de holgura complementaria y que por lo tanto ambas son óptimas a sus respectivos problemas.

$$\begin{aligned} f(e) > b(e) \implies t(e) \geq a(e) \implies a(e) = 0 \\ f(e) < c(e) \implies t(e) \leq a(e) \implies b(e) = 0 \end{aligned}$$

$$\begin{aligned} a(e) > 0 \implies a(e) > t(e) \implies f(e) = b(e) \\ b(e) > 0 \implies t(e) > a(e) \implies f(e) = c(e). \end{aligned}$$

l.q.q.d.

V.2 Algoritmo para obtener el Flujo de Costo Mínimo.

Algoritmo FLUJOCOSTOMIN:

Entradas: V, E, I, T, a, b, c, fi;

Salidas: f, Y, ExisteFCM;

Comienzo

FLUJOFACT (V,E,I,T,b,c,fi; f,Y,ExisteFCM);

si (ExisteFCM)

entonces

repetir

$E^{\wedge} := E^{\wedge'} \gg E^{\wedge''} \gg E^{\wedge'''};$

$R^{\wedge}(f) := R^{\wedge}(V, E^{\wedge}, d);$ (* Construir $R^{\wedge}(f)$ según lo especificado *)

RUTAMINGENERAL (V,E[^], I[^], T[^], d, s; π , A, EsRaiz, CircuitAbs, C);

si (CircuitAbs)

entonces

$C^+ := C \ll E^{\wedge'};$

$C^- := C \ll E^{\wedge''};$

$G^+ :=$ arcos de E asociados a $C^+;$

$G^- :=$ arcos de E asociados a $C^-;$

$e := \text{Min} \{ \text{Min}_{e \in G^+} [c(e)-f(e)], \text{Min}_{e \in G^-} [f(e)-b(e)] \};$ (* $e > 0$ *)

si ($e \neq \infty$)

entonces

caso "e \in E de

$e \in G^+ : f(e) := f(e) + e;$

$e \in G^- : f(e) := f(e) - e;$

$e \notin G : f(e) := f(e);$

fin caso;

sino

ExisteFCM:= falso;

fin si;

fin si;

hasta que (no CircuitAbs) ó (no ExisteFCM);

fin si;

Fin.

CAPITULO VI

Problemas de Flujo Maximo de Costo Mınimo.

VI.1 Planteamiento.

Sea $R(V, E \gg \{u_r\}, a, c)$ con $u_r = (p, s)$ y

$$a: E \rightarrow \mathbf{R}^+, \quad c: E \rightarrow \mathbf{R}^+.$$

El problema de hallar el "flujo mıximo de costo mınimo" en $R(V, E, a, c)$ consiste en buscar un vector f tal que:

- i) f sea un flujo en $G(V, E \gg \{u_r\})$
- ii) $0 \leq f(e) \leq c(e), \quad \forall e \in E$.
- iii) $f(u_r)$ sea mıximo.
- iv) $\sum_{e \in E} a(e) \cdot f(e)$ sea mınimo bajo i), ii) e iii).

Es decir, se busca entre todos los flujos factibles en $R(V, E \gg \{u_r\}, a, c)$ aquellos con $f(u_r)$ mıximo, y entre ellos aquel que tenga el costo mınimo. Tenemos entonces una mezcla de dos problemas ya vistos: Flujo Mıximo y Flujo de Costo Mınimo.

Sean $F = \max f(u_r)$ sobre $R(V, E \gg \{u_r\}, a, c)$ y $A = \sum_{e \in E} a(e) + 1$.

Existen dos formas de llevar el problema de Flujo Mıximo de Costo Mınimo a un problema de Flujo de Costo Mınimo :

- a) Podemos considerar $R'(V, E', a', b', c')$ donde:

$$E' = E \gg \{u_r\}; \quad a'(u_r) = -A; \quad c'(u_r) = \infty;$$

$$a'(e) = a(e) \quad \forall e \in E; \quad b'(e) = 0 \quad \forall e \in E; \quad c'(e) = c(e) \quad \forall e \in E$$

Ası, dando un valor negativo $-A \ll 0$ al arco u_r , al minimizar $\sum_{e \in E'} a'(e) \cdot f(e)$ se tratarı de dar el valor mıs grande a $f(u_r)$.

- b) Otra forma es con $R''(V, E'', a'', b'', c'')$ con $E'' = E \gg \{u_r\}$, donde:

$$a''(u_r) = 0, a''(e) = a(e), \quad \forall e \in E$$

$$b''(u_r) = F, \quad b''(e) = 0, \quad \forall e \in E$$

$$c''(u_r) = \infty, \quad c''(e) = c(e), \quad \forall e \in E.$$

Definición: Dado un flujo factible f en $R(V, E \gg \{u_r\}, a, c)$, llamaremos $R^\wedge(f)$ la red $R^\wedge(V, E^\wedge, d^\wedge)$ donde: $E^\wedge = E^\wedge' \cup E^\wedge''$ construyendo E^\wedge' y E^\wedge'' según

si $u \in E$ con $f(e) < c(e)$, entonces $e' \in E^\wedge'$: $I(e') = I(e)$, $T(e') = T(e)$, $d(e') = a(e)$.
 si $u \in E$ con $f(e) > 0$, entonces $e'' \in E^\wedge''$: $I(e'') = T(e)$, $T(e'') = I(e)$, $d(e'') = -a(e)$.

Nota: obsérvese que en este caso $R^\wedge(f)$ no tendrá arcos asociados a u_r .

Definición: Llamamos una "Solución Parcial" al problema de Flujo Máximo de Costo Mínimo, a una solución óptima al problema $P(v)$, $v \in \mathbb{R}^+$ definido como:

$$\begin{aligned} \min \quad & a * f \\ \text{s.a.} \quad & E * f = 0 \\ & f(e) \leq c(e), \quad "e \in E \\ & f(u_p) \geq v \\ & f(e) \geq 0, \quad "e \in E. \end{aligned}$$

El dual $D(v)$ de este problema sería entonces:

$$\begin{aligned} \max \quad & w = v.g(u_p) - c.g(e) \\ \text{s.a.} \quad & \pi(T(e)) - \pi(I(e)) - g(e) \leq a(e), \quad "e \in E \\ & \pi(s) - \pi(p) + g(u_p) = 0 \\ & g(e) \geq 0, \quad "e \in E \end{aligned}$$

Teorema VI.1.

Si (π, g) es solución óptima de $D(v)$ entonces:

$$g(e) = \text{Max}_{"e \in E} [0; \pi(T(e)) - \pi(I(e)) - a(e)]$$

Demostración.

$g(e)$ debe ser mayor o igual a 0, $"e \in E$, además, en $D(v)$ cada $g(e)$ aparece en una única restricción: la restricción asociada al arco e . Luego se puede despejar:

$$g(e) \geq \pi(T(e)) - \pi(I(e)) - a(e)$$

de donde concluimos que el mínimo valor que puede tomar $g(e)$ es

$$\text{Max}_{"e \in E} [0; \pi(T(e)) - \pi(I(e)) - a(e)]$$

que corresponde a su cota inferior.

Teorema VI.2.

Sea f una solución factible a $P(v)$, entonces:

f es solución óptima a $P(v)$ **si y sólo si** $R^{\wedge}(f)$ no tiene circuitos absorbentes **si y sólo si** $"x \in V$, puede hallarse valores $\pi(x)$ t.q.:

- (1) $0 < f(e) < c(e) \implies \pi(T(e)) - \pi(I(e)) = a(e)$
- (2) $\pi(T(e)) - \pi(I(e)) > a(e) \implies f(e) = c(e)$
- (3) $\pi(T(e)) - \pi(I(e)) < a(e) \implies f(e) = 0.$

Demostración.

Supongamos que $R^*(f)$ si tiene un circuito absorbente C , veremos que en ese caso f no puede ser solución óptima a $P(v)$.

Por ser C un circuito absorbente: $\sum_{e \in C} [d(e)] < 0$.

Sean

$$C^+ = C \ll E^{\wedge '}, \quad C^- = C \ll E^{\wedge ''}$$

$$G^+ = \{e \in E / e \in C^+\}; \quad G^- = \{e \in E / e \in C^-\}$$

$$\sum_{e \in G^+} a(e) - \sum_{e \in G^-} a(e) < 0, \text{ además } e \in G^- \implies f(e) > 0 [b(e)], \text{ y } e \in G^+ \implies f(e) < c(e)$$

luego $e = \text{Min} \{ \text{Min}_{e \in G^+} [c(e) - f(e)], \text{Min}_{e \in G^-} [f(e)] \} > 0$, y entonces el nuevo flujo g :

$$g(e) = f(e) + e, \text{ si } e \in G^+$$

$$g(e) = f(e) - e, \text{ si } e \in G^-$$

$$g(e) = f(e), \text{ si } e \in G$$

es factible a $P(v)$ y su costo :

$$\begin{aligned} \mathbf{a} * \mathbf{g} &= \sum_{e \in E} a(e) * g(e) = \sum_{e \in G^+} a(e) * [f(e) + e] + \sum_{e \in G^-} a(e) * [f(e) - e] + \\ &\quad \sum_{e \in G} a(e) * f(e) = \\ &= \sum_{e \in E} a(e) * f(e) + e * \left[\sum_{e \in G^+} a(e) - \sum_{e \in G^-} a(e) \right] < \mathbf{a} * \mathbf{f} \end{aligned}$$

l.q.q.d.

Por otra parte si $R^{\wedge}(f)$ no tiene circuitos absorbentes, entonces podemos agregar a $R^{\wedge}(f)$ los arcos necesarios, con distancia $M \gg 0$, de forma que el nodo s sea una raíz y entonces podemos asegurar que existe distancia mínima de s a x , $\forall x \in V$.

En otras palabras, podemos encontrar valores $\pi(x)$, $\forall x \in V$ t.q.:

$$\pi(T(e)) - \pi(I(e)) \leq d(e), \forall e \in E.$$

Así, si $e \in E^{\wedge '}$ $\implies \pi(T(e')) - \pi(I(e')) \leq d(e') = a(e)$ (*)

si $e \in E^{\wedge ''}$ $\implies \pi(T(e'')) - \pi(I(e'')) \leq d(e'') = -a(e)$

$$\implies \pi(I(e)) - \pi(T(e)) \leq -a(e)$$

$$\implies \pi(T(e)) - \pi(I(e)) \geq a(e) \quad (**).$$

Luego,

si $0 < f(e) < c(e)$, por (*) y (**): $\pi(T(e)) - \pi(I(e)) = a(e)$ q.e.d. (1)

si $\pi(T(e)) - \pi(I(e)) > a(e)$, negando (*) $\implies e \in E^{\wedge '}$ $\implies f(e) = c(e)$ q.e.d (2)

si $\pi(T(e)) - \pi(I(e)) < a(e)$, negando (**) $\implies e \in E^{\wedge ''}$ $\implies f(e) = 0$ q.e.d. (3)

Supongamos por último que disponemos de valores $\pi(x)$, $\forall x \in V$, que satisfacen las condiciones (1), (2), (3).

Definamos $g(e) = \mathbf{Max} [0, \pi(T(e)) - \pi(I(e)) - a(e)]$, $\forall e \in E$ y $g(u_p) = \pi(s) - \pi(p)$

Entonces (π, g) es solución factible de $D(v)$, el dual de $P(v)$. Las condiciones de holgura complementaria entre $P(v)$ y $D(v)$ serían:

$$i) \quad f(e) < c(e) \quad \implies \quad g(e) = 0.$$

- ii) $f(e) > 0 \implies \pi(T(e)) - \pi(I(e)) - g(e) = a(e)$
- i') $g(e) > 0 \implies f(e) = c(e)$
- ii') $\pi(T(e)) - \pi(I(e)) - g(e) < a(e) \implies f(e) = 0.$

Al verificar que efectivamente f y (π, g) satisfacen estas condiciones, se habrá demostrado que cada una de ellas es solución óptima a su respectivo problema.

- i') Si $g(e) > 0 \implies$ (por definición) $\pi(T(e)) - \pi(I(e)) > a(e)$
 \implies (por (2)) $f(e) = c(e).$
- ii') Si $\pi(T(e)) - \pi(I(e)) - g(e) < a(e) \implies$ (por definición) $g(e) = 0$
 $\implies \pi(T(e)) - \pi(I(e)) < a(e) \implies$ (por (3)) $\implies f(e) = 0.$
- i) Negando i'): Si $g(e) = 0 \implies f(e) < c(e).$
- ii) Negando ii'): Si $f(e) > 0 \implies \pi(T(e)) - \pi(I(e)) - g(e) = a(e)$ l.q.q.d.

VI.2 Primer Algoritmo (para el problema de Flujo Máximo de Costo Mínimo).

Se busca partiendo de $f^0: f(e) = 0, \forall e \in E$, construir una secuencia f^1, f^2, \dots, f^k , factibles en $R(V, E \gg \{u_p\}, a, c)$, donde cada $f^i (u_p)$ crezca monótonicamente hasta alcanzar el valor máximo $f^k (u_p) = F$. Cada solución parcial f^i será óptima al problema $P(f^i (u_p))$. Al construir $R^\wedge(f^i)$ entonces ésta será sin circuitos absorbentes.

Para pasar de f^i a f^{i+1} , se busca el camino más corto de s a p en $R^\wedge(f^i)$. Este camino corresponderá a una cadena aumentante de s a p de costo mínimo en $R(V, E \gg \{u_p\}, a, c)$, y su longitud corresponderá al aumento mínimo en costo por unidad de flujo. Si no existe camino de s a p en $R^\wedge(f^i)$, entonces no existe cadena aumentante en $R(V, E \gg \{u_p\}, a, c)$, luego f^i es óptimo.

El algoritmo, presentado de manera informal, sería el siguiente:

Algoritmo PRIMAL-DUAL: (Flujo Máximo de Costo Mínimo I);

Entradas: $V, E, I, T, s, p, u_p, f_i$;

Salidas: f , ExisteFMCM;

Comienzo

$f(e) = f_i(e), \forall e \in E$; (* f_i es un flujo inicial de costo mínimo, generalmente 0 *)

Camino := verdadero; ExisteFMCM := verdadero;

mientras (Camino y ExisteFMCM) **hacer**

CONSTRUIR $R^\wedge(f)$;

si ($\exists C$ tal que C es un Camino Mínimo de s a p (C_s^P) en $R^\wedge(f)$)

entonces

Sea $G = C \gg \{u_p\} \in R(V, E \gg \{u_p\}, a, c)$; (* G Asociado a C en $R(V, E, a, b, c)$ *)

$$e = \text{Min} [\text{Min}_{e \in E^+} [c(e) - f(e)], \text{Min}_{e \in E^-} [f(e)]]; (* e > 0 *)$$

si $(e = \infty)$

entonces

ExisteFMCM := falso

sino

caso "e ∈ E de

$$e \in E^+ : f(e) := f(e) + e;$$

$$e \in E^- : f(e) := f(e) - e;$$

```
                fin caso;  
            fin si;  
sino  
    Camino := falso;  
    fin si;  
fin mientras;  
Fin.
```

VI.3 Segundo Algoritmo (para el problema de Flujo Máximo de Costo Mínimo).

Veremos a continuación una presentación algo distinta del mismo algoritmo (además la más utilizada). Esta otra presentación está basada en la segunda parte del teorema visto anteriormente. La diferencia esencial es que la búsqueda del camino más corto de **s** a **p** en $R^{\wedge}(f)$ se simplifica gracias al conocimiento de los valores de $\pi(x)$ de la iteración anterior.

Dispondremos en cada iteración de una solución **f** factible a $P(f(u_p))$, con valores de $\pi(x)$ que satisfacen:

- (1) $0 < f(e) < c(e) \implies \pi(T(e)) - \pi(I(e)) = a(e)$.
- (2) $\pi(T(e)) - \pi(I(e)) > a(e) \implies f(e) = c(e)$.
- (3) $\pi(T(e)) - \pi(I(e)) < a(e) \implies f(e) = 0$.

Cada iteración consta de dos etapas:

i) Cambio de flujo.

Se construye $R'(V, E', c)$ donde $E' = \{ e \in E \mid \pi(T(e)) - \pi(I(e)) = a(e) \}$ y en esta red se busca el flujo máximo. (Notese que los arcos sobre esta red son aquellos que estarían en la arborescencia de distancias mínimas en $R^{\wedge}(f)$).

(Evidentemente **f** no es un flujo en R' , pero sin embargo, el procedimiento de búsqueda de cadenas aumentantes y de modificación del flujo permanece válido al ser trasladado a R).

ii) Cambio de las variables duales (π).

Cuando no se puede aumentar ya el flujo (con **f** solución óptima de $P(f(u_p))$), Sea Y el último conjunto de nodos marcados por FlujoMax:

$$s \in Y, \quad p \notin Y.$$

$$e \in E_G^+(Y) \ll E' \implies f(e) = c(e)$$

$$e \in E_G^-(Y) \ll E' \implies f(e) = 0$$

Sean

$$A^+ = \{ e \in E_G^+(Y) \mid \pi(T(e)) - \pi(I(e)) < a(e) \}$$

$$A^- = \{ e \in E_G^-(Y) \mid \pi(T(e)) - \pi(I(e)) > a(e) \}$$

Si $A^+ = \{ \} \implies \quad "$ $e \in E_G^+(Y): \quad \pi(T(e)) - \pi(I(e)) \geq a(e) \implies f(e) = c(e)$.

$A^- = \{ \} \implies \quad "$ $e \in E_G^-(Y): \quad \pi(T(e)) - \pi(I(e)) \leq a(e) \implies f(e) = 0$.

Por lo tanto, si $A^+ \gg A^- = \{ \} \implies G^+(Y)$ es un corte saturado y el flujo actual es óptimo a $P(F)$.

Si $A^+ \neq \{ \}$ ó $A^- \neq \{ \}$ entonces, sea

$$\partial = \mathbf{Min} \left\{ \mathbf{Min}_{e \in E_A^+} [a(e) - \pi(T(e)) + \pi(I(e))]; \mathbf{Min}_{e \in E_A^-} [\pi(T(e)) - \pi(I(e)) - a(e)] \right\}$$

Es claro que $\partial > 0$ y así los valores

$$\begin{array}{ll} \pi(x) = \pi(x), & \text{si } x \in Y \\ \pi(x) = \pi(x) + \partial, & \text{si } x \in Y \end{array}$$

satisfacen (1), (2) y (3) y además representan las distancias mínimas de s a x en la nueva red $R^{\wedge}(f)$.

Algoritmo FLUJOMAXCOSTOMIN: (Flujo Máximo de Costo Mínimo II);

Entradas: $V, E, I, T, s, p, u_r, a, c, f_i$;

Salidas: $f, \text{ExisteFMCM}$;

Comienzo

$f(e) := f_i(e), \forall e \in E$; (* f_i es un flujo de costo mínimo, generalmente 0 *)

Dijkstra ($V, E, I, T, a, s; \pi, A, \text{EsRaiz}$); (* Valores iniciales de π , con $a(e) > 0 \forall e$ *)

si (EsRaiz)

entonces;

repetir

ExisteFMCM := falso;

$E' = \{ e \in E / \pi(T(e)) - \pi(I(e)) = a(e) \}$;

FLUJOMAX ($V, E', I, T, u_r, s, p, b, c, f; f, Y, \text{MODIF}, \text{ExisteFMCM}$);

si (ExisteFMCM) (* Existe un flujo máximo por esta red *)

entonces

Construir $G^+(Y) := \{ e \in E / I(e) \in Y, T(e) \in Y \}$;

Construir $G^-(Y) := \{ e \in E / I(e) \in Y, T(e) \notin Y \}$;

Construir $A^+ := \{ e \in G^+(Y) / \pi(T(e)) - \pi(I(e)) < a(e) \}$;

Construir $A^- := \{ e \in G^-(Y) / \pi(T(e)) - \pi(I(e)) > a(e) \}$;

si ($A^+ \neq \emptyset \vee A^- \neq \emptyset$)

entonces

$\partial^+ := \min_{e \in A^+} [a(e) - \pi(T(e)) + \pi(I(e))]$; (* Si $A^+ = \emptyset, \partial^+ = \infty$ *)

$\partial^- := \min_{e \in A^-} [-\pi(T(e)) + \pi(I(e)) - a(e)]$; (* Si $A^- = \emptyset, \partial^- = \infty$ *)

$\partial := \min \{ \partial^+, \partial^- \}$; (* $\partial > 0$ *)

para todo $\{ x \in V / x \in Y \}$ hacer

$\pi(x) := \pi(x) + \partial$;

fin para;

fin si;

fin si;

hasta que (no ExisteFMCM) ó ($A^+ \neq \emptyset \vee A^- \neq \emptyset$);

fin si;

Fin.