

2 ejemplares  
N. 10

**Universidad Nacional Autónoma de México**  
**FACULTAD DE CIENCIAS**

*Implantación del Problema General de Flujo  
Máximo a Costo Mínimo a Través del  
Algoritmo de Floyd*

T E S I S

Que para obtener el Título de

A C T U A R I O

P r e s e n t a

GRACIELA GONZALEZ HITTA

15026



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**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 esta 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.

## INDICE

### INTRODUCCION

#### I GRAFICAS

- 1.1.- Caminós y Cadenas
- 1.2.- Lazos, Circuitos y Ciclos.
- 1.3.- Gráficas Parciales y Subgráficas
- 1.4.- Representación Matricial de una Gráfica

#### II REDES

- 2.1.- Introducción
- 2.2.- Redes.-definición
- 2.3.- Redes con entradas y salidas múltiples
- 2.4.- Capacidad de vértices

#### III FLUJOS EN REDES

- 3.1.- Introducción
- 3.2.- Flujo Factible

#### IV FLUJO MAXIMO

- 4.1.- Introducción
- 4.2.- Cortes
- 4.3.- Algoritmo de Ford-Fulkerson

#### V BASES QUE INTERVIENEN EN EL PLANTEAMIENTO DEL PROBLEMA DE FLUJO A COSTO MINIMO

- 5.1.- Introducción
- 5.2.- Construcción de una Gráfica Incremental
- 5.3.- Descomposición de un Patrón de Flujo.

VI PROBLEMA DE FLUJO A COSTO MINIMO

6.1.- Introducción

6.2.- Algoritmo de Floyd

6.3.- Matriz  $\theta$  de Caminos Mínimos

6.4.- Algoritmo Basado en la determinación de Circuitos Negativos

6.5.- Descripción del Algoritmo

VII RESUMEN

7.1.- Importancia del Algoritmo de Floyd dentro del Proceso

VIII EJEMPLOS

IX CONCLUSIONES

X REFERENCIAS

XI APENDICE

1.- Algoritmo para Minimizar el Costo de un flujo Basado  
\* en la determinación de circuitos negativos

2.- Diagrama de Bloque de la programación del Proceso.

2.1.- Comentarios referentes a la estructura del programa

XII DIAGRAMAS DE FLUJO DEL PROCESO

## I N T R O D U C C I O N

Uno de los problemas más interesantes e importantes dentro de la teoría de redes es el de determinar el valor del flujo máximo que puede ser transportado de una entrada  $s$  a una salida  $t$  dadas en una red.

Encontramos que es tan vasta la literatura de flujos en Redes que existen diferentes formulaciones del problema de flujo máximo en una red, así como muchas y muy variadas las aplicaciones que se pueden dar.

Debido a esto el estudio de teoría de redes en este campo se ha extendido rápidamente durante los últimos años, teniendo como estímulo el gran desarrollo y la rapidez de las máquinas computadoras.

En el presente trabajo hacemos una formulación al problema de Máximo flujo Mínimo Costo basandonos en los conceptos fundamentales de teoría de redes.

Proponemos una solución a este problema en dos partes:

- i) Determinar el flujo máximo dentro de una red.
- ii) Minimizar el costo de transporte de un flujo.

en forma de subrutinas y se aplica a dos ejemplos.

Para la solución de (i) utilizamos el algoritmo de FORD-FULKERSON, y para la solución de (ii) utilizamos un algoritmo basado en la determinación de circuitos negativos.

El desarrollo del trabajo es de la siguiente manera :

Primeramente introducimos los conceptos fundamentales de teoría de Redes así como las propiedades de las gráficas, que nos servirán más adelante como herramienta para plantear formalmente desde el punto de vista matemático nuestro problema.

En seguida se describe el algoritmo usado para resolver el problema de flujo máximo. En la sección siguiente se desarrollan los procesos de construcción de una red incremental así como de la descomposición de un patrón de flujo.

Esta sección contribuye a una mejor comprensión del funcionamiento del algoritmo usado para minimizar el costo de transportación de un flujo, dicho algoritmo está basado en la determinación de circuitos negativos dentro de una Red a través de los cuales se va disminuyendo el costo de transporte.

En seguida se da el planteamiento del problema de FLUJO A COSTO MINIMO y se describe el algoritmo utilizado para su solución (algoritmo de Floyd).

Finalmente se da el proceso de Máximo-flujo Mínimo-Costo programado siguiendo el proceso descrito en la sección 6.1 \_

## LISTA DE SIMBOLOS

$G = (X, A)$	Gráfica
$A$	Conjunto de arcos
$X$	Conjunto de vértices
$B = [b_{ij}]$	Matriz de Adyacencia
$D = [d_{ij}]$	Matriz de incidencia
$(x_i, y_i)$	$i$ -ésimo arco
$(x, y)$	arco con dirección $x$ a $y$
$\uparrow(x_i)$	Conjunto de correspondencias
	Conjunto de vértices $x_j$
	tales que $(x_i, x_j) \in A$
$\uparrow^-(x)$	Conjunto de vértices $x_j$
	tales que $(x_j, x_i) \in A$
$a_j$	$j$ -ésimo arco
$x_i$	$i$ -ésimo vértice
$s$	Fuente(entrada) de una Red
$t$	Deposito(salida) de una Red
$G = (X, A, d)$	Red
$d$	Función de correspondencia
	de $A$ en $\mathbb{R}$
$q(x)$	Capacidad del vértice $x$



$q(x,y)$	Capacidad del arco $(x,y)$
$\xi$	Vector de flujo
	$\xi = (f_1, f_2, \dots, f_m)$
$f_i$	Flujo en el arco $a_i$
$f(x,y)$	Flujo en el arco $(x,y)$
	$f(x,y) = \xi(x,y)$
$(Y; \bar{Y})$	Corte que separa a s de t en una red dada.
$C(Y; \bar{Y})$	Capacidad del corte $(; \bar{Y})$
$\xi(Y; \bar{Y})$	Flujo del corte $(Y; \bar{Y})$
$c_{ij}$	Costo en el arco $(i,j)$
$c(x,y)$	Costo en el arco $(x,y)$
$C = [c_{ij}]$	Matriz de costos mínimos
$\theta = [\theta_{ij}]$	Matriz de caminos mínimos
$\phi$	Circuito
$c(\phi)$	Costo del circuito
$n$	número de nodos de una red
$m$	número de arcos de una red
$G^u(\xi)$	Red incremental
$v$	valor de un flujo

## I.- GRÁFICAS.- Definición

Una gráfica es una colección de puntos llamados vértices denotados por  $x_1, x_2, \dots, x_n$  los cuales forman el conjunto  $X$ , y una colección de líneas  $a_1, a_2, \dots, a_m$  las cuales forman el conjunto  $A$ ; tendremos que cada línea del conjunto  $A$  une a dos puntos del conjunto  $X$ , dichos puntos pueden ser diferentes ó coincidir en un mismo punto.

Al referirnos a una gráfica la denotaremos por  $G = (X, A)$ .

Si las líneas de  $A$  tienen una dirección (usualmente indicada por una flecha) serán llamadas arcos y la gráfica formada será llamada gráfica orientada (fig 1.1).

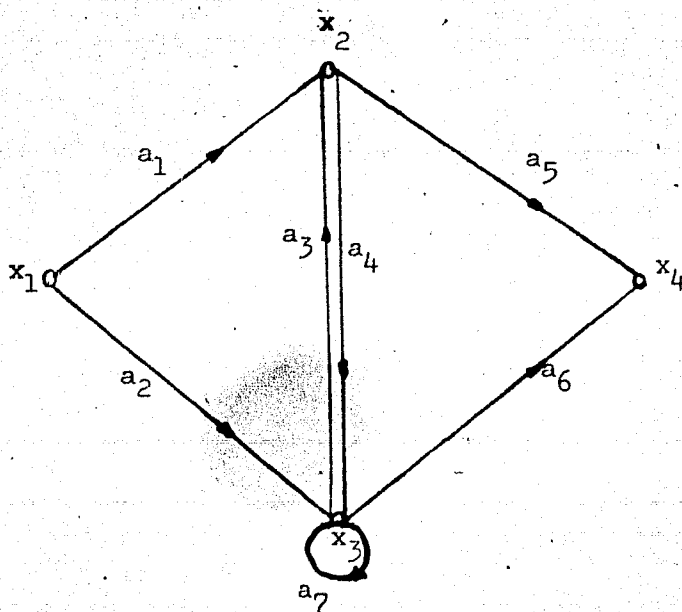


FIGURA 1.1

Si las líneas del conjunto  $A$  no tienen orientación serán llamadas ejes y la gráfica formada será una gráfica no orientada (fig 1.2).

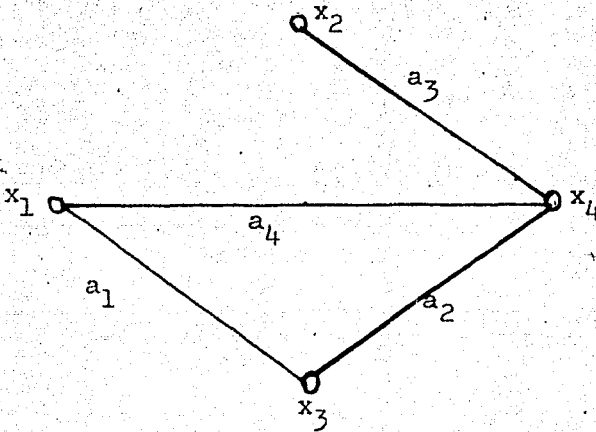


FIGURA 1.2

Ocasionalmente se pueden considerar también gráficas mixtas en el caso en el que algunas líneas del conjunto  $A$  estén dirigidas, pero otras no.

El orden de una gráfica  $G = (X, A)$  es igual al número de elementos de  $X$ .

Dado un arco de la forma  $(x, y)$  se dice que  $x$  es la extremidad inicial del arco, e  $y$  es la extremidad final; entonces el sentido de un arco estará dado por la dirección del primer -

vértice al segundo. Así en la fig 1.1  $(x_1, x_2)$  se refiere al arco  $a_1$  y  $(x_2, x_3)$  al arco  $a_3$ .

Otra alternativa para describir la dirección en gráficas orientadas es asociar al conjunto de vértices  $X$  una función

$\uparrow$ , la cual muestra como los vértices están relacionados unos a otros.  $\uparrow$  es un mapeo del conjunto  $X$  en  $X$ , la gráfica se denota ahora por  $G = (X, \uparrow)$ .

En el ejemplo de la fig 1.1 tenemos:

$\uparrow(x_1) = \{x_2, x_3\}$  i.e.  $x_2$  y  $x_3$  son los vértices finales de los arcos cuyo nodo inicial es  $x_1$ .

$\uparrow(x_2) = \{x_3, x_4\}$

$\uparrow(x_3) = \{x_2, x_4, x_3\}$

$\uparrow(x_4) = \emptyset$  donde  $\emptyset$  denota al conjunto vacío.

En el caso de gráficas no orientadas y gráficas mixtas la función  $\uparrow$  se aplicará a una gráfica orientada equivalente, en la que cada eje será remplazado por dos arcos en direcciones opuestas.

Aplicando lo anterior al ejemplo de la fig 1.2 tenemos :

$\uparrow(x_1) = \{x_4, x_3\}$        $\uparrow(x_2) = x_4$     etc.

En general definiremos

$\uparrow(x_i) = \{x_j \in X \mid (x_i, x_j) \in A\}$  i.e el conjunto  $\uparrow(x_i)$

esta formado por todos los vértices  $x_j$  que esten en  $X$  tales que formen un arco de la forma  $(x_i, x_j)$  contenido en  $A$ .

$$\text{Asi mismo } \uparrow^{-1}(x_i) = \{x_k \in X \mid (x_k, x_i) \in A\}$$

$\uparrow^{-1}(x_i)$  es la función inversa de  $\uparrow$ .

En la fig 1.1 tenemos :

$$\uparrow^{-1}(x_3) = \{x_2, x_1, x_3\}$$

En gráficas no orientadas tendremos que :

$$\uparrow^{-1}(x_i) = \uparrow(x_i) \forall x_i \in X.$$

lo anterior es debido a que a cada arco no orientado le asociamos dos arcos orientados, en direcciones opuestas, asi el conjunto de nodos finales respecto al nodo  $x_i$  es igual al conjunto de nodos iniciales respecto al nodo  $x_i$ .

Ejemplo :

En la fig 1.2 tenemos que:

$$\uparrow(x_1) = \{x_3, x_4\}$$

$$\uparrow^{-1}(x_1) = \{x_3, x_4\}$$

For lo tanto  $\uparrow(x_1) = \uparrow^{-1}(x_1)$

Cuando la función  $\uparrow$  no se aplica sobre un único vértice sino sobre un conjunto de vértices, sea el caso de  $X_q = \{x_1, x_2, \dots, x_q\}$  tendremos lo siguiente:

$$\uparrow(X_q) = \uparrow(x_1) \cup \uparrow(x_2) \cup \dots \cup \uparrow(x_q)$$

•• para la gráfica de la fig 1.1  $\uparrow(\{x_1, x_4\}) = \{x_3, x_2\}$

y  $\uparrow(\{x_1, x_2\}) = \{x_2, x_3, x_4\}$

## 1.1 CAMINOS Y CADENAS

Una cadena es una sucesión de arcos ó ejes  $(a_1, a_2, \dots, a_{i-1}, a_i, a_{i+1}, \dots, a_k)$  tal que la extremidad final del arco  $a_i$  coincide con una extremidad del arco  $a_{i+1}$  y la otra extremidad coincide con una extremidad del arco  $a_{i-1}$ .

Así, si dados dos vértices  $x_i, x_j$  si existe una cadena de la forma  $(a_1, a_{i+1}, a_{i+2}, \dots, a_{j-1}, a_j)$  donde una extremidad del arco  $a_1$  es  $x_i$ , y una extremidad del arco  $a_j$  es  $x_j$ , entonces diremos que existe una cadena que une a los vértices  $x_i, x_j$ .

No es necesario que la orientación que tienen los arcos de la gráfica sea la misma que la orientación de la cadena; por ejemplo en la fig 1.1  $(a_1, a_3, a_2)$  es una cadena.

Una cadena que no contiene dos veces el mismo arco se denomina CADENA SIMPLE y una cadena que no pasa dos veces por el mismo vértice se llama CADENA ELEMENTAL.

Ejemplos para la fig 1.1

$(a_1, a_3, a_2, a_6)$  no es una cadena.

$(a_1, a_3, a_3, a_4, a_2)$  es una cadena, pero no es simple ya que contiene dos veces el arco  $a_3$ , tampoco es una cadena elemental ya que pasa dos veces por los vértices  $x_2$  y  $x_3$ .

$(a_1, a_5)$  es una cadena simple y elemental.

Utilizando ahora el concepto de cadena definido anteriormente diremos:

$G = (X, A)$  es una gráfica Conexo si dados cualesquiera dos vértices  $x_i, x_j$  siempre existe una cadena que los une.

Definición.- Un camino es una sucesión de arcos  $(a_1, a_2, \dots, a_{i-1}, a_i, a_{i+1}, \dots, a_k)$  tales que la extremidad inicial de  $a_i$  coincide con la extremidad final de  $a_{i-1}$ , y la extremidad final de  $a_i$  coincide con la extremidad inicial de  $a_{i+1}$ .

En un camino si es muy importante la orientación de los arcos.

Ejemplo en la fig 1.1  $(a_1, a_4)$  no es un camino, pero  $(a_1, a_5)$  si lo es.

Similarmente a los conceptos de cadena simple y cadena elemental tenemos:

Un CAMINO SIMPLE es un camino que no contiene dos veces un mismo arco, y un CAMINO ELEMENTAL es un camino que no pasa dos veces por el mismo vértice.

Ejemplo en la fig 1.1  $(a_3, a_4, a_3)$  es un camino que no es simple ni elemental.

## 1.2 LAZOS, CIRCUITOS Y CICLOS

Un lazo es un arco en el cual el vértice inicial y el vértice-final coinciden.

Ejemplo en el fig 1.1 el arco  $a_7$  es un lazo i.e es un arco de la forma  $(x_i, x_i)$ .

NOTA.- En este trabajo no se consideran gráficas con este tipo de arcos.

Un ciclo es una cadena simple cuyo vértice inicial y final coinciden.

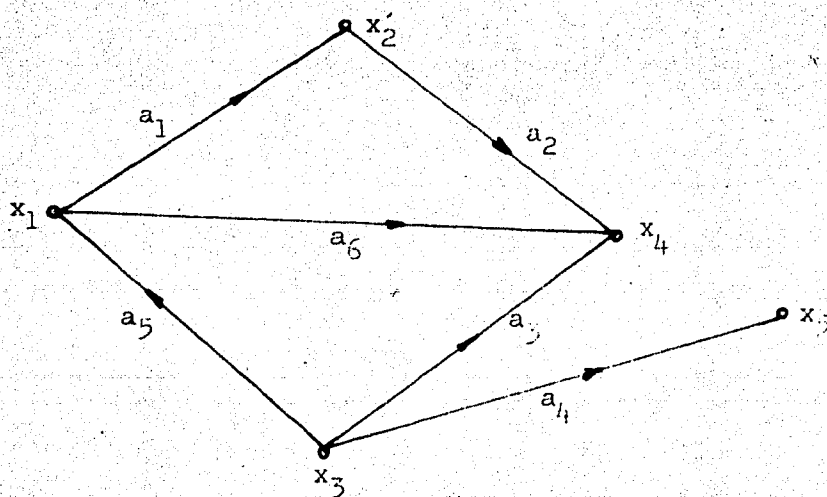


FIG 1.3

$(a_2, a_3, a_5, a_1)$  en la fig 1.3 es un ciclo; el vértice inicial es igual al vértice final,  $x_1$ .

$(a_6, a_3, a_5)$  en la fig 1.3 es otro ciclo con  $x_1$ , vértice inicial - final.

Un CIRCUITO, es un camino simple, que no pasa dos veces por el mismo vértice excepto el inicial y el final que son el mismo. En la gráfica de la fig 1.3 no hay circuitos.

Dada una gráfica  $G$  no orientada, un circuito EULERIANO (camino) es un circuito (camino) el cual atravieza todo eje (arco) de  $G$  una y solo una vez.



No todas las gráficas contienen circuito Euleriano (ver ejemplo de la gráfica fig 1.4).

Si un circuito Euleriano (ó camino) existe, significa que podremos dibujar la gráfica siguiendo este circuito sin despegar el lápiz del papel y sin pasar dos veces por el mismo trazo.

La gráfica de la fig 1.5 tiene un circuito Euleriano dado por la siguiente secuencia (comenzando por el vértice  $x_1$ ).

$a_1, a_2, a_3, a_4, a_{15}, a_{14}, a_{13}, a_{12}, a_{11}, a_{16}, a_{17}, a_{10}, a_9, a_8, a_5, a_7, a_6$

La dirección que sigue cada arco es indicada por las flechas de la fig 1.5.

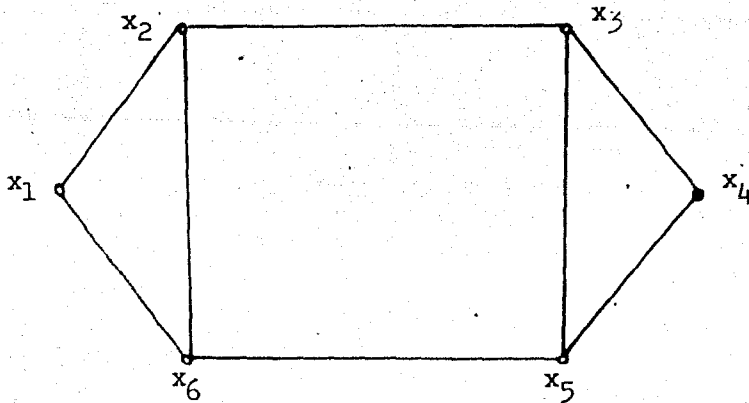


FIG 1.4.- Gráfica sin circuito Euleriano.

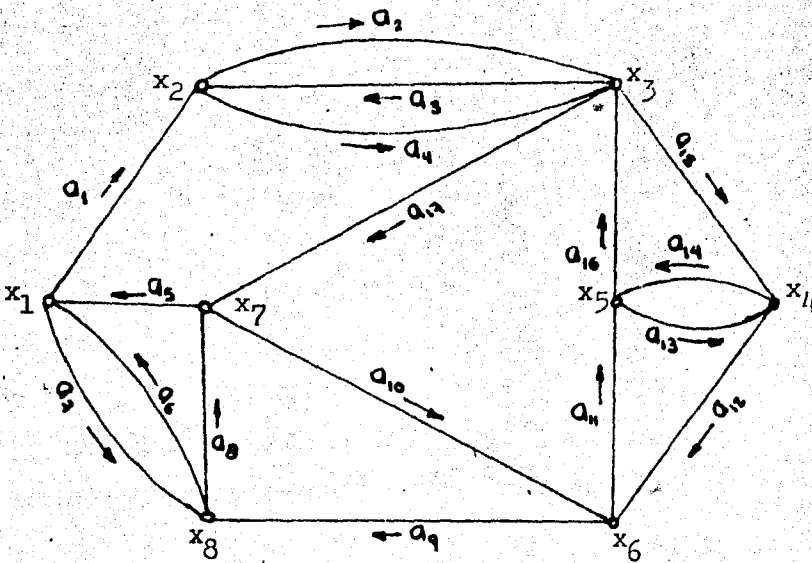


FIG 1.5.- Gráfica con un circuito Euleriano.

### 1.3 GRAFICAS PARCIALES Y SUBGRAFICAS

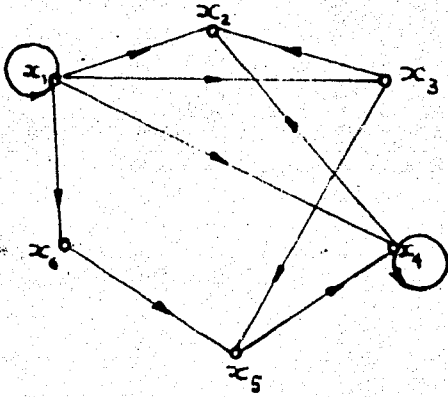
Definición.- Dada una gráfica  $G = (X, A)$  y un subconjunto de arcos  $V \subset A$ ; entonces  $G' = (X, V)$  es una gráfica parcial de  $G$ .

Por lo tanto una gráfica parcial es una gráfica con el mismo número de vertices, pero solamente con un subconjunto de arcos de la gráfica original.

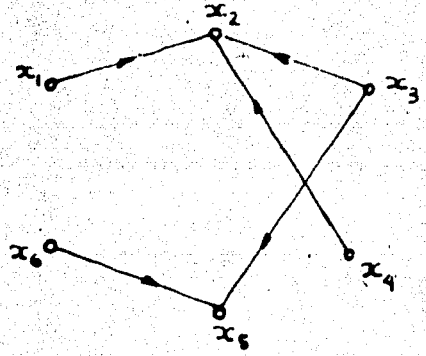
Definición.- Dada una gráfica  $G = (X, A)$  y un subconjunto  $B \subset X$  y si  $UB = \{(x_i, x_j) \in A \mid x_i, x_j \in B\}$  entonces  $G' = (B, UB)$  es una subgráfica de  $G$ . Por lo tanto una subgráfica contiene solo un

subconjunto  $B$  del conjunto  $X$  de vértices de la gráfica original y contiene a todos los arcos cuyo vértices inicial y final estén ambos en el conjunto  $B$ .

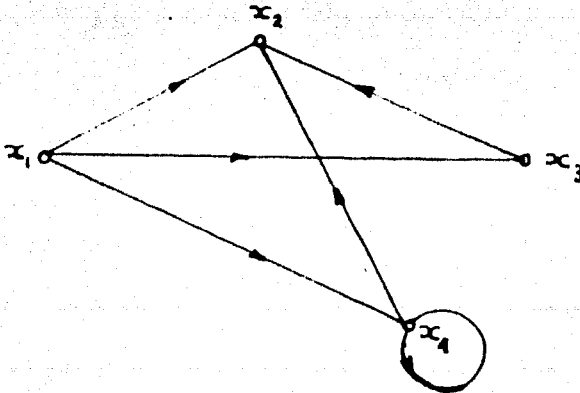
Ejemplo:



(a) Gráfica



(b) Gráfica Parcial



(c) Subgráfica

FIG 1.6

En la fig 1.6 (b) se muestra una gráfica parcial de la gráfica mostrada en la fig 1.6 (a), también se da un ejemplo de una subgráfica en la fig 1.6 (c) conteniendo solo los vértices  $x_1, x_2, x_3, x_4$ .

#### 1.4 REPRESENTACION MATRICIAL DE UNA GRAFICA

Una manera conveniente de representar una gráfica algebraicamente es por medio del uso de matrices de la siguiente manera :

MATRIZ ADYACENTE.

Dada una gráfica  $G = (X, A)$ , su matriz adyacente es  $B = [b_{ij}]$  y esta dada por :

$$b_{ij} = \begin{cases} 1 & \text{si el arco } (i, j) \in A \\ 0 & \text{si el arco } (i, j) \notin A \end{cases}$$

Ejemplo.

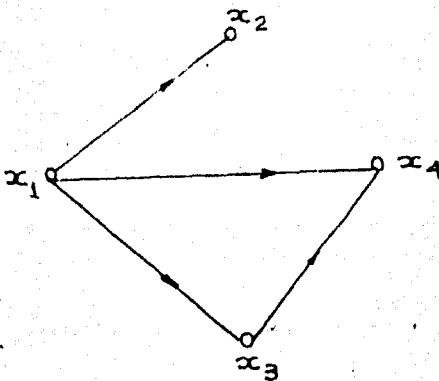


FIG 1.7

La matriz adyacente de la gráfica mostrada en la fig 1.7 es:

$$B = \begin{array}{c} \begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \end{array} \begin{array}{c} x_1 \quad x_2 \quad x_3 \quad x_4 \\ \begin{array}{|c|c|c|c|} \hline 0 & 1 & 1 & 1 \\ \hline 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 \\ \hline 0 & 0 & 0 & 0 \\ \hline \end{array} \end{array}$$

"La matriz de adyacencia define completamente la estructura de una gráfica.

Por medio de esta matriz podemos conocer el número total de arcos que llegan a cada vértice, al sumar el total de elementos asociados a cada columna  $x_j$ , así como el número total de arcos que salen de cada vértice al sumar el total de elementos de cada renglón  $x_i$ .

Por ejemplo: El número total de arcos que llegan al vértice  $x_1 = 0$ , y el número total de arcos que salen de  $x_1 = 3$ .

También podemos determinar por medio de esta matriz el conjunto  $\uparrow(x_i) =$  al conjunto de columnas cuya entrada es 1 en el renglón  $x_i$ , similarmente  $\uparrow^{-1}(x_i) =$  conjunto de renglones cuya entrada es 1 en la columna  $x_j$ ." (1)

Ejemplo :

$$\begin{aligned} \uparrow(1) &= \{x_2, x_3, x_4\} \\ \uparrow^{-1}(1) &= \emptyset \\ \uparrow(2) &= \emptyset \\ \uparrow^{-1}(2) &= \{x_1\} \end{aligned}$$

(1) tomado de Christofides (I) pag.13

## MATRIZ DE INCIDENCIA.

Toda la información acerca de la estructura de una gráfica se puede determinar a partir de esta matriz, la cual se define así:

Dada una gráfica  $G = (X, A)$  con  $n$  vértices y  $m$  arcos, la matriz de incidencia de  $G$  se denota  $D = [d_{ij}]$  con  $D_{n \times m}$  donde

$$d_{ij} = \begin{cases} 1 & \text{si } x_i \text{ es el vértice inicial del arco } a_j \\ -1 & \text{si } x_i \text{ es el vértice final del arco } a_j \\ 0 & \text{en otro caso} \end{cases}$$

Ejemplo:

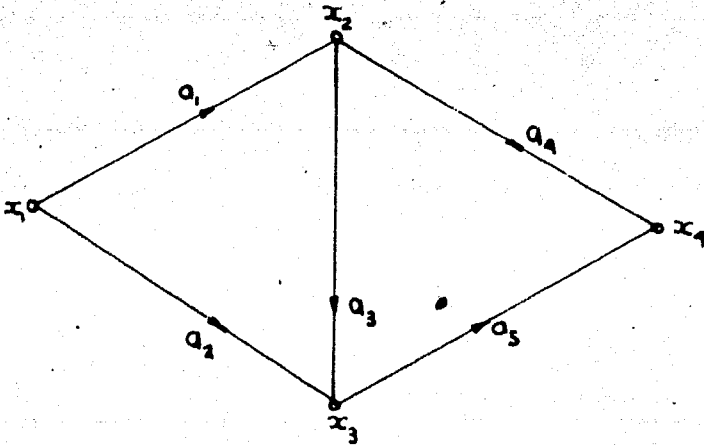


FIG 1.8

A continuación se da la matriz de incidencia para la gráfica de la fig 1.8 .

$$D = \begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \end{array} \begin{array}{ccccc} a_1 & a_2 & a_3 & a_4 & a_5 \\ \hline 1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 1 & 0 \\ 0 & -1 & -1 & 0 & 1 \\ 0 & 0 & 0 & -1 & -1 \end{array}$$

Como se observa en esta matriz de incidencia, y en general esto sucede en todas las gráficas; cada arco es adyacente exactamente a dos vértices, por esto, cada columna de la matriz de incidencia contiene un 1 y un -1.

## II.- REDES

### 2.1 Introducción.

Debido a que todas las propiedades de las gráficas son válidas dentro de la estructura que constituye una red, podemos decir que, en general los conceptos básicos de teoría de gráficas formarán las bases fundamentales en las que se apoya la conceptualización de teoría de Redes.

De esta manera la teoría de gráficas constituirá un soporte a través del cual nosotros podemos dar solución a los diferentes problemas que se presenten dentro de una red.

El tipo de red sobre el cual trabajaremos en el presente trabajo, será aquel en el que la red solo tenga asociado un solo nodo denominado ENTRADA (s) así como un solo nodo denominado SALIDA (t).

En dicho tipo de red solo se tendrán limitaciones de capacidad sobre los arcos.

La razón por la cual se propone trabajar con este tipo de red es por que resulta mucho más sencillo el manejo de datos dentro de los algoritmos que se utilizan para la obtención de soluciones a nuestros problemas.

En particular se demuestra mas adelante que cualquier tipo de red se puede transformar al tipo de red que aquí se trabaja sin que por esto se altere su estructura original.

En las secciones 2.3 y 2.4 se explica el procedimiento que se sigue para transformar una red con características diferen-



tes, a una red del tipo mas sencillo, que es con la cual trabajaremos.

## 2.2 REDES.- Definición

Una red es una gráfica en la cual se ha definido una aplicación de  $A$  en  $\mathbb{R}$  de la siguiente manera:

A cada arco  $(x,y) \in A$  se le asocia una cantidad la cual puede ser positiva, negativa ó nula.

Se puede denotar a una red como :

$G = (X,A,d)$  donde  $d$ , es una función  $d:A \longrightarrow \mathbb{R}$ ;  $d$  puede tener diferentes interpretaciones tales como:

- la longitud del arco  $(x,y)$
- costo de usar el arco  $(x,y)$
- un flujo
- una capacidad, etc.

Los arcos de una red pueden representar:

- vias de comunicación
- líneas aéreas
- canales
- rutas de transporte, etc.

Los nodos ó vértices de una red pueden representar:

- aeropuertos
- ciudades
- centro de trabajo
- etapas de un proyecto, etc

### 2.3 REDES CON ENTRADAS Y SALIDAS MÚLTIPLES.

Uno de los problemas frecuentes en la transportación de un flujo a través de una red se presenta cuando existen entradas y salidas múltiples.

Este problema es fácil de resolver si trabajamos con una red "equivalente" a la red original; definida mediante el siguiente proceso:

Si tenemos una red con  $s$  vértices entrada, i.e

$$S = \{x_1, x_2, \dots, x_s\} \text{ y } t \text{ vértices salida i.e } T = \{y_1, y_2, \dots, y_t\}$$

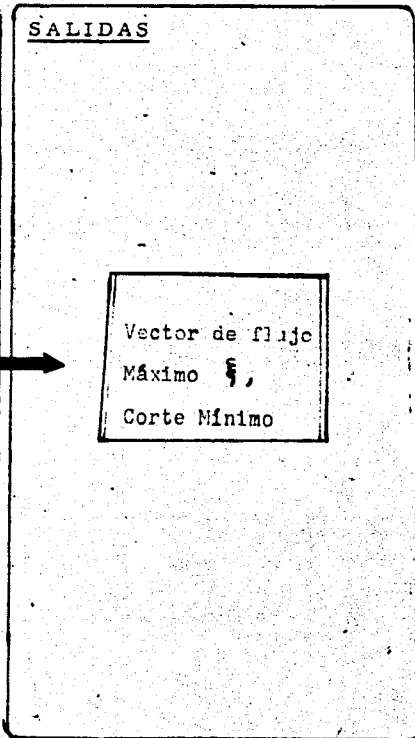
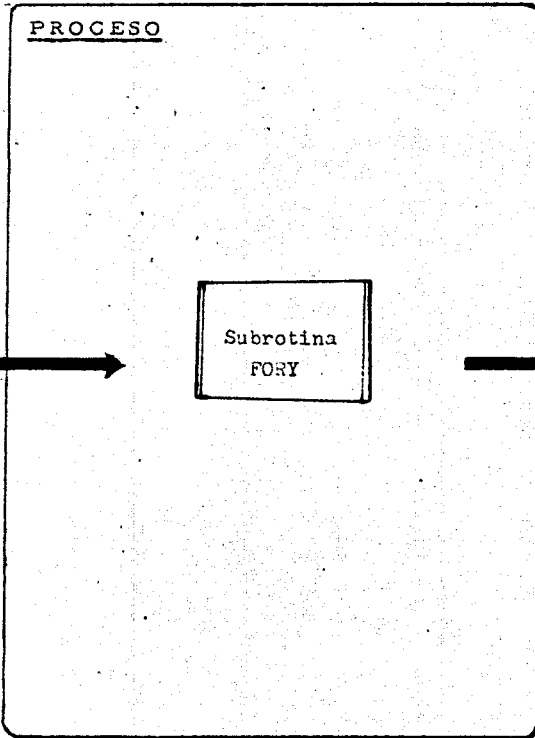
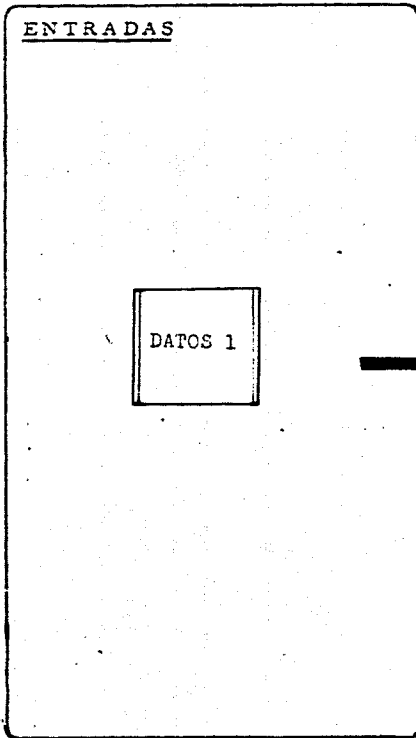
Podemos convertir esta red a una nueva red "equivalente" al agregar a la red original un vértice artificial ( $s_0$ ) a la entrada, y un vértice artificial ( $t_0$ ) a la salida, de la siguiente manera:

Asociamos un arco del vértice  $s_0$  a cada uno de los vértices del conjunto  $S$ , así como un arco de cada uno de los vértices del conjunto  $T$  al vértice  $t_0$ .

HOJA PARA DESCRIPCION  
FUNCIONAL DE PROGRAMAS

IDENTIFICACION

SISTEMA	_____
PROGRAMA	SUBROUTINA FORY
FUNCION GENERAL	ENCONTRAR EL FLUJO MAXIMO EN UNA RED

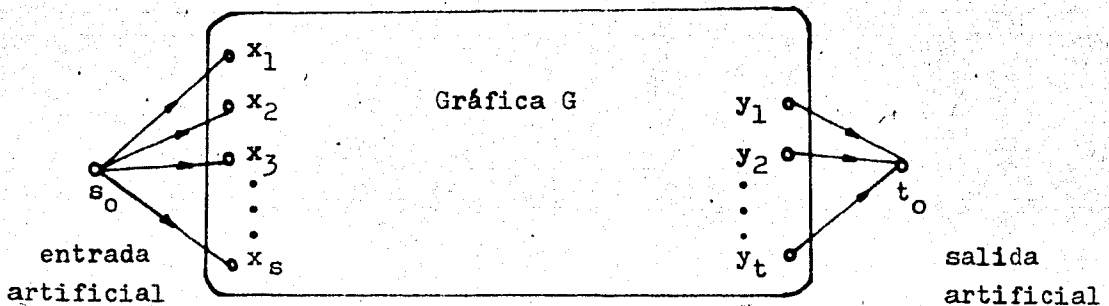


OBSERVACIONES: \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

La capacidad de los arcos que van de  $s_0$  a  $S$  puede ser infinita ó bien puede ser igual a la disponibilidad que tengamos en nuestra fuente (entrada).

La capacidad de los arcos que van de  $T$  a  $t_0$  estará limitada por la demanda existente asociada a cada salida de  $T$ , ó bien puede ser infinita.

A continuación se muestra a través de un esquema el procedimiento.



$$S = \{\text{entradas}\}$$

$$T = \{\text{salidas}\}$$

La figura muestra como múltiples entradas y salidas pueden ser reducidas a una sola entrada y salida.

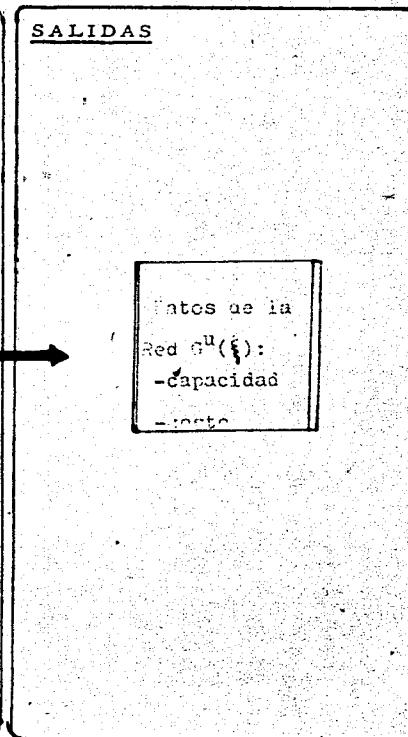
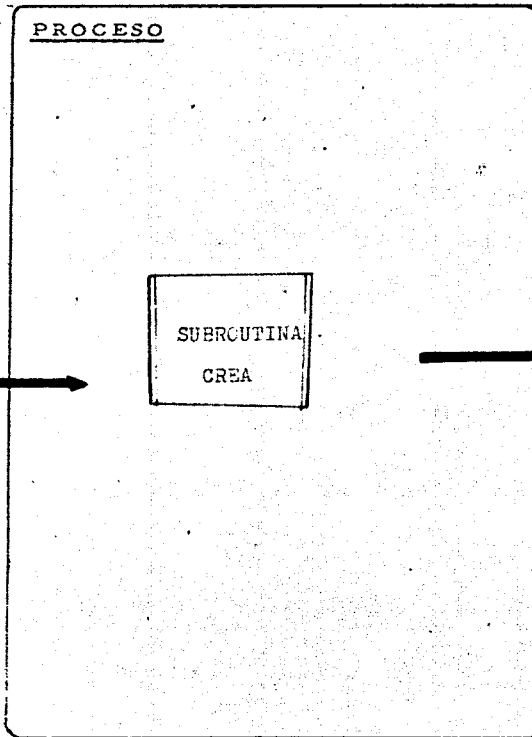
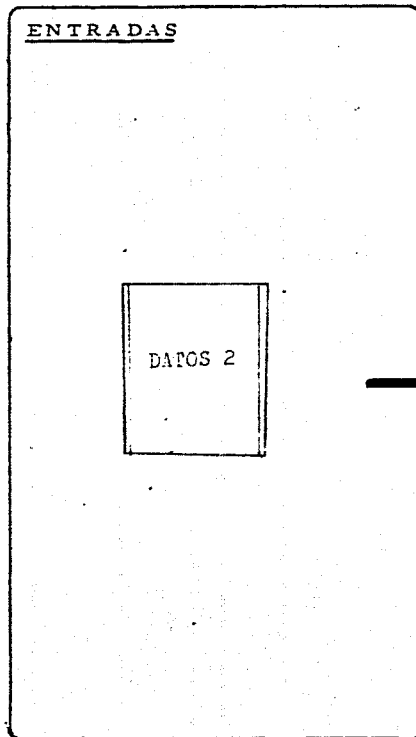
#### 2.4 CAPACIDAD DE VERTICES.

Hasta ahora nos hemos referido a redes en las cuales suponemos que existen limitaciones de capacidad solo sobre los arcos.

HOJA PARA DESCRIPCION  
FUNCIONAL DE PROGRAMAS

IDENTIFICACION

SISTEMA \_\_\_\_\_  
PROGRAMA \_\_\_\_\_ SUBROUTINA CREA  
FUNCION GENERAL \_\_\_\_\_ GENERAR LA RED INCREMENTAL  $G^u(\xi)$



OBSERVACIONES: \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

Es posible encontrar limitaciones de capacidad también sobre los vértices de una red, por lo cual damos a continuación un procedimiento con el que podemos transformar una red con limitaciones de capacidad sobre sus vértices a una red con restricciones de capacidad solo sobre sus arcos.

Sea  $G = (X, A, q)$  una red en la que cada vértice  $x \in X$  tiene una capacidad  $q(x) \geq 0$ , y cada arco  $(x, y) \in A$  tiene una capacidad  $q(x, y) \geq 0$ .

Esta red se puede transformar fácilmente a una red "equivalente" en la cual solo se tendrán limitaciones de capacidad asociadas a los arcos.

Para lograr esta transformación, definimos una nueva red  $G' = (X', A', c')$  como sigue:

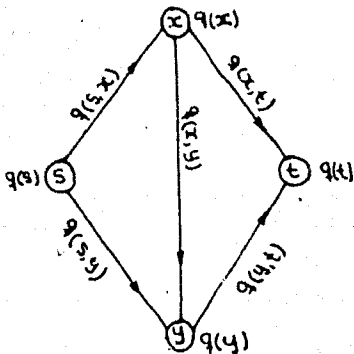
A todo  $x \in X$  se le hace corresponder dos vértices  $x', x'' \in X'$ , si  $(x, y) \in A$  entonces  $(x'', y') \in A'$ ;  $\forall x \in X (x', x'') \in A'$ .

Se define una función  $c'$  de la siguiente manera:

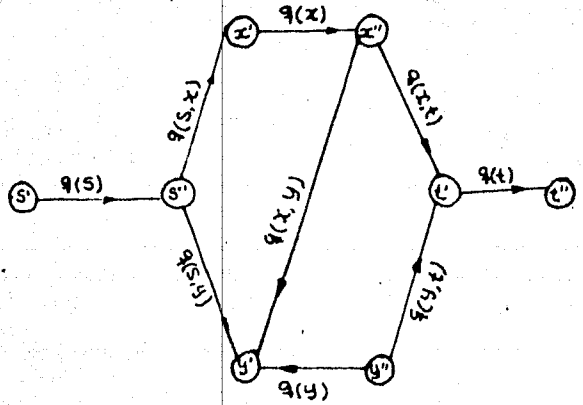
$$c'(x'', y') = q(x, y) \quad \forall (x, y) \in A.$$

$$c'(x', x'') = q(x) \quad \forall x \in X.$$

Ejemplo:



$G = (X, A, q)$

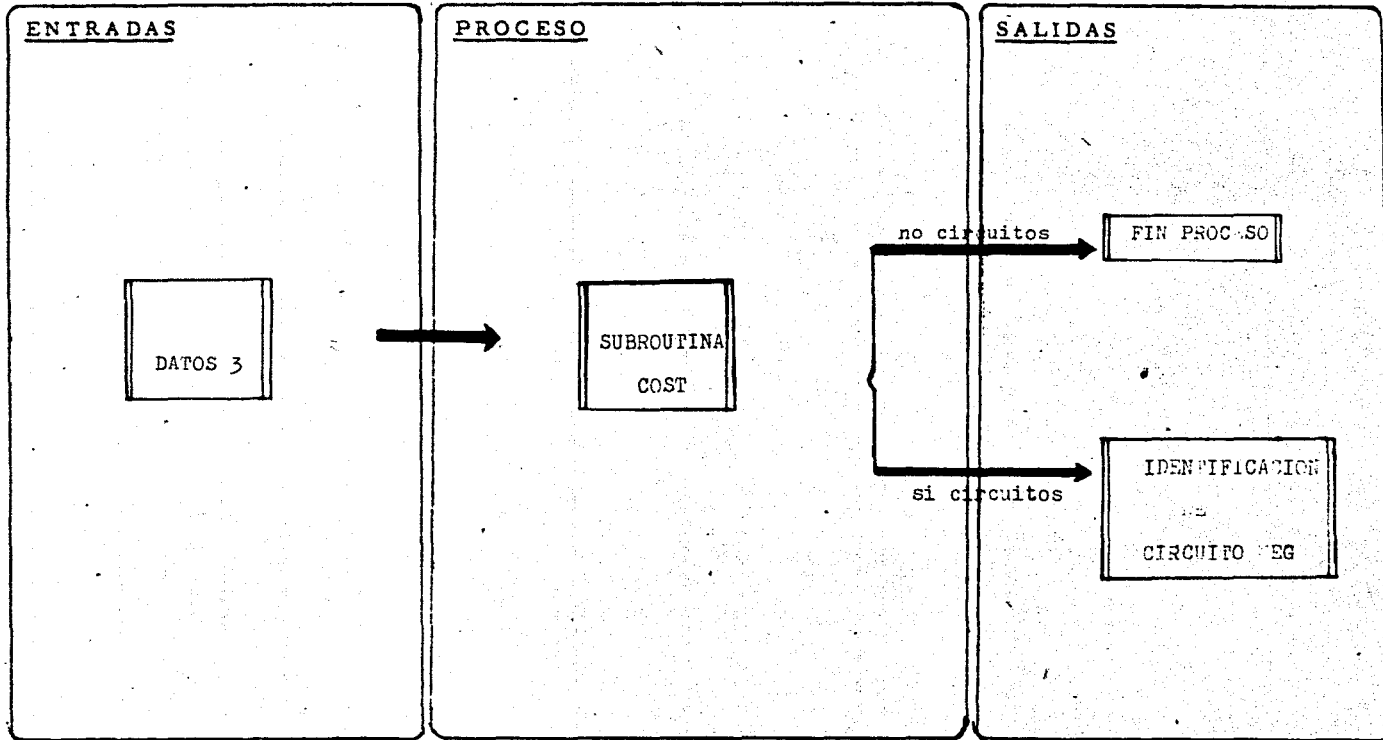


$G' = (X', A', c')$

IDENTIFICACION

HOJA PARA DESCRIPCION  
FUNCIONAL DE PROGRAMAS

SISTEMA \_\_\_\_\_  
PROGRAMA SUBROUTINA COST  
FUNCION GENERAL BUSQUIDA DE CIRCUITOS NEGATIVOS



OBSERVACIONES: \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

En conclusión lo que se hace es transferir la capacidad que opera sobre cada vértice de la red a un arco, de tal manera que la nueva red  $G'$  obtenida sera un tipo de red con restricciones de capacidad solo sobre los arcos.

NOTA.- Se aclara que de aqui en adelante se considerarán únicamente redes con una sola entrada (s) y una sola salida (t), y restricciones de capacidad asociadas solo sobre los arcos. i.e. en el caso de que alguna transformación sea necesaria, se supone realizada.

El número de nodos de una red lo denotaremos por  $n$ , y el número de arcos se denotara por  $m$ .



## III.- FLUJOS EN REDES

## 3.1 Introducción.

En este capítulo se presentan los conceptos necesarios para plantear un problema de flujo máximo formalmente, así como las bases sobre las cuales se aplican los algoritmos que nos permitirán obtener soluciones a nuestros problemas.

## 3.2 FLUJO FACTIBLE.- Definición.

Sea  $G = (X, A, d)$  una red dada, donde se tiene que  $(X, A)$  es una gráfica conexa, y  $d$  es una aplicación de  $A$  en  $\mathbb{R}^+$ , a cada arco le hace corresponder un número en  $\mathbb{R}^+$ , dicho número se llama la capacidad del arco  $(x, y)$  y se denotará como  $q(x, y) \forall (x, y) \in A$ .

Definición.- Sean  $s$  y  $t$  dos vértices diferentes  $s, t \in X$ , un flujo factible de valor  $v$  de  $s$  a  $t$  en  $G$  es una aplicación  $f$  de  $A$  en  $\mathbb{R}^+$ , satisface lo siguiente:

$$3.21 \quad \sum_y f(x, y) - \sum_y f(y, x) = \begin{cases} v & \text{si } x = s \\ 0 & \text{si } x \neq s, t \\ -v & \text{si } x = t \end{cases}$$

$$3.22 \quad 0 \leq f(x, y) \leq q(x, y) \quad \text{para todo } (x, y) \in A$$

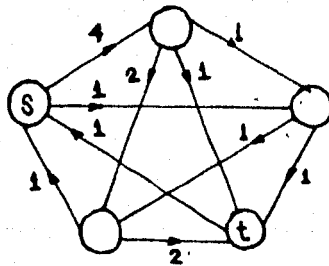
La variable  $v$  que aparece en 3.21 es un número no negativo que va a representar el valor del flujo, i.e la cantidad del flujo que se va a distribuir dentro de la red.

Se formará una ecuación 3.21 para cada nodo de la red, la cual representará la ecuación de conservación del flujo para todo nodo intermedio de la red ( $\{x \in X, x \neq s, x \neq t\}$ ).

En otras palabras, la ecuación 3.21 para cualquier nodo diferente de  $s$  (entrada) y  $t$  (salida) muestra que el flujo se conserva.

La desigualdad 3.22 significa que la cantidad de flujo  $f(x,y)$  enviado a través del arco  $(x,y)$  va a estar acotada superiormente por la capacidad del arco  $(x,y)$ , y no puede ser negativa.

En la siguiente figura se da un ejemplo de un flujo factible de  $s$  a  $t$ . Se considera que las capacidades son lo suficientemente grandes para no ser rebasadas por el flujo indicado en cada arco. El valor del flujo es  $v = 3$



Ejemplo.- La figura muestra una red con un flujo factible de valor  $v = 3$  (el número sobre cada arco indica el flujo).

En particular tenemos que un flujo de valor  $v = 0$ , i.e.

$f(x,y) = 0 \quad \forall (x,y) \in A$  es un flujo factible ya que satisface

3.1 y 3.2, es decir:

$$\sum_y f(x,y) - \sum_y f(y,x) = 0 \quad \text{si } x = s$$

$$\sum_y f(x,y) - \sum_y f(y,x) = 0 \quad \text{si } x \neq s, t$$

$$\sum_y f(x,y) - \sum_y f(y,x) = 0 \quad x = t$$

y además como  $f(x,y) = 0 \quad \forall (x,y) \in A$  cumple con  $0 \leq f(x,y) \leq q(x,y)$

• un flujo de valor  $v = 0$  es factible.

NOTA.- Hacemos la aclaración que debido a que denotamos por :

$n$  = número de nodos en una red

$m$  = número de arcos en una red

tendremos que dada una red  $G = (X, A, d)$ , la cardinalidad del conjunto  $X$  es  $n$ , y la cardinalidad del conjunto  $A$  es  $m$ .

Así al referirnos más adelante a un flujo  $\xi$  en una red  $G = (X, A, d)$  tendremos que  $\xi = (f_1, f_2, \dots, f_m)$  i.e.  $\xi$  es un vector de flujo, el cual representa a los flujos que recorren cada uno de los arcos de la red  $G$ .

## IV.- FLUJO MAXIMO

## 4.1 Introducción.

En este capítulo se discutirá el problema básico de flujo máximo en una red dada.

Se propone un método para resolver este problema, basado en un algoritmo desarrollado por Ford-Fulkerson utilizando la llamada "técnica de marcaje".

El algoritmo de Ford-Fulkerson se resume justamente en el teorema 1 (que se establecerá más adelante).

Inicialmente se dan a conocer los conceptos necesarios para establecer dicho teorema.

## 4.2 CORTES

Definición.- Un corte en una red  $G = (X, A, d)$  dada, separando a (s) de (t) es un conjunto de arcos denotado por  $(Y; \bar{Y})$  tal que:

$$i) Y \subset X, \quad s \in Y; \quad t \in \bar{Y} = X - Y$$

$$ii) (Y; \bar{Y}) = \{(x, y) \in A \mid x \in Y, y \in \bar{Y}\} \quad (\text{arcos que salen del conjunto } Y)$$

analogamente

$$(\bar{Y}; Y) = \{(x, y) \in A \mid x \in \bar{Y}, y \in Y\} \quad (\text{arcos que llegan al conjunto } Y)$$

Si  $(Y; \bar{Y})$  es un corte y  $\xi$  es un flujo en  $G$ , entonces

$$\xi(Y; \bar{Y}) = \sum_{(x,y) \in (Y; \bar{Y})} f(x,y) \quad (\text{flujo que sale del corte})$$

$$\xi(\bar{Y}; Y) = \sum_{(x,y) \in (\bar{Y}; Y)} f(x,y) \quad (\text{flujo que entra al corte})$$

A continuación se da un ejemplo con el fin de ilustrar el concepto de un corte.

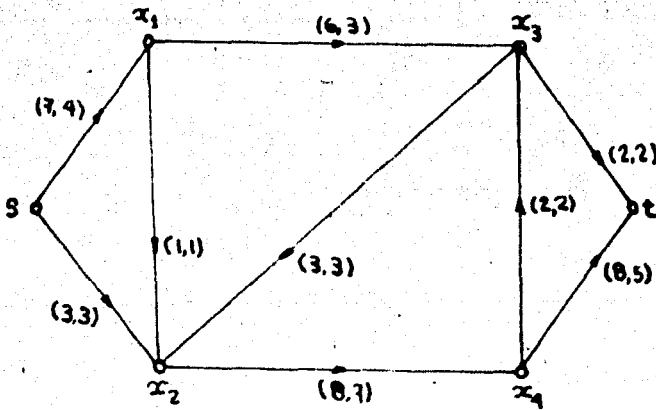


FIG 4.1

En la fig 4.1 el primer valor asociado a cada arco representa la capacidad del arco, y el segundo el flujo.

Sean los conjuntos:

$$Y = \{s, x_2\} \quad \bar{Y} = X - Y = \{x_1, x_3, x_4, t\}$$

entonces el corte  $(Y; \bar{Y})$  será el conjunto de arcos

$$\{Y; \bar{Y}\} = \{(s, x_1), (x_2, x_4)\}$$

El flujo para el corte  $(Y; \bar{Y})$  y el conjunto  $(\bar{Y}; Y)$  es el sig:

$$\begin{cases} (Y; \bar{Y}) = 4+7 = 11 & \text{Flujo que sale del corte } (Y; \bar{Y}) \end{cases}$$

$$\begin{cases} (\bar{Y}; Y) = 1+3 = 4 & \text{Flujo que entra al conjunto } (\bar{Y}; Y) \end{cases}$$

CAPACIDAD DE UN CORTE.- La capacidad de un corte  $(Y; \bar{Y})$  se define como:

$$C(Y; \bar{Y}) = \sum_{(x,y) \in (Y; \bar{Y})} q(x,y) \quad \text{donde } q(x,y) = \text{capacidad del arco } (x,y)$$

Ejemplo.- En la fig 1.4  $C(Y; \bar{Y}) = 7+8 = 15$

El corte de capacidad mínima es aquel corte que respecto a todos los cortes que se determinan en una red posee la capacidad mínima.

Al referirnos a dicho corte lo haremos con el nombre de 'corte mínimo'.

El corte mínimo intuitivamente es aquel que determina el cuello de botella en una red, i.e. detecta aquellos arcos - que van a estar saturados y que por lo tanto limitan la cantidad de flujo que puede pasar a través de la red.

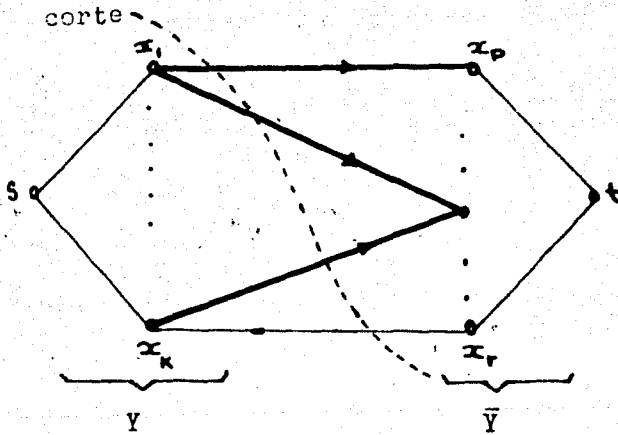
TEOREMA 1.- (Teorema Máximo flujo-Mínimo corte).

El valor máximo de un flujo dirigido de  $s$  a  $t$  en una red dada es igual a la capacidad del mínimo corte que separa a  $s$  de  $t$ .

Antes de presentar la demostración del teorema 1, consideraremos el siguiente lema.

LEMA 1.- Dada una red  $G = (X, A, d)$  y  $(Y; \bar{Y})$  un corte de esa red, todos los caminos de  $s$  a  $t$  usan por lo menos uno de los arcos del corte.

Justificación .- Sea  $(Y; \bar{Y})$  un corte cualquiera, entonces dado que  $s \in Y$ ,  $t \in \bar{Y}$  es obvio que cualquier camino que sigamos partiendo de  $s$  para llegar a  $t$  tendra que atravesar algún arco del conjunto  $(Y; \bar{Y})$ .



Cualquier camino de  $s$  a  $t$  atraviesa un arco del corte  $(Y; \bar{Y})$ .

En particular el lema 1 es valido si  $(Y; \bar{Y})$  es el corte mínimo.

DEMOSTRACION DEL TEOREMA 1.- Aparentemente es obvio que el valor máximo del flujo dirigido de  $s$  a  $t$  no puede ser mayor que la capacidad del mínimo corte  $= C(Y_m; \bar{Y}_m)$ , dado que todos los caminos de  $s$  a  $t$  usan uno de los arcos del corte.

La demostración consiste en probar que existe un flujo  $y$  que su valor es  $= C(Y_m; \bar{Y}_m)$ .

Sea  $\xi = (f_1, f_2, \dots, f_m)$  un vector de flujo que representa al flujo que recorre cada uno de los arcos de la red.

Sea  $(Y_0; \bar{Y}_0)$  un corte definido en la red; el cual va a estar formado por la aplicación recursiva del paso b (definido abajo).

PASO (a) Inicialmente comenzamos con  $Y_0 = \{s\}$

PASO (b) Si  $x \in Y_0$ , y si se cumplen cualquiera de las dos condiciones siguientes:  $f(x, y) < q(x, y)$  ó  $f(y, x) > 0$ , el vértice ( $y$ ) pasará a formar parte del conjunto  $Y_0$ ; este proceso se repetirá hasta que  $Y_0$  no pueda ser incrementado con un nuevo vértice, ó bien que  $t \in Y_0$ .

De aquí que solo podemos llegar a dos situaciones:

1)  $t \in Y_0$

2)  $t \notin Y_0$

Analizando ambos casos tenemos lo siguiente:

Caso (1)  $t \in Y_0$ . Esto quiere decir que existe una cadena elemental del vértice  $s$  al vértice  $t$ ; sucede por lo tanto que todo arco (de la cadena) de la forma  $(x_i, y_j)$  es recorrido en la misma dirección que el sentido que tiene dentro de la red, y



para el cual se cumple la condición siguiente  $f(x_i, y_j) < q(x_i, y_j)$ .

Ahora todo arco (de la cadena) de la forma  $(y_j, x_i)$  será recorrido en dirección opuesta al sentido que tiene el arco dentro de la red y para el cual se cumple la condición siguiente  $f(y_j, x_i) > 0$ .

NOTA.- Esta cadena de arcos de  $s$  a  $t$  se denomina CADENA AUMENTACIONES DE FLUJOS.

Sea

$$(1.1) \quad \Psi_f = \min_{(x_i, y_j)} [q(x_i, y_j) - f(x_i, y_j)]$$

(para arcos recorridos en el mismo sentido)

$$(1.2) \quad \Psi_b = \min_{(x_i, y_j)} [f(x_i, y_j)]$$

(para arcos recorridos en sentido inverso)

$$(1.3) \quad \Psi = \min [\Psi_f, \Psi_b]$$

Si  $\Psi$  es agregado al flujo de todos los arcos recorridos en el sentido  $(x_i, y_j)$ , y disminuido en los arcos recorridos en sentido contrario  $(y_j, x_i)$ , la resultante es un nuevo flujo factible  $f'$  con  $\Psi$  unidades mayor que el flujo anterior. El aumento  $\Psi$  de flujo en los arcos recorridos en el sentido  $(x_i, y_j)$  no puede violar la restricción de capacidad asociada a cada arco ya que  $\Psi \leq \Psi_f$  y la disminución de flujo a los arcos

Sabemos por otro lado que la capacidad del corte  $(Y; \bar{Y}_0)$  es

$$C(Y_0; \bar{Y}_0) = \sum_{(x_i, y_j) \in (Y_0; \bar{Y}_0)} q(x_i, y_j) \quad \text{----- (v)}$$

Por lo que de (iv) e (v) concluimos que el valor del flujo es

$$v = C(Y_0; \bar{Y}_0)$$

donde  $(Y; \bar{Y}_0)$  es el corte mínimo, ya que de no ser así hubiéramos podido seguir encontrando vértices que ingresaran al conjunto  $Y_0$  hasta que se incluyera al vértice  $t$ ; y así incrementar el flujo anterior  $\Psi$  unidades y una vez más volver a definir un nuevo conjunto  $Y_0$ .

Ya que en el caso (1) el valor del flujo es continuamente incrementado por lo menos en una unidad, si suponemos que las capacidades  $q(x_i, y_j)$  son enteros, entonces el flujo máximo debe ser encontrado en un número finito de pasos, cuando el caso (2) ocurra. Entonces el valor del flujo encontrado será igual al valor de la capacidad del corte  $(Y_0; \bar{Y}_0)$  el cual es por lo tanto el corte mínimo. L. Q. D.

recorridos en sentido inverso no puede formar un flujo negativo ya que  $\varphi \leq \varphi_0$ .

Así usando el nuevo flujo obtenido volvemos a aplicar los pasos (a) y (b) definiendo un nuevo conjunto  $Y_0$  y repetimos el proceso.

Caso (2)  $t \notin Y_0$ . Lo anterior implica que  $t \in \bar{Y}_0$ .

De acuerdo con el paso (b) tendremos lo siguiente:

- i)  $f(x_i, y_j) = q(x_i, y_j)$  para todos los arcos  $(x_i, y_j) \in (Y_0; \bar{Y}_0)$
- ii)  $f(x_i, y_j) = 0$  para todo arco  $(x_i, y_j) \in (\bar{Y}_0; Y_0)$

Lo anterior se cumple, ya que de no ser así podríamos seguir encontrando vértices que ingresaran al conjunto  $Y_0$  hasta que se incluyera al vértice  $t$  (caso 1).

Ahora por la ecuación 3.21 sabemos que el valor de un flujo es :

$$v = \sum_{(x_i, y_j) \in (Y_0; \bar{Y}_0)} f(x_i, y_j) - \sum_{(x_i, y_j) \in (\bar{Y}_0; Y_0)} f(x_i, y_j)$$

sustituyendo (i), (ii) obtenemos

$$v = \sum_{(x_i, y_j) \in (Y_0; \bar{Y}_0)} q(x_i, y_j) - 0 \quad \text{----- (iv)}$$

Lema 2.- El vector de flujo  $\xi'$  definido en el caso (1) es un flujo factible sobre una red dada.

Demostración.-  $\xi'$  es un flujo ya que como partimos de un vector de flujo  $\xi = (f_1, f_2, \dots, f_m)$ .

$$\xi' = \xi + \psi \quad \therefore \xi' \text{ sigue siendo un vector de flujo.}$$

$\xi'$  es un flujo factible ya que por construcción se tiene que el aumento de flujo  $\psi$  no viola las restricciones de capacidad; es decir :

$$\psi \leq q(x,y) - f(x,y)$$

$$\psi \leq f(x,y)$$

Por lo tanto cada flujo  $\xi'$  definido en el caso (1) es factible I.Q.Q.D.

4.3.- ALGORITMO DE MARCAJE.- Para encontrar el flujo máximo (FORD-FULKERSON) en una red dada.

Introducción.

El método constructivo usado en la demostración del teorema 1 propone un algoritmo para encontrar el flujo máximo de un vértice  $s$  a un vértice  $t$  dados en una red con límites de capacidad en sus arcos.

Este algoritmo se inicia con un flujo factible arbitrario (se puede usar un valor de flujo igual a cero, ya que como se demostró en la sección 3.2 es un flujo factible).

A continuación se da la formalización del algoritmo.

PASO 0: Sea  $f$  un flujo factible, marcamos el vértice  $s$  con la marca  $\{t, \infty\}$  (todos los otros vértices no deberán estar marcados).

PASO 1: PROCESO DE MARCAJE.

Sea  $Y$  el conjunto de nodos que han sido marcados con la etiqueta  $\{z, \varphi\}$  (con  $z =$  número de un nodo,  $\varphi =$  número  $> 0$ ) entonces inicialmente  $Y = \{s\}$   
 Se podrá marcar un vértice  $y \notin Y$  en alguno de los siguientes casos :

a) MARCAJE DIRECTO .- En el caso que exista un arco  $(x,y)$  con  $x \in Y$  y además  $f(x,y) < q(x,y)$ , si se cumple esto entonces vamos a marcar el vértice  $(y)$  con la etiqueta  $\{x, \text{Min}(\varphi, q(x,y) - f(x,y)) > 0\}$  y marcamos el arco  $(x,y)$  con el signo  $(+)$ .

b) MARCAJE INVERSO .- En el caso que exista un arco  $(y,x)$  con  $x \in Y \exists f(y,x) > 0$ , marcaremos al vértice  $(y)$  con la etiqueta  $\{x, \text{Min}(\varphi, f(x,y)) > 0\}$  y marcamos el arco  $(y,x)$  con el signo  $(-)$ .

(1.1) Si ya no es posible marcar ningún vértice  $(y)$  no ha sido marcado terminar.

El flujo obtenido es máximo.

(1.2) Si  $(t)$  ha sido marcado ir al paso 2.

(1.3) Vamos a incluir en  $Y$  al vértice  $(y)$  marcado. Si el vértice  $(y)$  es igual a  $(t)$  ir al caso (1.2), si no ir al paso 1.

PASO 2: PROCESO DE CAMBIO DE FLUJO.

Cuando un vértice  $(y)$  haya recibido la etiqueta  $\{z, \varphi\}$  diremos que  $z$  es el predecesor de  $(y)$ . Una vez que  $(t)$  ha sido marcado; podemos regresar a partir de  $(t)$ , de predecesor en predecesor hasta el vértice  $(s)$ ; este procedimiento muestra la existencia de una cadena elemental de  $s$  a  $t$  (llamada cadena aumento-flujo), si a esta cadena le agregamos el arco  $(t,s)$  obtendremos un ciclo  $\mathcal{M}$ ; tomamos como sentido de recorrido a  $\mathcal{M}$  el sentido del

arco  $(t,s)$  y notemos que se cumple lo siguiente:

$M^+$  = arcos del ciclo  $M$  marcados con signo (+)

$M^-$  = arcos del ciclo  $M$  marcados con signo (-)

Sea  $\{Z, \varphi_t\}$  la etiqueta del vértice  $t$  y pongamos

$$\xi'(x,y) = \begin{cases} \xi(x,y) + \varphi_t & \text{si } (x,y) \in M^+ \\ \xi(x,y) - \varphi_t & \text{si } (x,y) \in M^- \\ \xi(x,y) & \text{si } (x,y) \notin M \end{cases}$$

Borrar todas las marcas de los vértices excepto la de  $(s)$ ,reemplazamos el flujo  $\xi$  por  $\xi'$  e ir al paso 1.

Ejemplo.- Obtención de un flujo máximo por medio del algoritmo de Ford-Fulkerson en la red de la fig 4.2

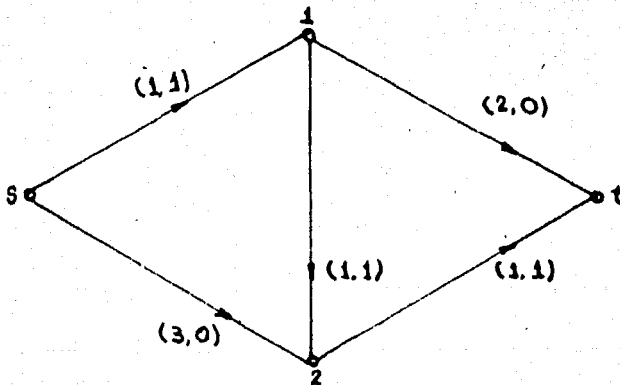


FIG 4.2.- El primer número asociado a cada arco representa su capacidad, y el segundo el flujo.

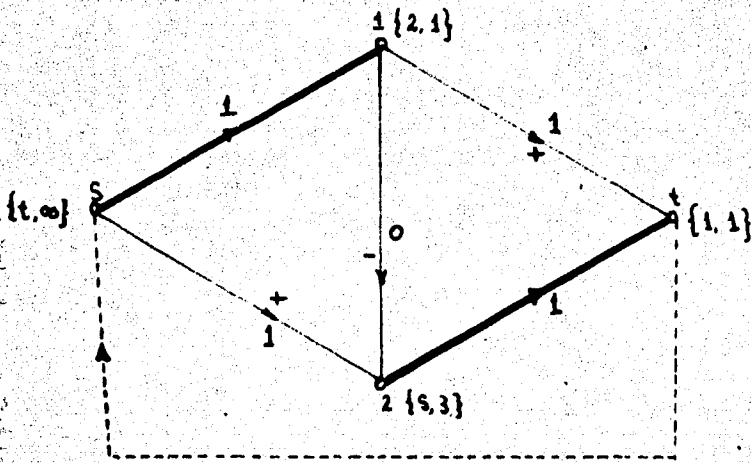
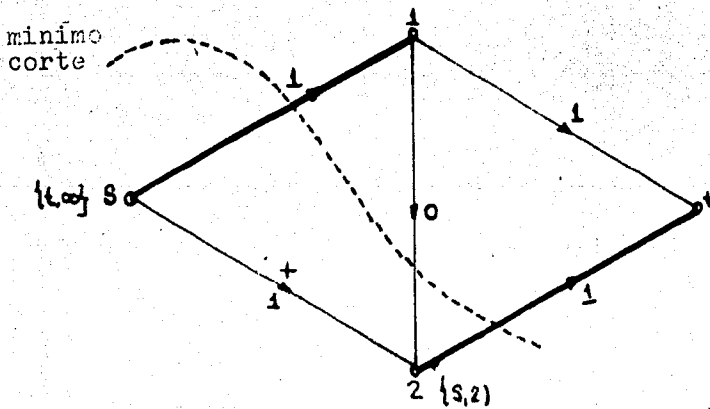


FIG 4.3 (a)



— arcos saturados  
i.e.  $q(x,y) = f(x,y)$

FIG 4.3 (b)

En la primera aplicación del proceso de marcaje para la obtención del flujo máximo sobre la red de la fig 4.2 llegamos a etiquetar el vértice (t) con la etiqueta  $\{1,1\}$  por



lo cual tenemos que  $\Psi_t = 1$ ;  $\mathcal{M} = \{(s,2), (1,2), (1,t)\}$   
 $\mathcal{M}^+ = \{(s,2), (1,t), (t,s)\}$   $\mathcal{M}^- = \{(1,2)\}$ ; el flujo final es  
 mostrado en la fig 4.3 (a).

En la segunda iteración, fig 4.3(b) los únicos vértices  
 posibles a marcar son (s) y (2) y el algoritmo no puede con-  
 tinuar.

Entonces nuestro corte de capacidad mínima es el siguien-  
 te:

$$(Y; \bar{Y}) = \{(s,1), (2,t)\} \text{ con } Y = \{s,2\} ; \bar{Y} = \{1,t\}$$

$$\begin{cases} (Y; \bar{Y}) = f(s,1) + f(2,t) = 2 & \text{(flujo que sale de } (Y; \bar{Y})) \\ (\bar{Y}; Y) = 0 & \text{(flujo que entra a } (Y; \bar{Y})) \end{cases}$$

tenemos que la capacidad del corte es:

$$C(Y; \bar{Y}) = q(s,1) + q(2,t) = 1 + 1 = 2$$

el valor del flujo en el corte  $(Y; \bar{Y})$  es :

$$v(\bar{Y}) = \sum_{(x_i, y_j) \in (Y; \bar{Y})} f(x_i, y_j) - \sum_{(x_i, y_j) \in (\bar{Y}; Y)} f(x_i, y_j) =$$

$$(f(s,1) + f(2,t)) - f(1,2) = 2 - 0 = 2$$

••  $C(Y; \bar{Y}) = 2 = v$  i.e. el valor máximo del flujo = 2

## V.- BASES QUE INTERVIENEN EN EL PLANTEAMIENTO DEL PROBLEMA DE FLUJO A COSTO MÍNIMO

### 5.1 Introducción.

En los capítulos anteriores se ha dado ya un procedimiento para resolver el problema de flujo máximo en una red. Esto nos ha servido, simplemente para conocer la cantidad de flujo máximo que podemos hacer pasar por la red así como la manera de distribuir dicho flujo a través de los arcos en función de su capacidad, sin embargo, dicha forma de distribuir el flujo no necesariamente es la mejor en cuanto a costo de utilización de los arcos.

Este es el problema que se tratará a continuación. En particular tenemos que en este capítulo se muestran las transformaciones que se pueden realizar en una red; conociendo el flujo que se transporta en ella, así como los cambios que se pueden realizar en la manera de redistribuir dicho flujo. Se pretende dar información acerca de las bases que se utilizan como herramienta para encontrar una solución al problema de flujo a costo mínimo.

El objetivo en esta sección es pues, que exista una mayor comprensión del procedimiento que se utiliza para resolver el problema de flujo a costo mínimo.

## 5.2 CONSTRUCCION DE UNA RED INCREMENTAL

### Introducción.

En esta sección se muestra que dada una red, y un flujo que la atraviesa de  $s$  a  $t$ , la podemos transformar sin que por esto se altere su estructura original. Así en la nueva red generada podremos aplicar los procedimientos que se presentan más adelante para minimizar el costo de dicho flujo.

### DESCRIPCION DEL PROCESO

Sea  $G = (X, A, d)$  una red y  $f$  un vector de flujo en  $G$ , entonces denotaremos a la red incremental por  $G^u(f) = (X^u, A^u, d^u)$  definida de la siguiente manera:

$$X^u = X \qquad A^u = A_1^u \cup A_2^u$$

A cada arco  $(x, y) \in A$  lo vamos a dividir en 2 arcos de la siguiente manera:

i) Un arco de la forma  $(x^u, y^u)$  con capacidad:

$$q^u(x^u, y^u) = q(x, y) - f(x, y)$$

ii) Un arco de la forma  $(y^u, x^u)$  (sentido inverso a  $(x^u, y^u)$ ) y con capacidad:

$$q^u(y^u, x^u) = f(x, y)$$

NOTA.- En el caso en que algún arco de la red incremental definido por (i) ó (ii) tenga asociada una capacidad

nula, i.e. capacidad = cero, dicho arco se omite en  $G^u(\xi)$ .

El conjunto de arcos  $A^u$  estará formado por dos conjuntos;  $A_1^u$  y  $A_2^u$ ,  $A^u = A_1^u \cup A_2^u$  en donde  $A_1^u$  es el conjunto que contiene los arcos de la clase (i), y  $A_2^u$  es el conjunto que contiene los arcos de la clase (ii).

Entonces la red incremental  $G^u(\xi)$  es una red generada a partir de la red original  $G$  y un flujo  $\xi$  en  $G$ .

Observación.- La transformación de una red a una red incremental dado un vector de flujo  $\xi$ , no causa ninguna alteración en las soluciones que se obtengan a los problemas planteados, si en lugar de trabajar sobre la red  $G$  (red original) trabajamos sobre la estructura de la red  $G^u(\xi)$  (red incremental). Esto es debido a que en realidad es la misma estructura sobre la cual se trabaja.

Así el vector de flujo máximo  $\xi$  encontrado en la red  $G$  por medio del algoritmo de marcaje (sección 4.3) es el vector de flujo máximo dentro de la red  $G^u(\xi)$  también.

La razón es la siguiente:

El proceso de encontrar una cadena aumenta-flujo en una red  $G = (X, A, d)$  con  $\xi =$  vector de flujos, se puede considerar como la búsqueda de un camino de  $s$  a  $t$  en la red incremental, i.e. el algoritmo de Marcaje es la búsqueda del mínimo corte en la red  $G^u(\xi)$ ; si llegamos a marcar a  $t$

entonces habremos encontrado un camino  $P$  de  $(s)$  a  $(t)$  en  $G^u(\xi)$ .

La cadena aumenta-flujo encontrada en la red  $G$  es un camino  $P$  definido en la red  $G^u(\xi)$ , en donde los arcos de  $P$  que estan en el conjunto  $A_1^u$  son los arcos de sentido  $(x^u, y^u)$ , y los arcos de  $P$  que estan en el conjunto  $A_2^u$  son los arcos de sentido inverso,  $(y^u, x^u)$ .

La explicación anterior nos muestra pues, que los resultados obtenidos sobre una red  $G = (X, A, d)$  son validos para su correspondiente red incremental  $G^u(\xi)$ .

A continuación se muestra un ejemplo en el que a partir de una red  $G$  dada, y un vector de flujo  $\xi$  se genera la red incremental  $G^u(\xi)$ .

En la fig 4.4 se muestra la red  $G$  y en la fig 4.4 (b) se muestra la red incremental  $G^u(\xi)$ .

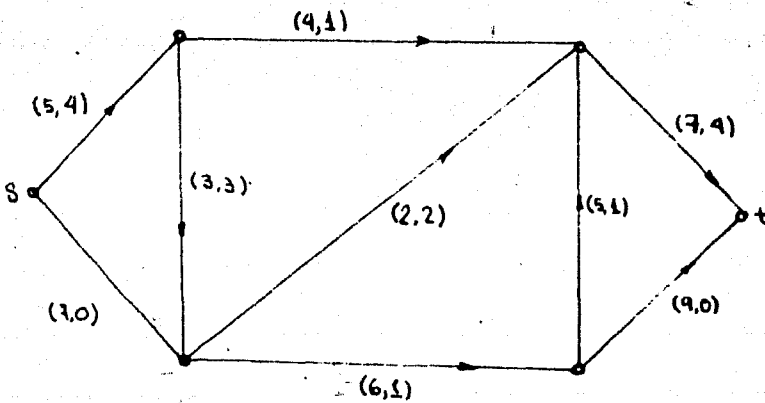


FIG 4.4 .- Red  $G = (X, A, d)$

La primer etiqueta asociada a cada arco representa su capacidad y, la segunda el flujo.

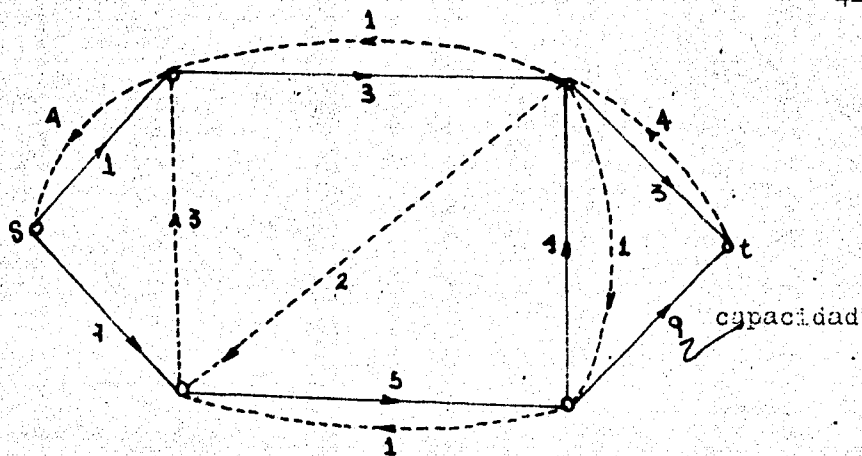


FIG 4.4 (b).- red incremental  $G^u(\{ \})$

### 5.3 DESCOMPOSICION DE UN PATRON DE FLUJO

Llamaremos patrón de flujo a la manera de distribuir un flujo a través de una red.

En otras palabras un patrón de flujo es el vector de flujo que representa a todos los flujos que recorren cada uno de los arcos de una red.

Por ejemplo; en la fig 4.2 el patrón de flujo es el vector  $\xi = (f(s,1), f(s,2), f(1,2), f(1,t), f(2,t))$  con:

$$f(s,1) = 1$$

$$f(s,2) = 0$$

$$f(1,2) = 1$$

$$f(1,t) = 0$$

Así el valor del flujo es  $v = \dots$ .

Una descomposición de un patrón de flujo es una nueva distribución de éste, diferente a la que tenía originalmente, sin que cambie por esto su valor ( $v$ ).

#### DESCRIPCIÓN DEL PROCESO

Denotaremos por  $h_0(S)$  un patrón de flujo en una red  $G$ , donde  $S$  representa al conjunto de arcos de la red  $G$  en los cuales se transportan unidades de flujo.

Así tendremos que para los arcos  $(x_i, y_j) \in S$  tienen un flujo  $f(x_i, y_j) = h$  (el valor  $h$  no es necesariamente igual en todos los arcos), y los arcos  $(x_i, y_j) \notin S$  tienen un flujo  $f(x_i, y_j) = 0$ .

$h_0(S)$  debe satisfacer las condiciones 3.21 y 3.22 ; esto es, que el flujo sea factible con este patrón de flujo en la red  $G$ .

Para que  $h_0(S)$  represente un patrón de flujo con éstas características el conjunto de arcos ( $S$ ) deberá formar un camino en  $G$  de  $s$  a  $t$ , ó bien un circuito de  $G$ .

Ahora bien, dados dos patrones de flujo  $\xi$  y  $\Upsilon$ , denotaremos  $\xi + \Upsilon$  como el patrón de flujo para el cual el total de flujo transportado en el arco  $(x_i, y_j)$  es  $\xi_{ij} + \Upsilon_{ij}$ .

$$f(2,t) = 1$$

en este caso el valor del flujo es  $v = 1$ .

Una descomposición de un patrón de flujo es una nueva distribución de éste, diferente a la que tenía originalmente, sin que cambie por esto su valor ( $v$ ).

#### DESCRIPCION DEL PROCESO

Denotaremos por  $h_0(S)$  un patrón de flujo en una red  $G$ , donde  $S$  representa al conjunto de arcos de la red  $G$  en los cuales se transportan unidades de flujo.

Así tendremos que para los arcos  $(x_i, y_j) \in S$  tienen un flujo  $f(x_i, y_j) = h$  (el valor  $h$  no es necesariamente igual en todos los arcos), y los arcos  $(x_i, y_j) \notin S$  tienen un flujo  $f(x_i, y_j) = 0$ .

$h_0(S)$  debe satisfacer las condiciones 3.21 y 3.22 ; esto es, que el flujo sea factible con este patrón de flujo en la red  $G$ .

Para que  $h_0(S)$  represente un patrón de flujo con éstas características el conjunto de arcos ( $S$ ) deberá formar un camino en  $G$  de  $s$  a  $t$ , ó bien un circuito de  $G$ .

Ahora bien, dados dos patrones de flujo  $\xi$  y  $\Upsilon$ , denotaremos  $\xi + \Upsilon$  como el patrón de flujo para el cual el total de flujo transportado en el arco  $(x_i, y_j)$  es  $\xi_{ij} + \Upsilon_{ij}$ .



TEOREMA 2.- Si  $\xi$  es cualquier patrón de flujo que recorre una red  $G$  dada, de  $s$  a  $t$ , con valor (entero)  $v$ , entonces; se puede descomponer como:

$$\xi = 1 \cdot (P_1) + 1 \cdot (P_2) + \dots + 1 \cdot (P_v) + 1 \cdot (\phi_1) + 1 \cdot (\phi_2) + \dots + 1 \cdot (\phi_k)$$

donde  $P_1, P_2, \dots, P_k$  son caminos elementales de  $s$  a  $t$  en  $G$ , y  $\phi_1, \phi_2, \dots, \phi_k$  son circuitos elementales de  $G$ .

(Los  $P_i$  y  $\phi_i$  no son necesariamente distintos, i.e. la intersección  $P_i$  y  $\phi_i$  en la red  $G$  no necesariamente es vacía).

NOTA.- La demostración se realizará dentro de la estructura de una gráfica por necesidades de construcción.

Demostración.- A partir de la gráfica  $G = (X, A)$  y el patrón de flujo  $\xi$  construimos una nueva gráfica llamada gráfica unitaria, denotada por  $G^e = (X^e, A^e)$  de la siguiente manera:

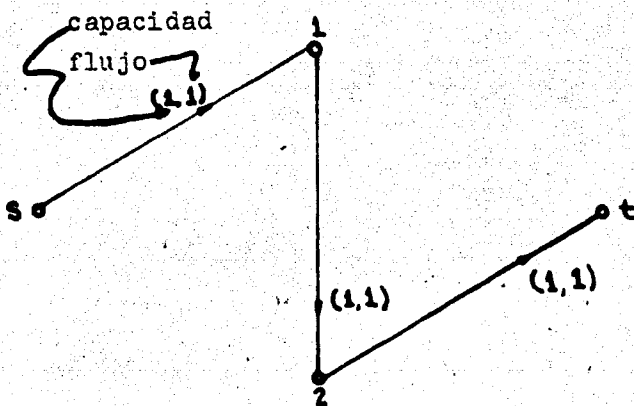
El conjunto de vértices  $X^e$  es igual al conjunto de vértices  $X$ , si  $\xi_{ij}$  es el flujo en el arco  $(x_i, y_j)$  de  $G$ , entonces lo representamos en  $\xi_{ij}$  arcos paralelos entre los correspondientes vértices  $x_i^e$  y  $y_j^e$  de  $G^e$ . La gráfica  $G^e$  es entonces una  $s$ -gráfica ya que cada arco en  $G^e$  corresponde a una unidad de flujo en la gráfica  $G$ .

Si  $\xi_{ij} = 0$  para  $(x_i, y_j)$ , no se incluirá ningún arco entre  $x_i^e$  y  $y_j^e$ .

La gráfica  $G^e = (X^e, A^e)$  representa al flujo en la gráfica  $G = (X, A)$ .

Antes de continuar con la demostración del teorema 2 daremos un ejemplo que ilustre el procedimiento de construcción de una red  $G^e = (X^e, A^e, d^e)$  que representa el flujo en una red dada  $G = (X, A, d)$ .

Para el patrón de flujo de la red  $G = (X, A, d)$  mostrado en la fig 4.2 damos a continuación el esquema de la red  $G^e = (X^e, A^e, d^e)$



$$X^e = \{s, 1, 2, t\}$$

$$A^e = \{(s, 1), (1, 2), (2, t)\}$$

$$d^e : q(s, 1) = 1$$

$$q(1, 2) = 1$$

$$q(2, t) = 1$$

$$\text{RED } G^e = (X^e, A^e, d^e)$$

(continuación de la demostración del teorema 2)

Por las condiciones 3.21 y 3.22 que cumple un flujo factible, y dado que cada arco de la red  $G^e$  corresponde a una unidad de flujo transportado en  $G$ , se cumple lo siguiente:

- i) El número de arcos cuyo vértice inicial es  $x_i^e$  es igual al número de arcos cuyo vértice final es  $x_i^e$ ; lo anterior se cumple para todo  $x_i^e \in X^e$ ,  $x_i^e \neq s^e$  ó  $t^e$ .

En otras palabras lo anterior significa que el número de arcos que llegan a cualquier nodo intermedio es igual

al número de arcos que salen de cualquier nodo intermedio.

ii) El número de arcos cuyo vértice inicial es  $s^e$  es igual al número de arcos que llegan a la salida  $t^e$ , el cual es igual al valor del flujo  $= v^e$ .

Si agregamos a la red  $G^e$   $v$  arcos dirigidos del vértice  $t^e$  al vértice  $s^e$ , la red  $G^e$  poseera un circuito Euleriano (por las propiedades de un circuito Euleriano ya que el grado de los vértices es par siempre). Ahora, si desconectamos estos  $v$  arcos del circuito Euleriano lo que obtendremos serán  $v$  caminos de  $s$  a  $t$  los cuales en su totalidad atraviesan cada arco de  $G^e$  exactamente una vez.

Si denotamos a estos caminos como  $P'_1, P'_2, \dots, P'_v$ , por la definición de circuito Euleriano (sección 1.2) los caminos  $P'_i$  no son necesariamente elementales (i.e. pueden pasar varias veces por el mismo vértice), y como cualquier camino no elemental puede ser considerado como la suma de un camino elemental (de  $s^e$  a  $t^e$ ) y un número de circuitos elementales (con arcos diferentes); entonces obtenemos lo siguiente:

$$\xi = 1_0(P'_1) + 1_0(P'_2) + \dots + 1_0(P'_v) + 1_0(\phi_1) + \dots + 1_0(\phi_k)$$

donde  $P'_i$  son los caminos elementales de  $s^e$  a  $t^e$  y  $\phi_i$  son los circuitos elementales. L.Q.Q.D.

## VI.- PROBLEMA DE FLUJO A COSTO MINIMO

### 6.1 Introducción

En los capítulos anteriores se ha dado un procedimiento para la obtención de un flujo máximo de  $s$  a  $t$  en una red dada, sin hacer referencia alguna a los costos.

Ahora trataremos de resolver el problema de la obtención de un flujo máximo de  $s$  a  $t$  cuyo costo de transportación sea el más económico. Para esto, trataremos de minimizar la variable costo.

En este problema, cada arco  $(x,y)$  tiene dos números asociados: uno que es la capacidad  $q(x,y)$ , y otro que es el costo por unidad de flujo transportado  $c(x,y)$ .

El algoritmo utilizado para minimizar el costo de un flujo transportado a través de una red de  $s$  a  $t$ , es un algoritmo basado en la determinación de circuitos de costo negativos.

Un circuito de costo negativo representará para nosotros la posibilidad de poder disminuir el costo de transportación del flujo al hacer una redistribución del flujo en los arcos que forman el circuito negativo, sin que por esto el valor del flujo se altere.

## 6.2 ALGORITMO DE FLOYD

Este algoritmo es utilizado para detectar los circuitos negativos en una red dada.

Se aplica a una matriz de costos  $C$ , donde  $c_{ij} = \text{costo}(x_i, y_j)$   $\forall (x_i, y_j)$  que este en la red.

El algoritmo de Floyd considera cada entrada de la matriz  $C$  como distancias, i.e. el costo de cada arco  $(x_i, y_j)$  se considera como la distancia de  $(x_i)$  a  $(y_j)$ .

Así mediante la matriz  $C$  se trata de buscar los caminos más cortos entre todo par de vértices; suponiendo que la matriz  $C$  no satisface las condiciones de triangularidad, i.e.

$$c_{ij} < c_{ik} + c_{kj} \quad \forall i, j, k$$

## DESCRIPCION DEL ALGORITMO

La matriz de costo  $C$  se inicializa de la siguiente manera:

$c_{ii} = 0$  para toda  $i = 1, 2, \dots, n$ ; y  $c_{ij} = \infty$  para todo arco  $(x_i, y_j) \notin G$ .

PASO 1.-  $k = 0$

iteración

PASO 2.-  $k = k - 1$

PASO 3.- Para toda  $i \neq k \Rightarrow c_{ik} \neq \infty$ , y toda  $j \neq k \Rightarrow c_{kj} \neq \infty$  se realiza la siguiente operación:

$$c_{ij} = \min [c_{ij}, (c_{ik} + c_{kj})] \quad 6.11$$

(fin del algoritmo)

- PASO 4.- (a) si para algún  $c_{ii} < 0$ , entonces habremos encontrado un circuito de costo negativo, el cual se inicia en el vértice  $x_i$  y termina en  $x_i$ .
- (b) si todo  $c_{ii} \geq 0$  y  $k = n$  ( $n =$  número de vértices de  $G$ ), la matriz  $C$  contendrá entonces las longitudes de todos los caminos mínimos, y no es posible encontrar nuevos ciclos negativos. El algoritmo termina.
- (c) si  $c_{ii} \geq 0$  pero  $k < n$  ir al paso 2 y continuar.

### 6.3 MATRIZ $\Theta$ DE CAMINOS-MINIMOS.

Esta matriz desempeña un papel importante dentro del proceso de búsqueda de los circuitos negativos de una red; a través de ella nosotros podremos localizar que arcos están incluidos en cada circuito negativo que se encuentre.

Mediante cada una de sus entradas, correspondientes a cada nodo, sabremos qué arcos llegan a él y de qué nodo proceden; de esta manera si localizamos por medio del algoritmo de Floyd un circuito que se inicia y termina en el vértice  $x_i$ , podremos buscar entonces en la matriz  $\Theta$  todos los vértices predecesores de  $x_i$  y así iremos encontrando los arcos contenidos en ese circuito.

La matriz se denota como  $\Theta = [\theta_{ij}]_{n \times n}$ .

La matriz  $\Theta$  trabaja paralelamente con el paso 3 del algoritmo de Floyd ya que mediante la triple ecuación (6.11) va ir guardando los caminos mínimos para cada par de vértices  $x_i, y_j$ .

La matriz  $\Theta$  se inicializa como sigue:

$$\Theta_{ij} = x_i \quad \forall (x_i, y_j) \in G.$$

Esta matriz funciona dentro del algoritmo de Floyd de la siguiente manera:

$$\Theta_{ij} = \begin{cases} \Theta_{ik}, & \text{si } (c_{ik} + c_{kj}) < c_{ij} \\ \Theta_{ij}, & \text{si } c_{ij} \leq (c_{ik} + c_{kj}) \end{cases}$$

donde los índices  $i, j, k$  toman los valores según lo indica el algoritmo de Floyd (sección 6.2).

Al final del proceso (i.e. cuando ya no existen circuitos negativos en la red) la matriz  $\Theta$  contendrá los caminos mínimos entre cualquier par de vértices  $x_i$  y  $y_j$ ; el camino entre un par de vértices  $x_i, y_j$  se da por toda la secuencia de vértices intermedios que están en el camino que va de  $x_i$  a  $y_j$  ;

$$x_i, x_\nu, \dots, x_\gamma, x_\beta, x_\alpha, y_j$$

donde  $x_\alpha = \Theta_{ij}$ ,  $x_\beta = \Theta_{i\alpha}$ ,  $x_\gamma = \Theta_{i\beta}$ , etc, hasta que  $x_i = \Theta_{i\nu}$

Así de esta manera se podrá identificar los circuitos negativos.

#### 6.4 ALGORITMO BASADO EN LA DETERMINACION DE CIRCUITOS NEGATIVOS

Sea la red  $G(X, A, d)$ , y sea  $\xi$  un flujo factible de valor  $v$  en  $G$ ; (este flujo puede ser el flujo máximo si lo calculamos de acuerdo al algoritmo de Ford-Fulkerson).

Con el flujo factible  $\xi$ , definimos una red incremental  $G^u(\xi) = (X^u, A^u, d^u)$  siguiendo el procedimiento dado en la sección 5.2; el costo asociado a cada arco de la nueva red se asigna así :

Para cada arco  $(x_i^u, y_j^u) \in A_1^u$  el costo asociado es  $c^u(x_i^u, y_j^u) = c(x_i, y_j)$ , y para cada arco  $(y_j^u, x_i^u) \in A_2^u$  el costo asociado es  $c^u(y_j^u, x_i^u) = -c(x_i, y_j)$ .

Entonces la nueva red  $G^u(\xi) = (X^u, A^u, d^u)$  tendrá representadas las capacidades y los costos (incrementales) de acuerdo a la distribución inicial del flujo  $\xi$ ; ahora si llegasemos a encontrar un circuito negativo en  $G^u(\xi)$ , lo que se hace es hacer una nueva distribución del flujo  $\xi$  en  $G$  y así disminuir el costo.

El algoritmo esta basado en el siguiente teorema:

Teorema 3.-  $\xi$  es un flujo de valor  $v$  con costo mínimo si y solo si no hay circuitos  $\phi$  en  $G^u(\xi)$ , tales que la suma de los costos en los arcos de  $\phi$  sea negativa.



Demostración.- Sea  $c[\xi]$  el costo del flujo  $\xi$  en la red  $G$ , y  $c[\Phi | G^u(\xi)]$  la suma de todos los costos de los arcos en el circuito  $\Phi$  en la red  $G^u(\xi)$ .

1) Suponemos que  $\xi$  es un flujo de costo mínimo.

Si suponemos que  $c[\Phi | G^u(\xi)] < 0$  para algún circuito  $\Phi$  de  $G^u(\xi)$ , entonces la circulación de una nueva unidad de flujo adicional a través del circuito  $\Phi$  produce una nueva distribución del flujo  $= \xi + 1 \cdot (\Phi)$ ; esto sin alterar el valor  $v$  del flujo de  $s$  a  $t$ .

El costo de transportación de flujo de  $\xi + 1 \cdot (\Phi)$  es  $c[\Phi] + c[\Phi | G^u(\xi)] < c[\xi]$  lo cual contradice la hipótesis de que  $\xi$  es un flujo de costo mínimo.

2) Suponemos que  $\xi$  no es un flujo de costo mínimo.

si  $c[\Phi | G^u(\xi)] \geq 0$  para todo circuito  $\Phi$  en la red  $G^u(\xi)$  y sea  $\xi^*$  ( $= \xi$ ) es el flujo de costo mínimo de valor  $v$ .

Sea el patrón de flujo  $\xi^* - \xi$  para el cual el flujo en el arco  $(x_i, y_j)$  sea igual a  $\xi_{ij}^* - \xi_{ij}$ .

Ahora como  $\xi^*$  y  $\xi$  pueden ser descompuestos en una suma de flujos transportados a través de caminos de  $s$  a  $t$  en  $G$ , podremos entonces construir una red unitaria  $G^e$  para el patrón de flujo  $\xi^* - \xi$ .

Así tendremos que el conjunto de arcos cuyo nodo inicial es  $x_i^e$  es igual al conjunto de arcos cuyo nodo final es  $x_i^e$ , y esto es para todo  $x_i^e \in X^e \rightarrow x_i^e \neq s^e, t^e$ .

De acuerdo con la sección 5.3  $G^e$  consiste de una colección de circuitos unitarios de flujo  $\phi_1, \phi_2, \dots, \phi_k$ , entonces

$$\xi^a - \xi = 1 \circ (\phi_1) + 1 \circ (\phi_2) + \dots + 1 \circ (\phi_k)$$

Sabemos por otro lado que

$\xi^u = \xi + 1 \circ (\phi_1) + 1 \circ (\phi_2) + \dots + 1 \circ (\phi_k)$  es un flujo factible ya que cumple con las condiciones de un flujo factible 3.21 y 3.22, i.e. cualquier suma  $\xi + 1 \circ (\phi_1) + \dots + 1 \circ (\phi_l)$  es factible para toda  $1 \leq l \leq k$ , por lo tanto si consideramos el patrón de flujo  $\xi + 1 \circ (\phi_1)$  tenemos :

$$\begin{aligned} c[\xi + 1 \circ (\phi_1)] &= c[\xi] + c[\phi_1 | G(\xi)] \\ &\geq c[\xi] \end{aligned}$$

Consideremos ahora la red incremental  $G^u(\xi + 1 \circ (\phi_1))$  en esta red los arcos que tienen un costo menor, comparado con el costo que tienen en la red  $G^u(\xi)$  son los arcos que tienen un sentido 'inverso' en el circuito  $\phi_1$  (ya que en esos arcos es en donde se va a disminuir las unidades de flujo que se transporta).

Como los flujos  $1 \circ (\phi_1), 1 \circ (\phi_2), \dots$ , son únicos, tales arcos nunca podrán ser usados por ningún otro circuito  $\phi_2, \dots, \phi_k$  por lo tanto:

$$c[\phi_1 | G^u(\xi + 1 \circ (\phi_1))] \geq c[\phi_1 | G^u(\xi)]$$

para toda  $l = 2, \dots, k$

Entonces el costo del patrón de flujo  $\xi + 1 \circ (\phi_1) + 1 \circ (\phi_2)$

es el siguiente:

$$\begin{aligned}
 c[\xi + 1 \cdot (\phi_1) + 1 \cdot (\phi_2)] &= c[\xi + 1 \cdot (\phi_1)] + c[\phi_2 | G^u(\xi + 1 \cdot (\phi_1))] \\
 &\geq c[\xi + 1 \cdot (\phi_1)] + c[\phi_2 | G^u(\xi)] \\
 &\geq c[\xi + 1 \cdot (\phi_1)] \\
 &\geq c[\xi]
 \end{aligned}$$

Continuando con este proceso finalmente llegamos a que  $c[\xi^*] \geq c[\xi]$  lo cual contradice la suposición de que  $\xi^*$  es un flujo de costo mínimo, con lo cual queda demostrado el teorema.

En base a lo establecido por el teorema 3, todo el proceso anterior es requerido para encontrar un flujo de mínimo costo de valor  $v$  en  $G$ .

Se inicia el proceso con un flujo factible  $\xi$  de valor  $v$ , se forma la nueva red  $G^u(\xi)$  y se checa si existen circuitos negativos en dicha red; usando para esto el algoritmo de Floyd establecido en la sección 6.2, si existe un circuito de costo negativo  $\phi$  lo localizamos por medio de la matriz  $\Theta$  de caminos mínimos (sección 6.3).

Una vez que ya detectamos al circuito  $\phi$  cambiamos el flujo que se distribuye en esos arcos, agregando al flujo anterior una cantidad  $+\delta$  ó  $-\delta$  (según como se recorra el arco) de flujo. El valor  $v$  del flujo  $\xi$  permanece constante, y su costo es reducido en  $\delta \cdot c(\phi)$  donde  $c(\phi)$  es el costo del circuito  $\phi$ .

$\delta$  es escogido de tal manera que no rebasa las capacidades de los arcos de  $G^u(\xi)$ , i.e.

$$\delta = \min_{(x_i^u, y_j^u) \in \phi} [q^u(x_i^u, y_j^u)] \quad (6.11)$$

Esto implica que al cambiar el flujo  $\xi$  con  $\delta$  unidades en  $G$ , el nuevo flujo  $\xi'$  es factible. El proceso se vuelve a repetir comenzando con este nuevo flujo  $\xi'$  y generando una nueva red  $G^u(\xi')$ , y checando si ya no existen circuitos negativos una vez más.

Si no existen circuitos negativos entonces el flujo es un flujo de costo mínimo.

## 6.5 DESCRIPCION DEL ALGORITMO

PASO 1.- Use el algoritmo de Ford-Fulkerson (secc 4.3) para encontrar el flujo máximo factible de valor  $v$  en la red  $G$ .

PASO 2.- Respecto al flujo  $\xi$  generar la nueva red  $G^u(\xi)$  (como se explico en la seccion 5.2), i.e.

$$G^u(\xi) = (X^u, A^u, d^u) \quad A^u = A_1^u A_2^u \quad X^u = X$$

$$A_1^u = \{(x^u, y^u)\} \quad \text{con } q^u(x^u, y^u) = q(x, y) - \xi(x, y)$$

capacidad

$$c^u(x^u, y^u) = c(x, y) \cdot \text{costo}$$

$$A_2^u = \{(y^u, x^u)\} \text{ con } q^u(y^u, x^u) = \xi(x, y) \text{ capacidad}$$

$$c^u(y^u, x^u) = -c(x, y) \text{ costo}$$

PASO 3.- Con la matriz de costo  $C$ , generada de la red  $G^u(\xi)$ , (tomando sus valores como punto de partida) aplicamos el algoritmo de Floyd para encontrar cualquier circuito de costo negativo en  $G^u(\xi)$ .

Si tal circuito existe, lo identificamos y pasamos al paso 4. Si tal circuito no existe, el costo ya es mínimo para el flujo  $\xi$ , el algoritmo termina.

PASO 4.- Calcular  $\delta$  de acuerdo a la ecuación 6.11, i.e.

$$\delta = \min [q^u(x_i^u, y_j^u)]$$

$$(x_i^u, y_j^u) \in \Phi$$

con la cantidad  $\delta$  se realiza un cambio de flujo para los arcos contenidos en el circuito  $\Phi$  de la siguiente manera :

- (i) Para todo arco  $(x_i^u, y_j^u) \in \Phi, \gamma c^u(x_i^u, y_j^u) < 0$  se cambia el flujo  $\xi_{ji}$  en el correspondiente arco  $(y_j, x_i)$  de  $G$ , de  $\xi_{ji}$  a  $\xi_{ji} - \delta$ .
- (ii) Para todo arco  $(x_i^u, y_j^u) \in \Phi, \gamma c^u(x_i^u, y_j^u) > 0$  se cambia el flujo  $\xi_{ij}$  en el correspondiente arco  $(x_i, y_j)$  de la red  $G$ , de  $\xi_{ij}$  a  $\xi_{ij} + \delta$ .

PASO 5.- Con el nuevo patrón de flujo regresamos al paso 2.

## VII.-RESUMEN

Resumiendo la secuencia que se sigue en el proceso presentado para la obtención de un Flujo Máximo-Costo Mínimo es la siguiente:

- I.- APLICACION DEL ALGORITMO DE FORD-FULKERSON (secc 4.3)  
para la obtención de un flujo Máximo en una red G.
- II.- APLICACION DEL ALGORITMO BASADO EN LA DETERMINACION DE CIRCUITOS NEGATIVOS
- a) Generar la red incremental  $G^u(\xi)$  (secc 5.2)
- b) Construcción de la matriz de costos C, y la matriz de caminos mínimos  $\theta$ . (secc 6.3)  
(secc 6.2)
- c) Aplicación del algoritmo de Floyd (para la detección de circuitos negativos). (secc 6.2)

## 7.1.-IMPORTANCIA DEL ALGORITMO DE FLOYD DENTRO DEL PROCESO

Como se explicó en la sección 5.2 una red incremental  $G^u(\xi)$  generada a partir de una red  $G$ , y un patrón de flujo en  $G$ , es una red que va a estar formada por el mismo conjunto de vértices ( $X$ ) de la red  $G$ , y por dos conjuntos de arcos;  $A_1^u$  y  $A_2^u$ .

Sabemos que en el conjunto  $A_1^u$  están incluidos todos los arcos de la forma  $(x^u, y^u)$  para los cuales el correspondiente arco  $(x, y) \in G$ , tendrá un flujo  $f(x, y) < q(x, y)$ , a estos arcos se les asocia una capacidad  $q^u(x^u, y^u) = q(x, y) - f(x, y)$  y un costo  $c^u(x^u, y^u) = c(x, y)$ , y tenemos que el conjunto  $A_2^u$  es el conjunto formado por todos los arcos de la forma  $(y^u, x^u)$  para los cuales  $f(x, y) > 0$ ; la capacidad de estos arcos es  $q^u(y^u, x^u) = f(x, y)$ , y su costo es  $c^u(y^u, x^u) = -c(x, y)$ .

Ahora si analizamos los arcos de la red  $G$ , para los cuales  $f(x, y) = q(x, y)$ , i.e. aquellos arcos que están saturados; tendremos la siguiente situación:

Para los arcos de la forma  $(x^u, y^u) \in A_1^u$  tienen costo  $c^u(x^u, y^u) = \infty$  (ver algoritmo Floyd, secc 6.2) ya que son arcos que tendrán una capacidad  $= q^u(x^u, y^u) = q(x, y) - f(x, y) = 0$  (nula).

Sabemos por otro lado que todo arco dentro del Mínimo Corte encontrado por el algoritmo de Ford-Fulkerson debe



estar saturado, i.e.  $c(x_i, y_j) = q(x_i, y_j)$ , por lo que al generar la red incremental este tipo de arcos tendrán asociada una capacidad  $c^u(x_i^u, y_j^u) = c(x_i, y_j) - q(x_i, y_j) = 0$  (nula).

Por esta razón este tipo de arcos no aparecerán en la red  $G^u(\xi)$  y de esta manera su costo asociado dentro de la matriz de costos  $C$  es  $c_{ij} = \infty$  (ver algoritmo de Floyd secc 6.2), esto implica que ninguno de los arcos contenidos en el corte mínimo puede estar incluido en algún circuito negativo, ya que aplicando la triple ecuación (6.11) del algoritmo de Floyd en la búsqueda de un circuito negativo tenemos:

$$c_{ik} = \min(c_{ik}, c_{ij} + c_{jk}) \quad \forall i \neq j \ni c_{ij} \neq \infty \text{ y } \forall j \neq k \ni c_{jk} \neq \infty$$

entonces si  $c_{ij} = \infty$  implica que el camino más corto de  $i$  a  $k$  es  $c_{ik}$ .

En conclusión tenemos que el algoritmo de Floyd va a buscar los circuitos negativos en la red  $G$ , sin incluir aquellos arcos en los que el costo es  $\infty$ , ya que estos nunca estarán en un circuito negativo.

De esta manera el proceso de minimizar el costo de un flujo es bastante rápido.

## VIII.- EJEMPLOS

A continuación mostramos 3 ejemplos, por medio de los cuales se desea ilustrar el funcionamiento del algoritmo basado en la determinación de circuitos negativos.

En el primer ejemplo se trabaja con un patrón de flujo sin que este corresponda al flujo máximo y minimizamos su costo de transportación.

En los ejemplos 2 y 3 se trabaja con un patrón de flujo máximo. Los datos de los ejemplos 1 y 3 fueron probados con el proceso programado, y los resultados obtenidos concuerdan con los que a continuación se dan en cada caso.

EJEMPLO 1.- Sea la red  $G$  mostrada en la fig 8.11, donde el primer número asociado a cada arco representa su capacidad y el segundo, su costo.

Se desea encontrar el Mínimo Costo de un flujo de valor  $v = 20$ , de  $s$  a  $t$ .

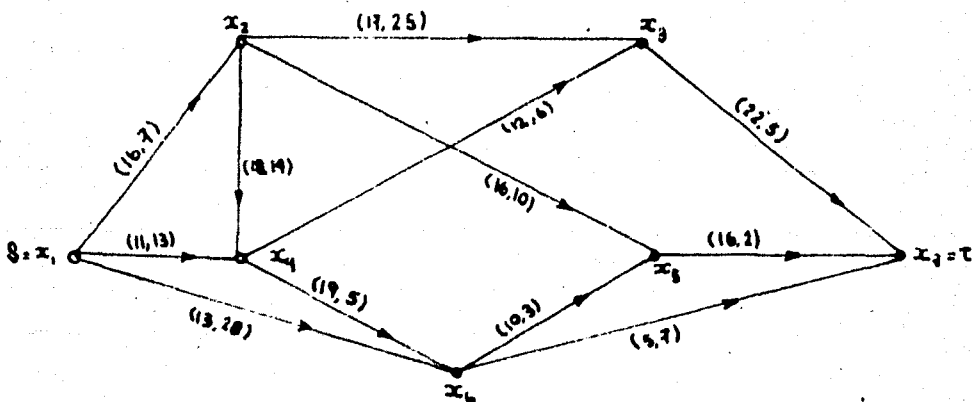


FIG 8.11

La primer etiqueta es la capacidad del arco, y la segunda el costo

PASO 1.- Para este ejemplo en particular trabajaremos con un patrón (arbitrario) de flujo factible de valor  $v = 20$  mostrado en la fig 8.12, el costo de este patrón de flujo es  $16(7+25+5)+4(28+7) = 732$ .

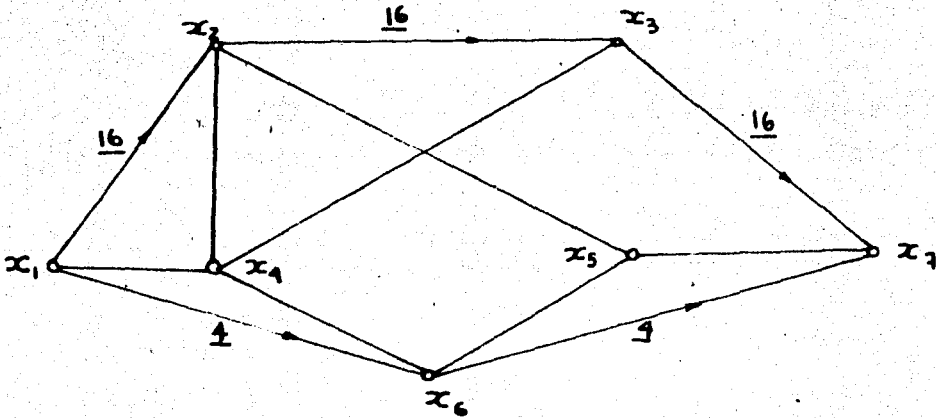


FIG 8.12 Patrón de flujo inicial; costo = 732

PASO 2.- Respecto al patrón de flujo dado formamos la red  $G^u(\xi)$  mostrada en la fig 8.13.

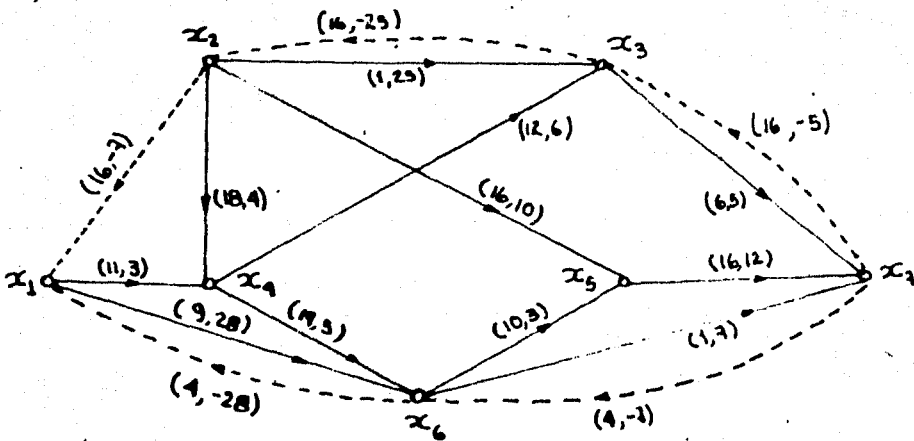


FIG 8.13.- Red  $G^u(\xi)$  respecto al flujo de la fig 8.12

PASO 3.- Formamos la matriz de costos  $C$ , y la matriz de caminos mínimos  $W$ .

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$
$x_1$	0	∞	∞	13	∞	28	∞
$x_2$	-7	0	25	4	10	∞	∞
$x_3$	∞	-25	0	∞	∞	∞	5
$x_4$	∞	∞	6	0	∞	5	∞
$x_5$	∞	∞	∞	∞	0	∞	12
$x_6$	-28	∞	∞	∞	3	0	7
$x_7$	∞	∞	-5	∞	∞	-7	0

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$
$x_1$	1	1	1	1	1	1	1
$x_2$	2	2	2	2	2	2	2
$x_3$	3	3	3	3	3	3	3
$x_4$	4	4	4	4	4	4	4
$x_5$	5	5	5	5	5	5	5
$x_6$	6	6	6	6	6	6	6
$x_7$	7	7	7	7	7	7	7

Matriz de costos de  $G^1$  (f)

Matriz de caminos mínimos

PASO 4.- Aplicando el algoritmo de Floyd vamos obteniendo las siguientes matrices de costo mínimo, junto con su matriz asociada de caminos mínimos. Primera iteración

$k = 1$

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$
$x_1$	0	∞	∞	13	∞	28	∞
$x_2$	-7	0	25	4	10	21	∞
$x_3$	∞	-25	0	∞	∞	∞	5
$x_4$	∞	∞	6	0	∞	5	∞
$x_5$	∞	∞	∞	∞	0	∞	12
$x_6$	-28	∞	∞	-13	3	0	7
$x_7$	∞	∞	-5	∞	∞	-7	0

Matriz de costo

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$
$x_1$	1	1	1	1	1	1	1
$x_2$	2	2	2	2	2	1	2
$x_3$	3	3	3	3	3	3	3
$x_4$	4	4	4	4	4	4	4
$x_5$	5	5	5	5	5	5	5
$x_6$	6	6	6	1	6	6	6
$x_7$	7	7	7	7	7	7	7

Matriz de caminos mínimos

Segunda iteración.-  $k = 2$

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$
$x_1$	0		13			28	
$x_2$	-7	0	25	4	10	21	
$x_3$	-32	-25	0	-21	-15	-4	5
$x_4$			6	0		5	
$x_5$					0		12
$x_6$	-28			-15	3	0	7
$x_7$		-5				-7	0

Matriz de costo Mínimo

Las entradas en blanco son tomadas como infinito ( $\infty$ )

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$
$x_1$	1	1	1	1	1	1	1
$x_2$	2	2	2	2	2	1	2
$x_3$	2	3	3	2	2	1	3
$x_4$	4	4	4	4	4	4	4
$x_5$	5	5	5	5	5	5	5
$x_6$	6	6	6	1	6	6	6
$x_7$	7	7	7	7	7	7	7

Matriz de caminos Mínimos

Tercera iteración.-  $k = 3$

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$
$x_1$	0			13		28	
$x_2$	-7	0	25	4	10	21	30
$x_3$	-32	-25	0	-21	-15	-4	5
$x_4$	-26	-19	6	-15	-9	2	11
$x_5$					0		12
$x_6$	-28			-15	3	0	7
$x_7$	-37	-30	-5	-26	-20	-9	0

Matriz de Costo Mínimo

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$
$x_1$	1	1	1	1	1	1	1
$x_2$	2	2	2	2	2	1	3
$x_3$	2	3	3	2	2	1	3
$x_4$	2	3	4	2	2	1	3
$x_5$	5	5	5	5	5	5	5
$x_6$	6	6	6	6	6	6	6
$x_7$	2	3	7	2	2	1	7

Matriz de caminos Mínimos

En esta iteración encontramos en la matriz de costos mínimos un costo  $c_{44} = -15$ , lo que implica que existe un circuito negativo, que se inicia y termina en el vértice  $x_4$ .

El circuito es identificado por medio de la matriz de caminos mínimos así :

tenemos que el vértice predecesor de  $x_4$  es  $x_2$  ,

el vértice predecesor de  $x_2$  es  $x_3$  ,

y el vértice predecesor de  $x_3$  es  $x_4$  .

Así el circuito negativo es  $(x_4, x_3, x_2, x_4)$ .

PASO 5.- Calculamos el valor  $\delta$  por medio de la condición 6.11

$$\delta = \min [q(x_4, x_3), q(x_3, x_2), q(x_2, x_4)]$$

$$\delta = \min [12, 16, 18] = 12$$

Introduciendo la cantidad  $\delta$  al flujo que circula dentro del circuito tenemos que para el:

arco  $(x_4, x_3)$  tendremos  $f'(x_4, x_3) = f(x_4, x_3) + \delta = 0 + 12 = 12$  ,

arco  $(x_3, x_2)$  tendremos  $f'(x_3, x_2) = f(x_3, x_2) - \delta = 16 - 12 = 4$  ,

arco  $(x_2, x_4)$  tendremos  $f'(x_2, x_4) = f(x_2, x_4) + \delta = 0 + 12 = 12$

El nuevo patrón de flujo después del cambio de flujo dentro del circuito negativo es mostrado en la fig 8.14.

Su costo ahora es:

$$16(7) + 4(25) + 16(5) + 12(4) + 12(6) + 4(28) + 4(7) =$$

$$112 + 100 + 80 + 48 + 72 + 140 = 552$$

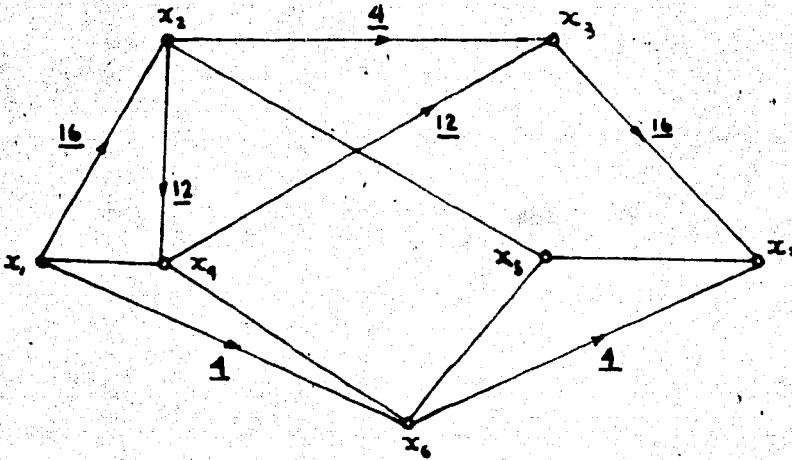


FIG 8.14 Nuevo patrón de flujo  $\xi$ , costo = 552

Regresamos nuevamente al Paso 2, la nueva red  $G^u(\xi)$  es mostrada en la fig 8.15.

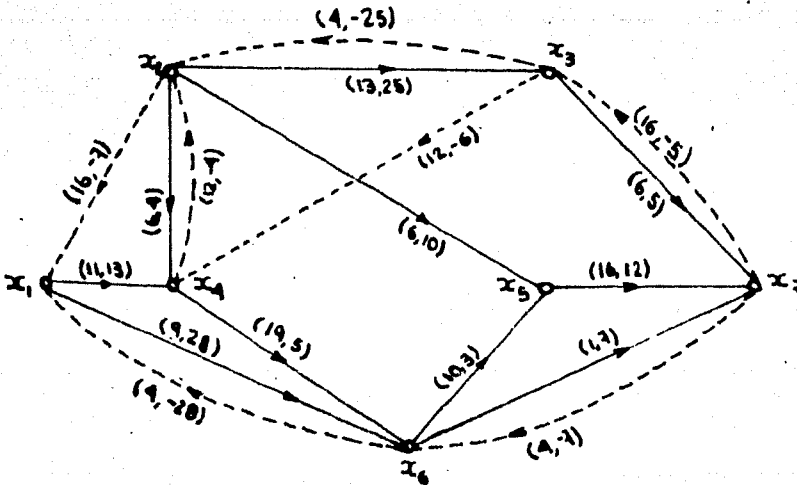


FIG 8.15.-  $G(\xi)$  relativa al patrón de flujo de la fig 8.14

Siguiendo el mismo procedimiento anterior obtenemos los siguientes resultados:

PASO 3.- Formamos la matriz de costo  $C$  respecto a  $G^u(\xi)$  de la fig 8.15.

PASO 4.- El nuevo circuito negativo detectado es  $(x_6, x_1, x_4, x_6)$  con un costo = -10

PASO 5.-  $\delta = \min [q(x_6, x_1), q(x_1, x_4), q(x_4, x_6)]$   
 $\delta = \min [4, 11, 19] = 4$

El nuevo patrón de flujo es mostrado en la figura 8.16 y su costo es 512.

PASO 2.- La nueva red  $G^u(\xi)$  relativa al nuevo patrón de flujo es mostrado en la fig 8.17

PASO 3.- Formamos la matriz de costo  $C$  respecto a la nueva red  $G^u(\xi)$  de la fig 8.17.

PASO 4.- El nuevo circuito negativo detectado es  $(x_7, x_3, x_2, x_5, x_7)$  con un costo de -8.

PASO 5.-  $\delta = \min [q(x_7, x_3), q(x_3, x_2), q(x_2, x_5), q(x_5, x_7)]$   
 $\delta = \min [16, 4, 16, 16] = 4$

El nuevo patrón de flujo es mostrado en la figura 8.18 y su costo es 480.

PASO 2.- La nueva red  $G^u(\xi)$  relativa al patrón de flujo es es mostrada en la fig 8.19.

PASO 3.- Formamos la nueva matriz de costos  $C$  respecto a  $G^u(\xi)$  de la fig 8.19.



PASO 4.- El nuevo circuito negativo detectado es  $(x_7, x_5, x_2, x_4, x_6, x_7)$  con un costo de  $-6$ .

PASO 5.-  $= \min [4, 4, 6, 15, 1] = 1$

El nuevo patrón de flujo es mostrado en la fig 8.20 y su costo es de 474.

A partir de este patrón de flujo  $\xi$  no son encontrados circuitos negativos en la red incremental  $G(\xi)$ ; por lo tanto el patrón de flujo  $\xi$  mostrado en la fig 8.20 es el patrón de flujo de costo mínimo para un flujo de valor  $v = 20$  transportado a través de la red  $G$ , y cuyo costo es 474.

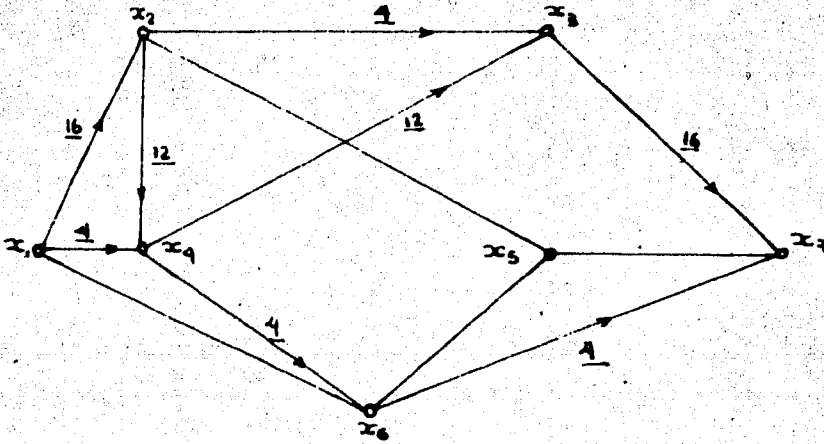


FIG 8.16.- Nuevo patrón de flujo  $\xi$ , costo = 512

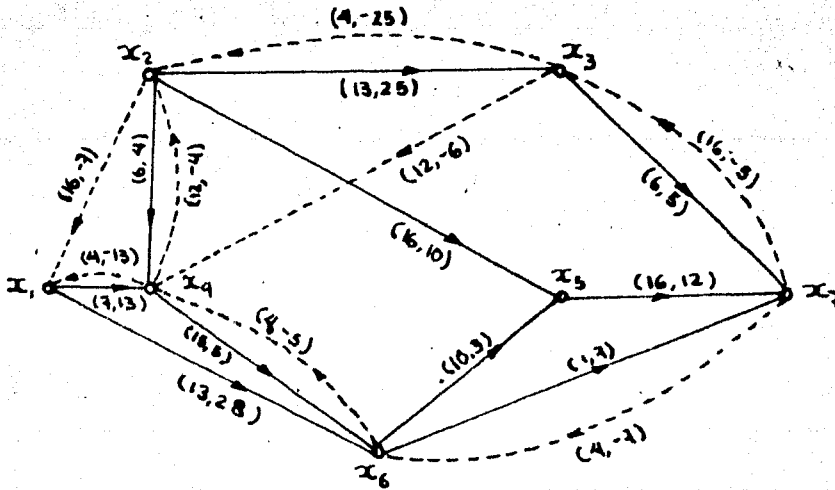


FIG 8.17.-  $G^u(\xi)$  relativa al patrón de flujo de la fig 8.16

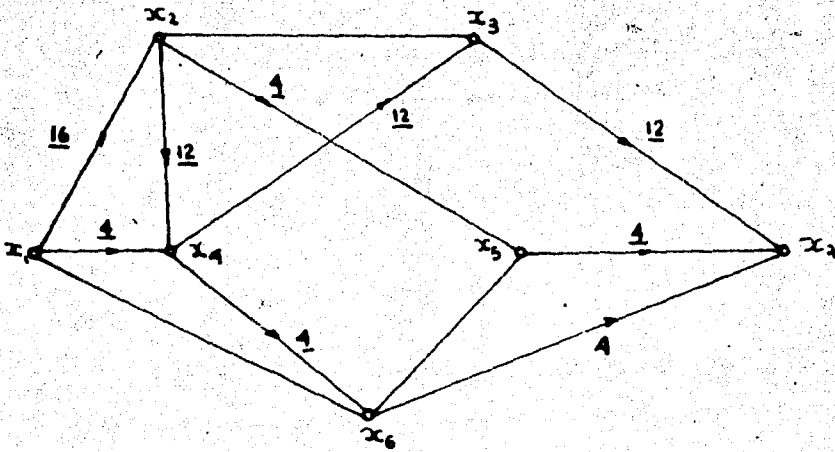


FIG 8.18.- Nuevo patrón de flujo  $\xi$ , costo = 480

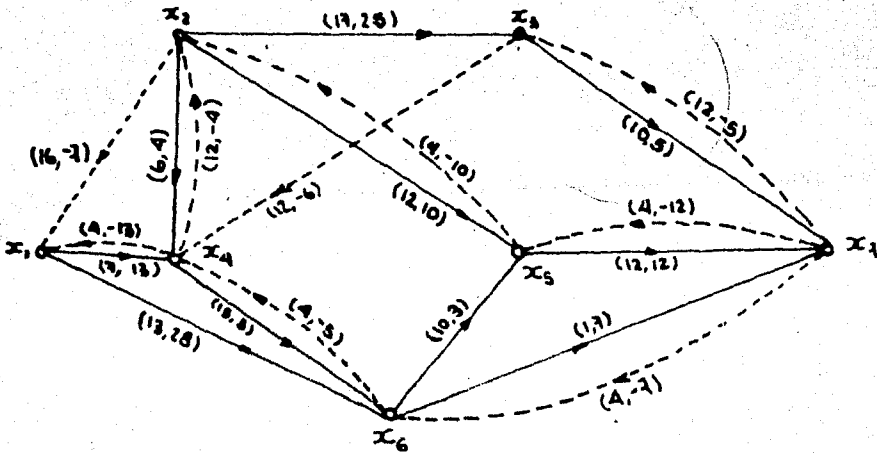


FIG 8.19.-  $G^u(\xi)$  relativa al patrón de flujo  $\xi$  de la fig 8.18

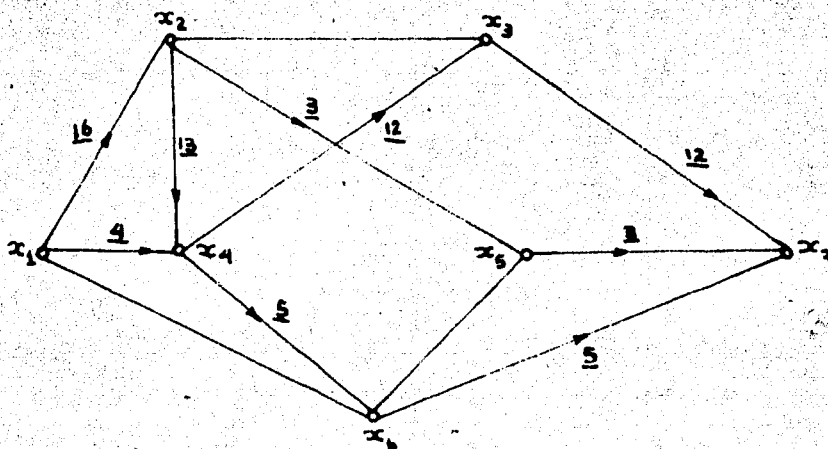


FIG 8.20.- Patrón de flujo  $\xi$  de valor  $v = 20$ , con costo mínimo  $= 474$

.. MATERIAL DE LOS UNIDOS ENTRADA ..  
 .....

MARCA	NODO INICIAL	NODO FINAL	CAPACIDAD	COSTO	FLUJO
0	1	2	10	7	10
0	1	4	11	13	0
0	1	8	13	20	4
0	2	3	17	20	10
0	2	4	18	4	0
0	3	5	16	10	0
0	3	7	22	5	10
0	4	3	12	0	0
0	4	6	19	5	0
0	7	7	16	12	0
0	7	5	5	7	4
0	5	5	10	5	0

TOTAL DE ARCOS DE LA NUEVA RED = 10

Datos del proceso programado correspondientes a la fig 8.12

VECTOR CICLO NEGATIVO

4  
2  
0  
4

COSTO -015  
COSTO-FLUJ 552 VALOR-FLUJO 20

CAMBIO FLUJO CON IFL = 12

•• MATRIZ DE DATOS EN ESA ITERACION ••

MARCA	NODO INICIAL	NODO FINAL	CAPACIDAD	CGSTU	FLUJO
1	1	2	16	7	16
0	1	4	11	13	0
1	1	6	13	20	4
1	2	3	17	25	4
1	2	4	16	4	12
0	2	5	16	10	0
1	3	7	22	5	16
1	4	3	12	6	12
0	4	6	19	5	0
0	5	7	16	12	0
1	6	7	5	7	4
0	6	5	10	3	0

TOTAL DE ARCOS DE LA NUEVA RED = 17

Obtención de resultados a través de proceso programado  
correspondientes a la fig 8.14

VECTOR CICLO NEGATIVO

0  
1  
2  
3

COSTO -010  
COSTO-TOT 512 VALOR-FLUJO 20

CAMBIO FLUJO CON IFL =

3 2 7 7 8 1 1



\*\*\* MATRIZ DE DATOS EN ESA ITERACION \*\*\*  
 .....  
 .....

MARCA	NODO INICIAL	NODO FINAL	CAPACIDAD	COSTO	FLUJO
1	1	2	16	7	16
1	1	4	11	13	4
0	1	6	13	28	0
1	2	3	17	25	4
1	2	4	18	4	12
0	2	5	16	10	0
1	3	7	28	5	16
1	4	3	17	0	12
1	4	6	19	5	4
0	5	7	16	12	0
1	6	7	5	7	4
0	6	5	10	3	0

TOTAL DE ARCOS DE LA NUEVA RED = 18

Otención de resultados a través del proceso programado  
 correspondientes a la fig 8.16

VECTOR CICLO NEGATIVO

7  
3  
2  
3  
7

COSTO -006

COSTO-TOT 490 VALOR-FLUJO 20

CAMBIO FLUJO CON IFL = 4

$\frac{3}{4} = \frac{3}{4}$

•• MATRIZ DE DATOS EN ESA ITERACION ••  
 .....

MARCA	NODO INICIAL	NODO FINAL	CAPACIDAD	COSTO	FLUJO
1	1	2	16	7	16
1	1	4	11	13	4
0	1	6	13	20	0
0	2	3	17	25	0
1	2	4	18	4	12
1	2	5	16	10	4
1	3	7	22	5	12
1	4	3	12	6	12
1	4	6	19	5	4
1	5	7	16	12	4
1	6	7	5	7	4
0	6	5	10	3	0

TOTAL DE ARCOS DE LA NUEVA RED = 19

Obtención de resultados a través del proceso programado  
 correspondientes a la fig 8.18

VECTOR CICLO NEGATIVO

7  
5  
2  
7  
0  
7

COSTO -000  
COSTO-TOT 474 VALOR-FLUJO 20

CAMBIO FLUJO CON IFL = 1

•• MATRIZ DE DATOS EN ESA ITERACION ••

.....

MARCA	NODO INICIAL	NODO FINAL	CAPACIDAD	COSTO	FLUJO
1	1	2	16	7	16
1	1	4	11	13	4
0	1	6	13	28	0
0	2	3	17	25	0
1	2	4	18	4	11
1	2	5	16	10	3
1	3	7	22	5	12
1	4	3	12	6	12
1	4	6	19	5	5
1	5	7	16	12	3
1	6	7	5	7	5
0	6	5	10	3	0

TOTAL DE ARCOS DE LA NUEVA RED = 18

Obtencion de resultados a través de proceso programado  
correspondientes a la fig 8.20

•• MATRIZ DE DATOS DE LA RED FINAL ••  
 .....  
 .....

MARCA	NODO INICIAL	NODO FINAL	CAPACIDAD	COSTO	FLUJO
1	1	2	16	7	10
1	1	4	11	13	4
0	1	6	13	22	0
0	2	3	17	25	0
1	2	4	18	4	13
1	2	5	16	10	3
1	3	7	22	5	12
1	4	3	12	6	12
1	4	6	19	5	5
1	4	7	16	12	3
1	6	7	5	7	5
0	6	5	10	3	0

EL COSTO MINIMO DE TRANSPORTE ES 474

EL VALOR MAXIMO DE FLUJO ES 20

SCAPE PRINT

EJEMPLO 2.- Encontrar el flujo Máximo de Costo Mínimo para la red G de la fig 8.23 por medio del proceso basado en la determinación de circuitos negativos.

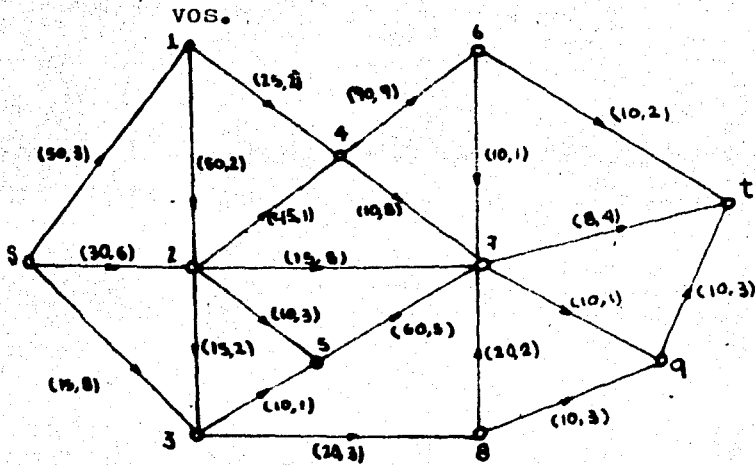


FIG 8.23.- El primer número asociado a cada arco representa la capacidad, y el segundo el costo.

PASO 1.- Aplicando el algoritmo de Ford-Fulkerson se construye el flujo Máximo de valor  $v = 35$ , el patrón de flujo es mostrado en la fig 8.24.

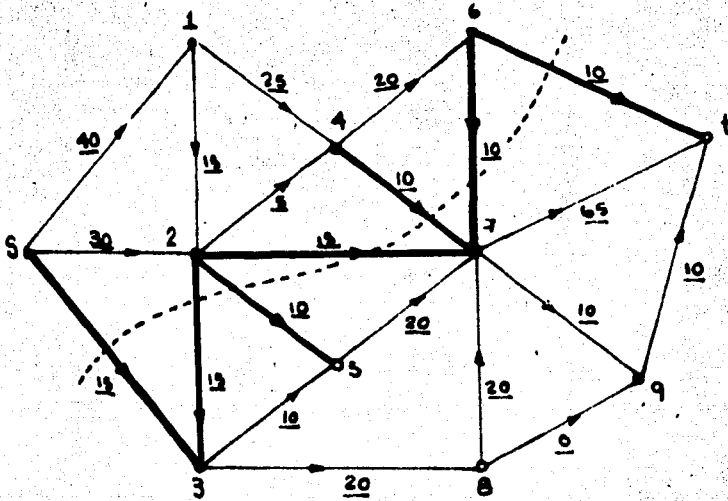


FIG 8.24.- Los arcos del corte mínimo son los arcos marcados, los números indican el flujo en cada arco. El costo de este patrón es = 1485

$$\text{El corte } (Y; \bar{Y}) = \{(s,3), (2,3), (2,5), (2,7), (4,7), (6,7), (6,t)\}$$

$$\text{con } Y = \{s,1,2,4,6\} ; \bar{Y} = \{3,5,7,8,9,t\}$$

PASO 2.- Respecto al patrón de flujo mostrado en la fig 8.24 generamos  $G^u(\mathbb{f})$  mostrada en la fig 8.25.

Aquí observamos que todo arco dentro del corte mínimo está saturado, entonces al descomponer cada arco  $(x,y) \in G$  en dos arcos;  $(y^u, x^u), (x^u, y^u)$  tendremos que  $q(y^u, x^u) = 0$  lo que implica que  $c(y^u, x^u) = \infty$  (secc 5.2 y 6.2).



Por lo cual tenemos que ninguno de los arcos contenidos en el corte mínimo estará contenido en un circuito negativo, por esta razón los podemos excluir del proceso de minimizar el costo, sin que esto altere nuestros resultados.

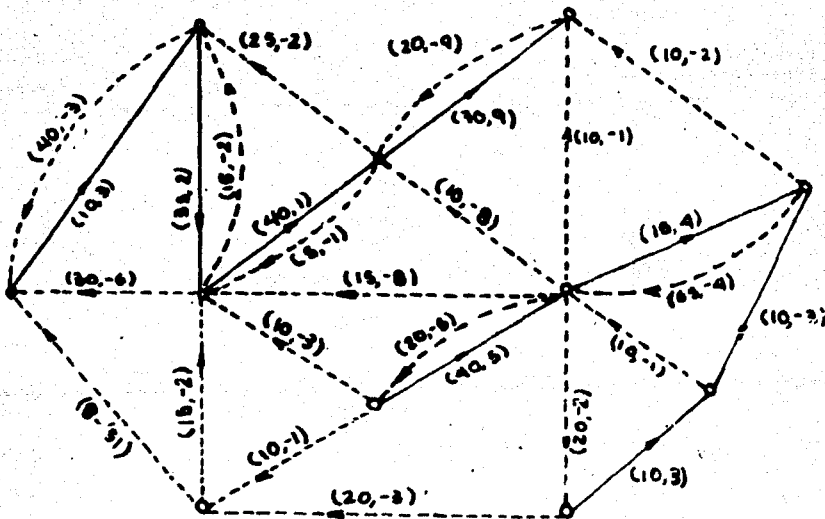


FIG 8.25

PASO 3.- Formamos la matriz de costos  $C$ , y la matriz de caminos mínimos  $\theta$ .

	s	1	2	3	4	5	6	7	8	9	t
s	0	3									
1	-3	0	2								
2	-3	-2	0	1							
3				0							
4		-2	-1		0		9				
5				-1		0		5			
6					-9		0				
7						-5		0	-2		4
8			-3						0	3	
9								-1	3	0	
t								-4		-3	0

Matriz de Costo Mnimo de G (f)

	s	1	2	3	4	5	6	7	8	9	t
s	s	s	s	s	s	s	s	s	s	s	s
1	1	1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7	7	7
8	8	8	8	8	8	8	8	8	8	8	8
9	9	9	9	9	9	9	9	9	9	9	9
t	t	t	t	t	t	t	t	t	t	t	t

Matriz de caminos Mnimos

Las entradas en blanco de la matriz de costo representan  $\infty$ .

PASO 4.- Aplicando el algoritmo de Floyd vamos obteniendo las siguientes matrices C y  $\theta$ .

Primera iteración  $k = s$

	s	1	2	3	4	5	6	7	8	9	t
s	0	3									
1	-3	0	2								
2	-6	-3	0	1							
3				0							
4		-2	-1		0		9				
5				-1		0		5			
6					-9		0				
7						-5		0	-2		4
8				-3					0	3	
9								-1	3	0	
t								-4	-3	0	

Matriz de costo

	s	1	2	3	4	5	6	7	8	9	t
s	s	s	s	s	s	s	s	s	s	s	s
1	1	1	1	1	1	1	1	1	1	1	1
2	2	s	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7	7	7
8	8	8	8	8	8	8	8	8	8	8	8
9	9	9	9	9	9	9	9	9	9	9	9
t	t	t	t	t	t	t	t	t	t	t	t

Matriz de caminos

Segunda iteración  $k = 1$ 

	s	1	2	3	4	5	6	7	8	9	t
s	0	3	5								
1	-3	0	2								
2	-6	-3	-1	1							
3				0							
4		-2	-1		0	9					
5				-1		0	5				
6					-9	0					
7					-5		0	-2		4	
8				-3				0	3		
9							-1	3	0		
t								-4	-3	0	

Matriz de costo mínimo

	s	1	2	3	4	5	6	7	8	9	t
s	s	s	1	s	s	s	s	s	s	s	s
1	1	1	1	1	1	1	1	1	1	1	1
2	2	s	1	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7	7	7
8	8	8	8	8	8	8	8	8	8	8	8
9	9	9	9	9	9	9	9	9	9	9	9
t	t	t	t	t	t	t	t	t	t	t	t

Matriz de caminos Mínimos

En esta iteración encontramos en la matriz de costo mínimo, un costo  $c_{22} = -1$ , lo que implica que existe un circuito negativo que se inicia y termina en el vértice 2.

Identificando el circuito por medio de la matriz de caminos  $\Theta$ , encontramos que el circuito de costo negativo es  $(2, s, 1, 2)$ ; los arcos que forman este circuito son  $\{(2, s), (s, 1), (1, 2)\}$

PASO 5.- Calculamos el valor  $\delta$ .

$$\delta = \min [q(2,s), q(s,1), q(1,2)]$$

$$\delta = \min [30, 10, 15] = 10$$

El nuevo patrón de flujo después de la introducción de la cantidad  $\delta$  es mostrado en la fig 8.26

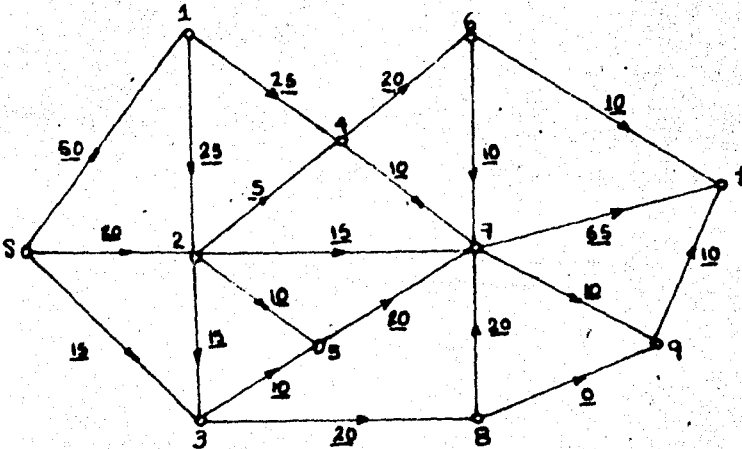


FIG 8.26.- Nuevo patrón de flujo  $\xi$ , costo = 1475

A partir de este patrón de flujo no existen más circuitos negativos en la red, por lo tanto el patrón de flujo mostrado en la fig 8.26 es el patrón del Flujo Máximo de Costo Mínimo para la red del ejemplo 2 (fig 8.23).

EJEMPLO 3.- Sea la red G mostrada en la fig 8.27, donde el primer número asociado a cada arco representa su capacidad y el segundo, su costo.

Se desea conocer cual es el máximo flujo a costo mínimo.

Los datos de esta red son probados con el proceso programado, ver resultados de este problema los obtenidos por el proceso programado.

En la fig 8.28 se muestra el corte mínimo obtenido por el algoritmo de Ford-Fulkerson.

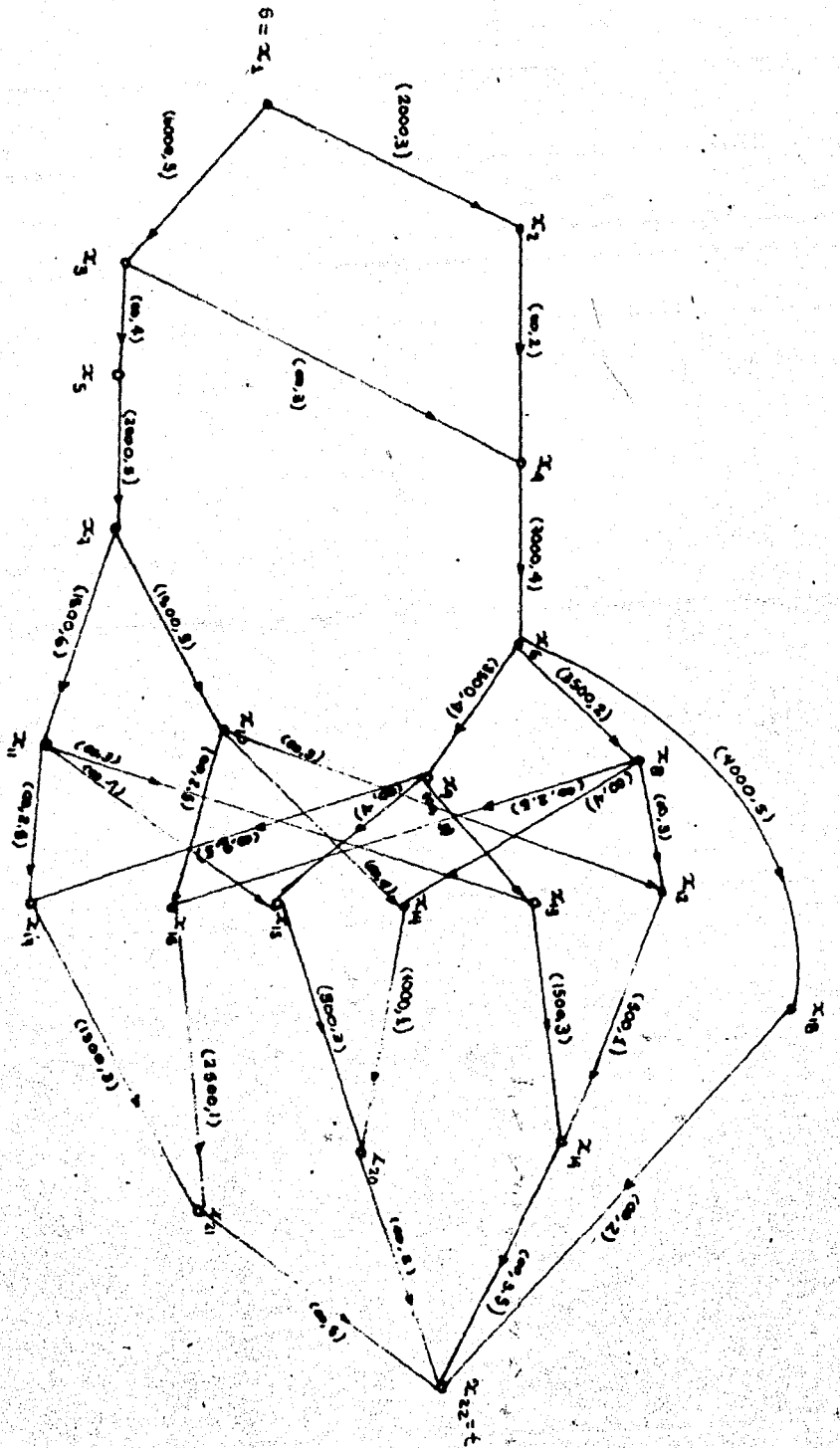


FIGURA B.21

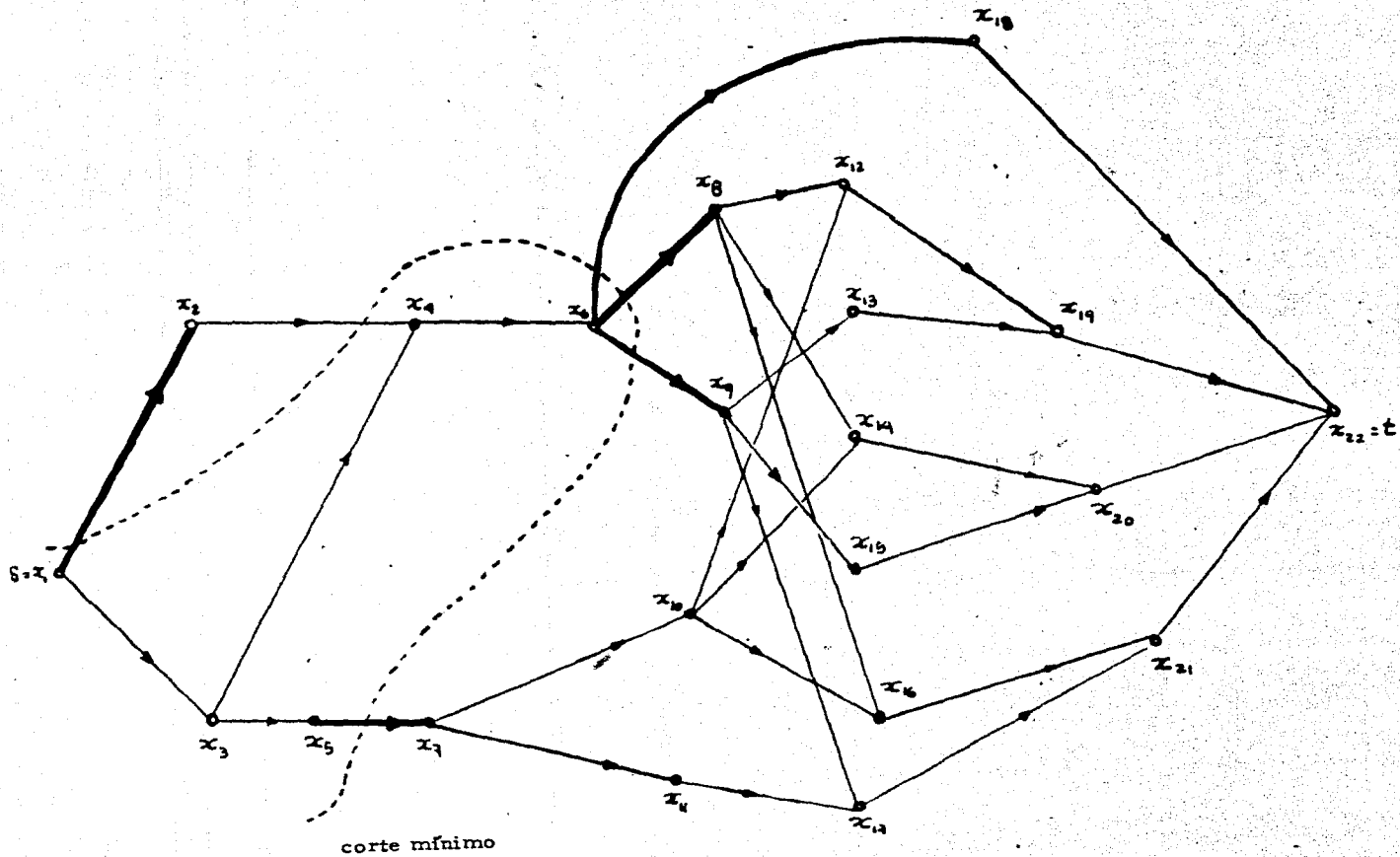


FIG. 8.28

== arcs saturados



\*\* MATRIZ DE LOS DATOS ENTRADA \*\*

MARCA	NODO INICIAL	NODO FINAL	CAPACIDAD	COSTO	FLUJO
0	1	2	2000	3.0	0
0	1	3	8000	5.0	0
0	2	4	10000	2.0	0
0	3	4	10000	3.0	0
0	3	5	10000	4.0	0
0	4	6	7000	4.0	0
0	5	7	3000	5.0	0
0	6	8	3000	2.0	0
0	6	9	3500	4.0	0
0	7	18	4000	5.0	0
0	7	10	1500	3.0	0
0	8	11	1500	6.0	0
0	8	12	10000	3.0	0
0	8	14	10000	4.0	0
0	8	16	10000	2.5	0
0	9	13	10000	3.0	0
0	9	15	10000	4.0	0
0	9	17	10000	2.5	0
0	10	12	10000	3.0	0
0	10	14	10000	4.0	0
0	10	16	10000	2.5	0
0	11	13	10000	3.0	0
0	11	15	10000	4.0	0
0	11	17	10000	2.5	0
0	12	19	500	1.0	0
0	13	19	1500	3.0	0
0	14	20	1000	1.0	0
0	15	20	500	3.0	0
0	16	21	2500	1.0	0
0	17	21	1500	3.0	0
0	18	22	10000	2.0	0
0	19	22	10000	3.5	0
0	20	22	10000	2.0	0
0	21	22	10000	5.0	0

ENTRE A CAMBIO FLUJO CON IC = 500

ENTRE A CAMBIO FLUJO CON IC = 1500

ENTRE A CAMBIO FLUJO CON IC = 500

ENTRE A CAMBIO FLUJO CON IC = 1000

ENTRE A CAMBIO FLUJO CON IC = 500

ENTRE A CAMBIO FLUJO CON IC = 1000

ENTRE A CAMBIO FLUJO CON IC = 1000

ENTRE A CAMBIO FLUJO CON IC = 500

ENTRE A CARLOS FLORES CON IC = 1930

\*\*\*\*\* MATRIZ CON DATOS FLUJO MAXIMO \*\*\*\*\*

NODO INICIAL	NODO FINAL	CAPACIDAD	COSTO	FLUJO
1	2	2000	3.0	2000
1	3	6000	5.0	5500
2	4	10000	2.0	0
3	4	10000	3.0	5500
3	5	10000	4.0	3500
4	6	7000	4.0	6000
5	7	5000	5.0	2000
6	8	3000	2.0	3500
6	9	3500	4.0	3500
6	10	4000	5.0	4000
7	10	1500	3.0	1500
7	11	1000	6.0	1500
8	12	10000	3.0	3500
8	14	10000	4.0	0
8	16	11000	2.5	0
9	13	10000	3.0	3500
9	15	10000	4.0	0
9	17	10000	2.5	0
10	12	10000	3.0	0
10	14	10000	4.0	2500
10	16	10000	2.5	0
11	13	10000	3.0	0
11	15	10000	4.0	1500
11	17	10000	2.5	0
12	19	500	1.0	500
13	19	1000	3.0	1500
14	20	1000	1.0	1000
15	20	500	3.0	500
16	21	2500	1.0	0
17	21	1000	3.0	0
18	22	10000	2.0	4000
19	22	20000	3.0	2000
20	22	10000	2.0	1500
21	22	10000	5.0	0

EL COSTO DE TRANSPORTE ES 215000.000

EL COSTE MINIMO CUMPLE LOS SIG NODOS

- 1
- 3
- 4
- 6
- 9

EL VALOR DEL FLUJO MAXIMO ES V = 7500

\*\*\* MATRIZ DE DATOS DE LA RED FINAL \*\*\*  
 .....

ARCA	NODO INICIAL	NODO FINAL	CAPACIDAD	COSTO	FLUJO
1	1	2	2000	3.0	2500
1	1	3	6000	3.0	5500
0	2	4	10000	2.0	0
1	3	4	10000	3.0	5500
1	3	5	10000	4.0	3500
1	4	6	7000	4.0	6000
1	5	7	2000	3.0	2000
1	6	8	3000	2.0	3500
1	6	9	3000	4.0	3500
1	6	10	4000	3.0	4000
1	7	10	1500	3.0	1500
1	7	11	1500	6.0	1500
1	8	12	10000	3.0	3500
0	8	14	10000	4.0	0
0	8	16	10000	2.5	0
1	9	13	10000	3.0	3500
0	9	15	10000	4.0	0
0	9	17	10000	2.5	0
0	10	12	10000	3.0	0
1	10	14	10000	4.0	2500
0	10	16	10000	2.5	0
0	11	13	10000	3.0	0
1	11	15	10000	4.0	1500
0	11	17	10000	2.5	0
1	12	19	500	1.0	500
1	13	19	1500	3.0	1500
1	14	20	1000	1.0	1000
1	15	20	500	3.0	500
0	16	21	4500	1.0	0
0	17	21	1500	3.0	0
1	18	22	10000	2.0	4000
1	19	22	10000	3.5	2000
1	20	22	10000	2.0	1500
0	21	22	10000	3.0	0

NO EXISTEN CIRCUITOS DE COSTO NEGATIVO

UNAPI PRINT

## IX.- CONCLUSIONES

En el presente trabajo se propone un procedimiento para resolver el problema de Máximo flujo-Mínimo Costo, que consiste en dividir el problema en dos etapas:

- 1) La primera es encontrar el flujo máximo en nuestra red.
- 2) y la segunda es minimizar el costo de transporte del flujo máximo obtenido.

La solución que se encontró para la primera parte del problema es obtenida mediante la aplicación de la técnica desarrollada por Ford-Fulkerson.

Una vez encontrada la solución al problema de máximo flujo, tratamos de encontrar una solución a la segunda parte del problema; para esto se realiza la aplicación de un algoritmo basado en la determinación de circuitos negativos dentro de una red para minimizar costos.

Para explicar el funcionamiento así como las bases del proceso utilizado para la obtención de soluciones a nuestro problema planteado, se da la información necesaria de Teoría de Gráficas así como de Teoría de Redes, misma que es utilizada más adelante en la aplicación de los algoritmos que intervienen en el proceso, así como en el manejo de la estructura de una red.

Se intentó en este aspecto que el trabajo fuera lo más completo posible, sin que por esto perdiera el objetivo principal† además de que el proceso utilizado se pudiera implementar computacionalmente.

Podemos decir que el funcionamiento de este algoritmo comparado con muchos de los algoritmos que se utilizan para resolver el problema de Flujo Máximo-Mínimo Costo es mucho más sencillo debido al manejo de datos, y a la forma de hacerlo, que el algoritmo requiere.

T.C.Hu en su libro de Integer Programming and Network Flows (secc 10.4) propone un procedimiento similar para resolver el problema; consistente también en la búsqueda de circuitos negativos, pero excluyendo del proceso solo a los arcos del corte mínimo (para los cuales no existirán circuitos negativos).

Debido a esto, construye 2 matrices : una que contiene los vértices contenidos en el conjunto  $X$  del corte mínimo, y la otra que contiene los vértices del conjunto  $\bar{Y}$  del corte mínimo.

Cosa que no sucede en el procedimiento que se propone en este trabajo, ya que la búsqueda de los circuitos negativos se realiza en una sola matriz de datos.

Usando el algoritmo de Floyd para la detección de dichos circuitos un investigador llamado Bennington publico una experiencia computacional (Nicos Christofides, capítulo 11, secc 5.1.3) indicando que el tiempo de máquina para encontrar la solución en una red con  $n$  vértices y  $m$  arcos es :

$$t = -2.3 + 0.0113(n) + 0.00166(m) \quad (\text{SEC IBM 360/75})$$

† obtener una solución al problema Flujo Máximo-Costo Mínimo.

Así en una red de 200 vértices y 500 arcos se requiere aproximadamente 8 segundos.

T.C.Hu (sec 10.4) comprueba que la búsqueda de una solución al problema de flujo-Costo mínimo mediante la determinación de circuitos de costo negativo tiene una eficacia equivalente respecto a los algoritmos que utilizan la técnica de Programación Lineal Primal-Dual.

Desde el punto de vista computacional es mucho más sencillo el manejo de datos, así como la obtención de una solución debido a que son mucho menos variables las que se manejan en el proceso.

El algoritmo de Floyd utilizado para la detección de los circuitos negativos en una red es el algoritmo fundamental en este proceso, ya que la forma en que realiza la búsqueda de los circuitos negativos permite que se excluyan todos aquellos arcos de la red los cuales no es posible que estén incluidos en algún circuito negativo; de esta manera la búsqueda es rápida, ya que no se incluyen en ella todos los arcos de la red.

Finalmente el trabajo presenta la programación del proceso aplicada a dos ejemplos.

Es importante comentar que los datos que constituyen la estructura de una red son leídos dentro del proceso progra-

madó a través de v́ectores de tal manera que la memoria de máquina que es utilizada sea mínima, esto es con el fin de que dicho proceso pueda ser aplicado a problemas de Redes bastante grandes.



## X.- REFERENCIAS.

## I.- GRAPH THEORY

AN ALGORITHMIC APPROACH

Nicos Chistofides

Management Science

Imperial College

London, 1975

## II.- INTEGER PROGRAMMING

AND NETWORK FLOWS

T.C.Hu

Professor, Mathematics Research Center

and

Department of Computer Sciences

University of Wisconsin, 1969

## III.- FLOWS IN NETWORKS

L.R. Ford, Jr. and D.R Fulkerson

Princeton Univ. Press, Princeton, N.J. 1962

## ARTICULOS

## IV.- ALGORITHM FOR SOLUTION OF A PROBLEM

OF MAXIMUM FLOW IN A NETWORK WITH

POWER ESTIMATION

E.A. Dinic

Moscow State University

18/Feb/1970

V.- DETERMINING THE MAXIMAL FLOW IN  
A NETWORK BY THE METHOD OF PRE-FLOWS

A.V. Karzanov

Institute for Control Problems

Soviet Math. Dokl

16/Apr/1973

VI.- SHORTEST PATH AND NETWORK FLOW  
ALGORITHMS

Eugene L. Lawler

Computer Science Division

Department of Electrical Engineering

and Computer Sciences and The Elec-

tronics Research Laboratory

University of California

at Berkeley

August, 1977

**XI.- APENDICE**

**1.- ALGORITMO PARA MINIMIZAR EL COSTO DE UN FLUJO  
BASADO EN LA DETERMINACIÓN DE CIRCUITOS NEGATIVOS**

**2.- DIAGRAMA DE BLOQUE DE LA PROGRAMACION DEL  
PROCESO PARA MINIMIZAR EL COSTO DE UN FLUJO**

**2.1.- Comentarios referentes a la estructura del programa.  
(diagramas de flujo de cada subrutina).**

1.- ALGORITMO BASADO EN LA DETERMINACION DE  
CIRCUITOS NEGATIVOS

PASO 1.- Use el algoritmo de Ford-Fulkerson (secc 4.3)  
para encontrar el flujo máximo factible.

PASO 2.- Generar la gráfica  $G^u(\xi)$  (secc 5.2) respecto al  
flujo  $\xi$ .

PASO 3.- Aplicar el algoritmo de Floyd (secc 6.2) para  
encontrar cualquier circuito de costo negativo  
en  $G^u(\xi)$ .

Si tal circuito existe, identificarlo e ir al  
paso 4. Si tal circuito no existe, el costo para  
el flujo  $\xi$  sera mínimo y el algoritmo termina.

PASO 4.- Calcular  $\delta$  según la ecuación 6.11

$$\delta = \min_{(x_i^u, y_j^u) \in \phi} [c^u(x_i^u, y_j^u)]$$

Cambiar el flujo a los arcos contenidos en el cir-  
cuito  $\phi$  de la siguiente manera :

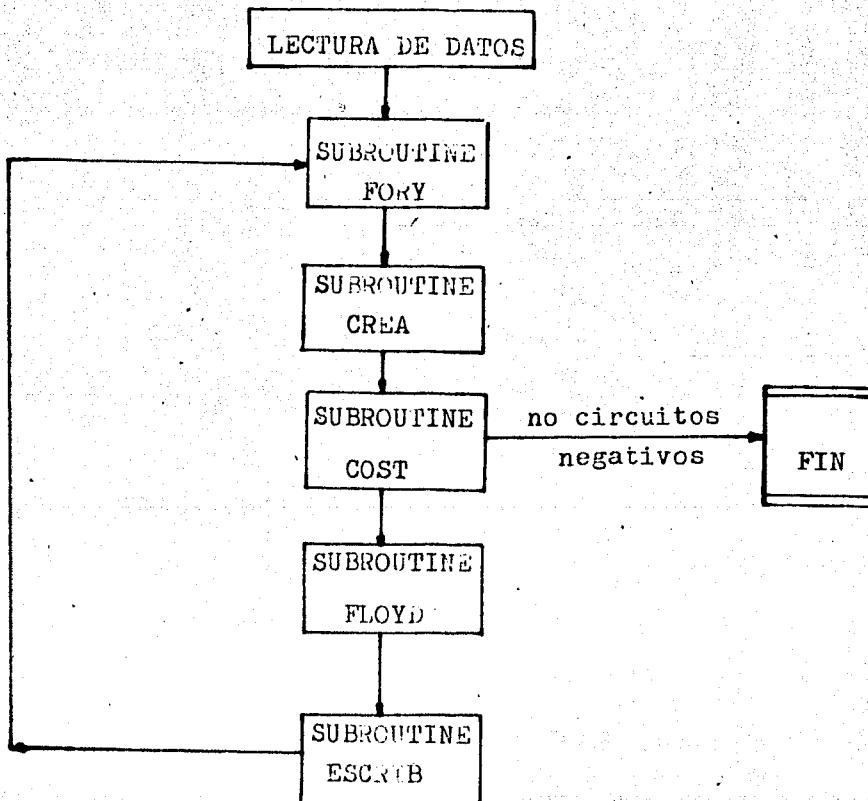
- (i) Para todo arco  $(x_i^u, y_j^u)$  con  $c^u(x_i^u, y_j^u) < 0$  se cam-  
bia el flujo  $\xi_{ji}$  en el correspondiente arco  $(y_j, x_i)$   
de  $G$ , de  $\xi_{ji}$  a  $\xi_{ji} - \delta$ .
- (ii) Para todo arco  $(x_i, y_j)$  con  $c^u(x_i^u, y_j^u) > 0$  se cambia  
el flujo  $\xi_{ij}$  en el correspondiente

arco  $(x_i, y_j)$  de la red  $G$ , de  $\xi_{ij}$  a  $\xi_{ij} + \delta$ .

PASO 5.- Con el nuevo patrón de flujo  $\xi$  regresar al paso 2.

PROCESO PARA ENCONTRAR EL FLUJO MAXIMO  
MINIMO COSTO BASADO EN LA DETERMINACION DE  
CIRCUITOS NEGATIVOS

DIAGRAMA DE BLOQUE











```

00456 171* 97 WRITE(6,*)
00457 172* 98 FORMAT(10,'LARGO')
00458 173* 99 TO 10
00459 174* 100 DO 105 IJ = 1,N
00460 175* IF ( IFLUJ(IJ) .EQ. 0 ) GO TO 105
00461 176* MARCAT(IJ) = 1
00462 177* CONTINUE
00463 178* DO 135 JJ = 1,N
00464 179* IF(IFLUJ(IJ) .EQ. 0) GO TO 135
00465 180* CUIZ = COSTZ+(FLUJ(IJ))*COST(JJ)
00466 181* 130 IF(NODO(IJ) .NE. 1) GO TO 135
00467 182* VALOR = VALOR + IFLUJ(IJ)
00468 183* CONTINUE
00469 184* WRITE(6,202)
00470 185* 202 FORMAT(1F10.3,*,** MATRIZ CON DATOS FLUJO MAXIMO **//,39*,
00471 186* -'*****'//,7X,*,
00472 187* -'NODOS',1X,*,NODO,1X,*,CAPACIDAD,1X,*,COSTO,5X,*,FLUJO,*,
00473 188* -'COSTO INICIAL',9X,*,FINAL'//)
00474 189* DO 1 I=1,N
00475 190* WRITE(6,201) NODO(I),NODO(I),ICA(I),COST(I),IFLUJ(I)
00476 191* 32 FORMAT(1X,14,3X,14,3X,16,7X,F6.1,4X,13)
00477 192* 33 CONTINUE
00478 193* WRITE(6,309) COSTZ
00479 194* 309 FORMAT(177,40X,'EL COSTO DE TRANSPORTE ES ',3X,F15.3)
00480 195* WRITE(6,201)VALOR
00481 196* 201 FORMAT(177,24X,'EL CORTE MINIMO CONTIENE LOS SIG NODOS',10X,
00482 197* 'EL VALOR DEL FLUJO MAXIMO ES',70X,'V = ',2X,19)
00483 198* DO 14 I = 1,N-1
00484 199* WRITE(6,201) I,I+1
00485 200* 20 FORMAT(137A,13)
00486 201* 19 CONTINUE
00487 202* RETURN
00488 203*
00489 204*
00490 205*
00491 206*
00492 207*
00493 208*
00494 209*
00495 210*
00496 211*
00497 212*
00498 213*
00499 214*
00500 215*
00501 216*
00502 217*
00503 218*
00504 219*
00505 220*
00506 221*
00507 222*
00508 223*
00509 224*
00510 225*
00511 226*
00512 227*
00513 228*
00514 229*
00515 230*
00516 231*
00517 232*
00518 233*
00519 234*
00520 235*
00521 236*
00522 237*
00523 238*
00524 239*
00525 240*
00526 241*
00527 242*
00528 243*
00529 244*
00530 245*
00531 246*
00532 247*
00533 248*
00534 249*
00535 250*
00536 251*
00537 252*
00538 253*
00539 254*
00540 255*
00541 256*
00542 257*
00543 258*
00544 259*
00545 260*
00546 261*
00547 262*
00548 263*
00549 264*
00550 265*
00551 266*
00552 267*
00553 268*
00554 269*
00555 270*
00556 271*
00557 272*
00558 273*
00559 274*
00560 275*
00561 276*
00562 277*
00563 278*
00564 279*
00565 280*
00566 281*
00567 282*
00568 283*
00569 284*
00570 285*
00571 286*
00572 287*
00573 288*
00574 289*
00575 290*
00576 291*
00577 292*
00578 293*
00579 294*
00580 295*
00581 296*
00582 297*
00583 298*
00584 299*
00585 300*
00586 301*
00587 302*
00588 303*
00589 304*
00590 305*
00591 306*
00592 307*
00593 308*
00594 309*
00595 310*
00596 311*
00597 312*
00598 313*
00599 314*
00600 315*
00601 316*
00602 317*
00603 318*
00604 319*
00605 320*
00606 321*
00607 322*
00608 323*
00609 324*
00610 325*
00611 326*
00612 327*
00613 328*
00614 329*
00615 330*
00616 331*
00617 332*
00618 333*
00619 334*
00620 335*
00621 336*
00622 337*
00623 338*
00624 339*
00625 340*
00626 341*
00627 342*
00628 343*
00629 344*
00630 345*
00631 346*
00632 347*
00633 348*
00634 349*
00635 350*
00636 351*
00637 352*
00638 353*
00639 354*
00640 355*
00641 356*
00642 357*
00643 358*
00644 359*
00645 360*
00646 361*
00647 362*
00648 363*
00649 364*
00650 365*
00651 366*
00652 367*
00653 368*
00654 369*
00655 370*
00656 371*
00657 372*
00658 373*
00659 374*
00660 375*
00661 376*
00662 377*
00663 378*
00664 379*
00665 380*
00666 381*
00667 382*
00668 383*
00669 384*
00670 385*
00671 386*
00672 387*
00673 388*
00674 389*
00675 390*
00676 391*
00677 392*
00678 393*
00679 394*
00680 395*
00681 396*
00682 397*
00683 398*
00684 399*
00685 400*
00686 401*
00687 402*
00688 403*
00689 404*
00690 405*
00691 406*
00692 407*
00693 408*
00694 409*
00695 410*
00696 411*
00697 412*
00698 413*
00699 414*
00700 415*
00701 416*
00702 417*
00703 418*
00704 419*
00705 420*
00706 421*
00707 422*
00708 423*
00709 424*
00710 425*
00711 426*
00712 427*
00713 428*
00714 429*
00715 430*
00716 431*
00717 432*
00718 433*
00719 434*
00720 435*
00721 436*
00722 437*
00723 438*
00724 439*
00725 440*
00726 441*
00727 442*
00728 443*
00729 444*
00730 445*
00731 446*
00732 447*
00733 448*
00734 449*
00735 450*
00736 451*
00737 452*
00738 453*
00739 454*
00740 455*
00741 456*
00742 457*
00743 458*
00744 459*
00745 460*
00746 461*
00747 462*
00748 463*
00749 464*
00750 465*
00751 466*
00752 467*
00753 468*
00754 469*
00755 470*
00756 471*
00757 472*
00758 473*
00759 474*
00760 475*
00761 476*
00762 477*
00763 478*
00764 479*
00765 480*
00766 481*
00767 482*
00768 483*
00769 484*
00770 485*
00771 486*
00772 487*
00773 488*
00774 489*
00775 490*
00776 491*
00777 492*
00778 493*
00779 494*
00780 495*
00781 496*
00782 497*
00783 498*
00784 499*
00785 500*
00786 501*
00787 502*
00788 503*
00789 504*
00790 505*
00791 506*
00792 507*
00793 508*
00794 509*
00795 510*
00796 511*
00797 512*
00798 513*
00799 514*
00800 515*
00801 516*
00802 517*
00803 518*
00804 519*
00805 520*
00806 521*
00807 522*
00808 523*
00809 524*
00810 525*
00811 526*
00812 527*
00813 528*
00814 529*
00815 530*
00816 531*
00817 532*
00818 533*
00819 534*
00820 535*
00821 536*
00822 537*
00823 538*
00824 539*
00825 540*
00826 541*
00827 542*
00828 543*
00829 544*
00830 545*
00831 546*
00832 547*
00833 548*
00834 549*
00835 550*
00836 551*
00837 552*
00838 553*
00839 554*
00840 555*
00841 556*
00842 557*
00843 558*
00844 559*
00845 560*
00846 561*
00847 562*
00848 563*
00849 564*
00850 565*
00851 566*
00852 567*
00853 568*
00854 569*
00855 570*
00856 571*
00857 572*
00858 573*
00859 574*
00860 575*
00861 576*
00862 577*
00863 578*
00864 579*
00865 580*
00866 581*
00867 582*
00868 583*
00869 584*
00870 585*
00871 586*
00872 587*
00873 588*
00874 589*
00875 590*
00876 591*
00877 592*
00878 593*
00879 594*
00880 595*
00881 596*
00882 597*
00883 598*
00884 599*
00885 600*
00886 601*
00887 602*
00888 603*
00889 604*
00890 605*
00891 606*
00892 607*
00893 608*
00894 609*
00895 610*
00896 611*
00897 612*
00898 613*
00899 614*
00900 615*
00901 616*
00902 617*
00903 618*
00904 619*
00905 620*
00906 621*
00907 622*
00908 623*
00909 624*
00910 625*
00911 626*
00912 627*
00913 628*
00914 629*
00915 630*
00916 631*
00917 632*
00918 633*
00919 634*
00920 635*
00921 636*
00922 637*
00923 638*
00924 639*
00925 640*
00926 641*
00927 642*
00928 643*
00929 644*
00930 645*
00931 646*
00932 647*
00933 648*
00934 649*
00935 650*
00936 651*
00937 652*
00938 653*
00939 654*
00940 655*
00941 656*
00942 657*
00943 658*
00944 659*
00945 660*
00946 661*
00947 662*
00948 663*
00949 664*
00950 665*
00951 666*
00952 667*
00953 668*
00954 669*
00955 670*
00956 671*
00957 672*
00958 673*
00959 674*
00960 675*
00961 676*
00962 677*
00963 678*
00964 679*
00965 680*
00966 681*
00967 682*
00968 683*
00969 684*
00970 685*
00971 686*
00972 687*
00973 688*
00974 689*
00975 690*
00976 691*
00977 692*
00978 693*
00979 694*
00980 695*
00981 696*
00982 697*
00983 698*
00984 699*
00985 700*
00986 701*
00987 702*
00988 703*
00989 704*
00990 705*
00991 706*
00992 707*
00993 708*
00994 709*
00995 710*
00996 711*
00997 712*
00998 713*
00999 714*
01000 715*
01001 716*
01002 717*
01003 718*
01004 719*
01005 720*
01006 721*
01007 722*
01008 723*
01009 724*
01010 725*
01011 726*
01012 727*
01013 728*
01014 729*
01015 730*
01016 731*
01017 732*
01018 733*
01019 734*
01020 735*
01021 736*
01022 737*
01023 738*
01024 739*
01025 740*
01026 741*
01027 742*
01028 743*
01029 744*
01030 745*
01031 746*
01032 747*
01033 748*
01034 749*
01035 750*
01036 751*
01037 752*
01038 753*
01039 754*
01040 755*
01041 756*
01042 757*
01043 758*
01044 759*
01045 760*
01046 761*
01047 762*
01048 763*
01049 764*
01050 765*
01051 766*
01052 767*
01053 768*
01054 769*
01055 770*
01056 771*
01057 772*
01058 773*
01059 774*
01060 775*
01061 776*
01062 777*
01063 778*
01064 779*
01065 780*
01066 781*
01067 782*
01068 783*
01069 784*
01070 785*
01071 786*
01072 787*
01073 788*
01074 789*
01075 790*
01076 791*
01077 792*
01078 793*
01079 794*
01080 795*
01081 796*
01082 797*
01083 798*
01084 799*
01085 800*
01086 801*
01087 802*
01088 803*
01089 804*
01090 805*
01091 806*
01092 807*
01093 808*
01094 809*
01095 810*
01096 811*
01097 812*
01098 813*
01099 814*
01100 815*
01101 816*
01102 817*
01103 818*
01104 819*
01105 820*
01106 821*
01107 822*
01108 823*
01109 824*
01110 825*
01111 826*
01112 827*
01113 828*
01114 829*
01115 830*
01116 831*
01117 832*
01118 833*
01119 834*
01120 835*
01121 836*
01122 837*
01123 838*
01124 839*
01125 840*
01126 841*
01127 842*
01128 843*
01129 844*
01130 845*
01131 846*
01132 847*
01133 848*
01134 849*
01135 850*
01136 851*
01137 852*
01138 853*
01139 854*
01140 855*
01141 856*
01142 857*
01143 858*
01144 859*
01145 860*
01146 861*
01147 862*
01148 863*
01149 864*
01150 865*
01151 866*
01152 867*
01153 868*
01154 869*
01155 870*
01156 871*
01157 872*
01158 873*
01159 874*
01160 875*
01161 876*
01162 877*
01163 878*
01164 879*
01165 880*
01166 881*
01167 882*
01168 883*
01169 884*
01170 885*
01171 886*
01172 887*
01173 888*
01174 889*
01175 890*
01176 891*
01177 892*
01178 893*
01179 894*
01180 895*
01181 896*
01182 897*
01183 898*
01184 899*
01185 900*
01186 901*
01187 902*
01188 903*
01189 904*
01190 905*
01191 906*
01192 907*
01193 908*
01194 909*
01195 910*
01196 911*
01197 912*
01198 913*
01199 914*
01200 915*
01201 916*
01202 917*
01203 918*
01204 919*
01205 920*
01206 921*
01207 922*
01208 923*
01209 924*
01210 925*
01211 926*
01212 927*
01213 928*
01214 929*
01215 930*
01216 931*
01217 932*
01218 933*
01219 934*
01220 935*
01221 936*
01222 937*
01223 938*
01224 939*
01225 940*
01226 941*
01227 942*
01228 943*
01229 944*
01230 945*
01231 946*
01232 947*
01233 948*
01234 949*
01235 950*
01236 951*
01237 952*
01238 953*
01239 954*
01240 955*
01241 956*
01242 957*
01243 958*
01244 959*
01245 960*
01246 961*
01247 962*
01248 963*
01249 964*
01250 965*
01251 966*
01252 967*
01253 968*
01254 969*
01255 970*
01256 971*
01257 972*
01258 973*
01259 974*
01260 975*
01261 976*
01262 977*
01263 978*
01264 979*
01265 980*
01266 981*
01267 982*
01268 983*
01269 984*
01270 985*
01271 986*
01272 987*
01273 988*
01274 989*
01275 990*
01276 991*
01277 992*
01278 993*
01279 994*
01280 995*
01281 996*
01282 997*
01283 998*
01284 999*
01285 1000*

```



00725 235\*  
 00726 236\*  
 00730 237\*  
 00732 238\*  
 00733 239\*  
 00734 240\*  
 00736 241\*  
 00737 242\*  
 00740 243\*  
 00742 244\*  
 00743 245\*  
 00745 246\*  
 00746 247\*  
 00747 248\*  
 00747 249\*  
 00747 250\*  
 00747 251\*  
 00747 252\*  
 00747 253\*  
 00747 254\*  
 00747 255\*  
 00747 256\*  
 00752 257\*  
 00753 258\*  
 00754 259\*  
 00755 260\*  
 00759 261\*  
 00757 262\*  
 00763 263\*  
 00761 264\*  
 00762 265\*  
 00763 266\*  
 00765 267\*  
 00765 268\*  
 00767 269\*  
 00772 270\*  
 00773 271\*  
 00774 272\*  
 00775 273\*  
 00776 274\*  
 00777 275\*  
 00778 276\*  
 00779 277\*  
 00780 278\*  
 00781 279\*  
 00782 280\*  
 00783 281\*  
 00784 282\*  
 00785 283\*  
 00786 284\*  
 00787 285\*  
 00788 286\*  
 00789 287\*  
 00790 288\*  
 00791 289\*  
 00792 290\*  
 00793 291\*  
 00794 292\*  
 00795 293\*  
 00796 294\*  
 00797 295\*  
 00798 296\*  
 00799 297\*  
 00800 298\*  
 00801 299\*  
 00802 300\*  
 00803 301\*  
 00804 302\*  
 00805 303\*  
 00806 304\*  
 00807 305\*  
 00808 306\*  
 00809 307\*  
 00810 308\*  
 00811 309\*  
 00812 310\*  
 00813 311\*  
 00814 312\*  
 00815 313\*  
 00816 314\*  
 00817 315\*  
 00818 316\*  
 00819 317\*  
 00820 318\*  
 00821 319\*  
 00822 320\*  
 00823 321\*  
 00824 322\*  
 00825 323\*  
 00826 324\*  
 00827 325\*  
 00828 326\*  
 00829 327\*  
 00830 328\*  
 00831 329\*  
 00832 330\*  
 00833 331\*  
 00834 332\*  
 00835 333\*  
 00836 334\*  
 00837 335\*  
 00838 336\*  
 00839 337\*  
 00840 338\*  
 00841 339\*  
 00842 340\*  
 00843 341\*  
 00844 342\*  
 00845 343\*  
 00846 344\*  
 00847 345\*  
 00848 346\*  
 00849 347\*  
 00850 348\*  
 00851 349\*  
 00852 350\*  
 00853 351\*  
 00854 352\*  
 00855 353\*  
 00856 354\*  
 00857 355\*  
 00858 356\*  
 00859 357\*  
 00860 358\*  
 00861 359\*  
 00862 360\*  
 00863 361\*  
 00864 362\*  
 00865 363\*  
 00866 364\*  
 00867 365\*  
 00868 366\*  
 00869 367\*  
 00870 368\*  
 00871 369\*  
 00872 370\*  
 00873 371\*  
 00874 372\*  
 00875 373\*  
 00876 374\*  
 00877 375\*  
 00878 376\*  
 00879 377\*  
 00880 378\*  
 00881 379\*  
 00882 380\*  
 00883 381\*  
 00884 382\*  
 00885 383\*  
 00886 384\*  
 00887 385\*  
 00888 386\*  
 00889 387\*  
 00890 388\*  
 00891 389\*  
 00892 390\*  
 00893 391\*  
 00894 392\*  
 00895 393\*  
 00896 394\*  
 00897 395\*  
 00898 396\*  
 00899 397\*  
 00900 398\*  
 00901 399\*  
 00902 400\*  
 00903 401\*  
 00904 402\*  
 00905 403\*  
 00906 404\*  
 00907 405\*  
 00908 406\*  
 00909 407\*  
 00910 408\*  
 00911 409\*  
 00912 410\*  
 00913 411\*  
 00914 412\*  
 00915 413\*  
 00916 414\*  
 00917 415\*  
 00918 416\*  
 00919 417\*  
 00920 418\*  
 00921 419\*  
 00922 420\*  
 00923 421\*  
 00924 422\*  
 00925 423\*  
 00926 424\*  
 00927 425\*  
 00928 426\*  
 00929 427\*  
 00930 428\*  
 00931 429\*  
 00932 430\*  
 00933 431\*  
 00934 432\*  
 00935 433\*  
 00936 434\*  
 00937 435\*  
 00938 436\*  
 00939 437\*  
 00940 438\*  
 00941 439\*  
 00942 440\*  
 00943 441\*  
 00944 442\*  
 00945 443\*  
 00946 444\*  
 00947 445\*  
 00948 446\*  
 00949 447\*  
 00950 448\*  
 00951 449\*  
 00952 450\*  
 00953 451\*  
 00954 452\*  
 00955 453\*  
 00956 454\*  
 00957 455\*  
 00958 456\*  
 00959 457\*  
 00960 458\*  
 00961 459\*  
 00962 460\*  
 00963 461\*  
 00964 462\*  
 00965 463\*  
 00966 464\*  
 00967 465\*  
 00968 466\*  
 00969 467\*  
 00970 468\*  
 00971 469\*  
 00972 470\*  
 00973 471\*  
 00974 472\*  
 00975 473\*  
 00976 474\*  
 00977 475\*  
 00978 476\*  
 00979 477\*  
 00980 478\*  
 00981 479\*  
 00982 480\*  
 00983 481\*  
 00984 482\*  
 00985 483\*  
 00986 484\*  
 00987 485\*  
 00988 486\*  
 00989 487\*  
 00990 488\*  
 00991 489\*  
 00992 490\*  
 00993 491\*  
 00994 492\*  
 00995 493\*  
 00996 494\*  
 00997 495\*  
 00998 496\*  
 00999 497\*  
 01000 498\*  
 01001 499\*  
 01002 500\*  
 01003 501\*  
 01004 502\*  
 01005 503\*  
 01006 504\*  
 01007 505\*  
 01008 506\*  
 01009 507\*  
 01010 508\*  
 01011 509\*  
 01012 510\*  
 01013 511\*  
 01014 512\*  
 01015 513\*  
 01016 514\*  
 01017 515\*  
 01018 516\*  
 01019 517\*  
 01020 518\*  
 01021 519\*  
 01022 520\*  
 01023 521\*  
 01024 522\*  
 01025 523\*  
 01026 524\*  
 01027 525\*  
 01028 526\*  
 01029 527\*  
 01030 528\*  
 01031 529\*  
 01032 530\*  
 01033 531\*  
 01034 532\*  
 01035 533\*  
 01036 534\*  
 01037 535\*  
 01038 536\*  
 01039 537\*  
 01040 538\*  
 01041 539\*  
 01042 540\*  
 01043 541\*  
 01044 542\*  
 01045 543\*  
 01046 544\*  
 01047 545\*  
 01048 546\*  
 01049 547\*  
 01050 548\*  
 01051 549\*  
 01052 550\*  
 01053 551\*  
 01054 552\*  
 01055 553\*  
 01056 554\*  
 01057 555\*  
 01058 556\*  
 01059 557\*  
 01060 558\*  
 01061 559\*  
 01062 560\*  
 01063 561\*  
 01064 562\*  
 01065 563\*  
 01066 564\*  
 01067 565\*  
 01068 566\*  
 01069 567\*  
 01070 568\*  
 01071 569\*  
 01072 570\*  
 01073 571\*  
 01074 572\*  
 01075 573\*  
 01076 574\*  
 01077 575\*  
 01078 576\*  
 01079 577\*  
 01080 578\*  
 01081 579\*  
 01082 580\*  
 01083 581\*  
 01084 582\*  
 01085 583\*  
 01086 584\*  
 01087 585\*  
 01088 586\*  
 01089 587\*  
 01090 588\*  
 01091 589\*  
 01092 590\*  
 01093 591\*  
 01094 592\*  
 01095 593\*  
 01096 594\*  
 01097 595\*  
 01098 596\*  
 01099 597\*  
 01100 598\*  
 01101 599\*  
 01102 600\*  
 01103 601\*  
 01104 602\*  
 01105 603\*  
 01106 604\*  
 01107 605\*  
 01108 606\*  
 01109 607\*  
 01110 608\*  
 01111 609\*  
 01112 610\*  
 01113 611\*  
 01114 612\*  
 01115 613\*  
 01116 614\*  
 01117 615\*  
 01118 616\*  
 01119 617\*  
 01120 618\*  
 01121 619\*  
 01122 620\*  
 01123 621\*  
 01124 622\*  
 01125 623\*  
 01126 624\*  
 01127 625\*  
 01128 626\*  
 01129 627\*  
 01130 628\*  
 01131 629\*  
 01132 630\*  
 01133 631\*  
 01134 632\*  
 01135 633\*  
 01136 634\*  
 01137 635\*  
 01138 636\*  
 01139 637\*  
 01140 638\*  
 01141 639\*  
 01142 640\*  
 01143 641\*  
 01144 642\*  
 01145 643\*  
 01146 644\*  
 01147 645\*  
 01148 646\*  
 01149 647\*  
 01150 648\*  
 01151 649\*  
 01152 650\*  
 01153 651\*  
 01154 652\*  
 01155 653\*  
 01156 654\*  
 01157 655\*  
 01158 656\*  
 01159 657\*  
 01160 658\*  
 01161 659\*  
 01162 660\*  
 01163 661\*  
 01164 662\*  
 01165 663\*  
 01166 664\*  
 01167 665\*  
 01168 666\*  
 01169 667\*  
 01170 668\*  
 01171 669\*  
 01172 670\*  
 01173 671\*  
 01174 672\*  
 01175 673\*  
 01176 674\*  
 01177 675\*  
 01178 676\*  
 01179 677\*  
 01180 678\*  
 01181 679\*  
 01182 680\*  
 01183 681\*  
 01184 682\*  
 01185 683\*  
 01186 684\*  
 01187 685\*  
 01188 686\*  
 01189 687\*  
 01190 688\*  
 01191 689\*  
 01192 690\*  
 01193 691\*  
 01194 692\*  
 01195 693\*  
 01196 694\*  
 01197 695\*  
 01198 696\*  
 01199 697\*  
 01200 698\*  
 01201 699\*  
 01202 700\*  
 01203 701\*  
 01204 702\*  
 01205 703\*  
 01206 704\*  
 01207 705\*  
 01208 706\*  
 01209 707\*  
 01210 708\*  
 01211 709\*  
 01212 710\*  
 01213 711\*  
 01214 712\*  
 01215 713\*  
 01216 714\*  
 01217 715\*  
 01218 716\*  
 01219 717\*  
 01220 718\*  
 01221 719\*  
 01222 720\*  
 01223 721\*  
 01224 722\*  
 01225 723\*  
 01226 724\*  
 01227 725\*  
 01228 726\*  
 01229 727\*  
 01230 728\*  
 01231 729\*  
 01232 730\*  
 01233 731\*  
 01234 732\*  
 01235 733\*  
 01236 734\*  
 01237 735\*  
 01238 736\*  
 01239 737\*  
 01240 738\*  
 01241 739\*  
 01242 740\*  
 01243 741\*  
 01244 742\*  
 01245 743\*  
 01246 744\*  
 01247 745\*  
 01248 746\*  
 01249 747\*  
 01250 748\*  
 01251 749\*  
 01252 750\*  
 01253 751\*  
 01254 752\*  
 01255 753\*  
 01256 754\*  
 01257 755\*  
 01258 756\*  
 01259 757\*  
 01260 758\*  
 01261 759\*  
 01262 760\*  
 01263 761\*  
 01264 762\*  
 01265 763\*  
 01266 764\*  
 01267 765\*  
 01268 766\*  
 01269 767\*  
 01270 768\*  
 01271 769\*  
 01272 770\*  
 01273 771\*  
 01274 772\*  
 01275 773\*  
 01276 774\*  
 01277 775\*  
 01278 776\*  
 01279 777\*  
 01280 778\*  
 01281 779\*  
 01282 780\*  
 01283 781\*  
 01284 782\*  
 01285 783\*  
 01286 784\*  
 01287 785\*  
 01288 786\*  
 01289 787\*  
 01290 788\*  
 01291 789\*  
 01292 790\*  
 01293 791\*  
 01294 792\*  
 01295 793\*  
 01296 794\*  
 01297 795\*  
 01298 796\*  
 01299 797\*  
 01300 798\*  
 01301 799\*  
 01302 800\*  
 01303 801\*  
 01304 802\*  
 01305 803\*  
 01306 804\*  
 01307 805\*  
 01308 806\*  
 01309 807\*  
 01310 808\*  
 01311 809\*  
 01312 810\*  
 01313 811\*  
 01314 812\*  
 01315 813\*  
 01316 814\*  
 01317 815\*  
 01318 816\*  
 01319 817\*  
 01320 818\*  
 01321 819\*  
 01322 820\*  
 01323 821\*  
 01324 822\*  
 01325 823\*  
 01326 824\*  
 01327 825\*  
 01328 826\*  
 01329 827\*  
 01330 828\*  
 01331 829\*  
 01332 830\*  
 01333 831\*  
 01334 832\*  
 01335 833\*  
 01336 834\*  
 01337 835\*  
 01338 836\*  
 01339 837\*  
 01340 838\*  
 01341 839\*  
 01342 840\*  
 01343 841\*  
 01344 842\*  
 01345 843\*  
 01346 844\*  
 01347 845\*  
 01348 846\*  
 01349 847\*  
 01350 848\*  
 01351 849\*  
 01352 850\*  
 01353 851\*  
 01354 852\*  
 01355 853\*  
 01356 854\*  
 01357 855\*  
 01358 856\*  
 01359 857\*  
 01360 858\*  
 01361 859\*  
 01362 860\*  
 01363 861\*  
 01364 862\*  
 01365 863\*  
 01366 864\*  
 01367 865\*  
 01368 866\*  
 01369 867\*  
 01370 868\*  
 01371 869\*  
 01372 870\*  
 01373 871\*  
 01374 872\*  
 01375 873\*  
 01376 874\*  
 01377 875\*  
 01378 876\*  
 01379 877\*  
 01380 878\*  
 01381 879\*  
 01382 880\*  
 01383 881\*  
 01384 882\*  
 01385 883\*  
 01386 884\*  
 01387 885\*  
 01388 886\*  
 01389 887\*  
 01390 888\*  
 01391 889\*  
 01392 890\*  
 01393 891\*  
 01394 892\*  
 01395 893\*  
 01396 894\*  
 01397 895\*  
 01398 896\*  
 01399 897\*  
 01400 898\*  
 01401 899\*  
 01402 900\*  
 01403 901\*  
 01404 902\*  
 01405 903\*  
 01406 904\*  
 01407 905\*  
 01408 906\*  
 01409 907\*  
 01410 908\*  
 01411 909\*  
 01412 910\*  
 01413 911\*  
 01414 912\*  
 01415 913\*  
 01416 914\*  
 01417 915\*  
 01418 916\*  
 01419 917\*  
 01420 918\*  
 01421 919\*  
 01422 920\*  
 01423 921\*  
 01424 922\*  
 01425 923\*  
 01426 924\*  
 01427 925\*  
 01428 926\*  
 01429 927\*  
 01430 928\*  
 01431 929\*  
 01432 930\*  
 01433 931\*  
 01434 932\*  
 01435 933\*  
 01436 934\*  
 01437 935\*  
 01438 936\*  
 01439 937\*  
 01440 938\*  
 01441 939\*  
 01442 940\*  
 01443 941\*  
 01444 942\*  
 01445 943\*  
 01446 944\*  
 01447 945\*  
 01448 946\*  
 01449 947\*  
 01450 948\*  
 01451 949\*  
 01452 950\*  
 01453 951\*  
 01454 952\*  
 01455 953\*  
 01456 954\*  
 01457 955\*  
 01458 956\*  
 01459 957\*  
 01460 958\*  
 01461 959\*  
 01462 960\*  
 01463 961\*  
 01464 962\*  
 01465 963\*  
 01466 964\*  
 01467 965\*  
 01468 966\*  
 01469 967\*  
 01470 968\*  
 01471 969\*  
 01472 970\*  
 01473 971\*  
 01474 972\*  
 01475 973\*  
 01476 974\*  
 01477 975\*  
 01478 976\*  
 01479 977\*  
 01480 978\*  
 01481 979\*  
 01482 980\*  
 01483 981\*  
 01484 982\*  
 01485 983\*  
 01486 984\*  
 01487 985\*  
 01488 986\*  
 01489 987\*  
 01490 988\*  
 01491 989\*  
 01492 990\*  
 01493 991\*  
 01494 992\*  
 01495 993\*  
 01496 994\*  
 01497 995\*  
 01498 996\*  
 01499 997\*  
 01500 998\*  
 01501 999\*  
 01502 1000\*

```

75 IN = IN + 1
   IF (IR .GT. 9) GO TO 80
   IF ((IR*IN) .LT. 0) GO TO 78
   GO TO 75
76 ISN = 
```

01030	042*	IFL = PA	002457
01037	043*	I=VALOR = C	002461
01040	044*	COT = 0	002462
01041	045*	I = 0	002463
01042	046*	K = 1	002464
01043	047*	DO 125 I = 1,A2	002466
01040	046*	J = 0	002504
01047	049*	U(K*J) = A1(I)	002504
01050	052*	U(K*2) = A1(I + 1)	002507
01051	051*	90 J= J + 1	002513
01052	052*	U(K*1) = MODU(J)	002515
01053	053*	U(K*2) = MODU(J)	002522
01054	054*	IF MODU(J) .EQ. A1(I) GO TO 110	002527
01055	055*	IF MODU(J) .NE. A1(I + 1) GO TO 130	002532
01060	056*	IF MODU(J) .NE. A1(I) GO TO 130	002540
01062	057*	IFLOW(J) = IFLOW(J) - IFL	002543
01063	058*	IF IFLOW(J) .NE. 0 GO TO 120	002544
01065	059*	MARCAT(J) = 0	002550
01066	060*	GO TO 125	002551
01067	061*	110 IF MODU(J) .NE. A1(I + 1) GO TO 130	002553
01071	062*	IFLOW(J) = IFLOW(J) + IFL	002557
01072	063*	120 MARCAT(J) = 1	002565
01073	064*	GO TO 125	002567
01074	065*	130 IF (LST. N) GO TO 90	002571
01076	068*	WRITE(10,134)	002574
01100	067*	134 FORMAT(//10X,'ERROR 1')	002604
01111	066*	125 CONTINUE	002604
01165	064*	DO 135 J=1,N	002604
01166	067*	IF IFLOW(J) .EQ. 0 GO TO 130	002613
01168	071*	COT = COT + (IFLOW(J))*(COST(J))	002614
01171	072*	130 IF MODU(J) .NE. 1 GO TO 135	002623
01173	073*	I=VALOR = IVALOR + IFLOW(J)	002625
01174	074*	135 CONTINUE	002634
01176	075*	RETURN	002634
01177	076*	C *****	002644
01177	077*	SUBROUTINE ESCRITA(I,IFL,N,C,COT,I=VALOR,CG,IR,ISA)	003001
01177	078*	C	003001
01177	079*	C	003001
01177	080*	C	003001
01177	081*	C	003001
01177	082*	C	003001
01182	083*	DIMENSION C(N,N),A1(JG)	003001
01183	084*	INTEGER C,A1	003001
01184	085*	IF (ISA .EQ. 3) GO TO 304	003001
01185	086*	WRITE(6,10)	003004
01186	087*	10 FORMAT(10I,///,10X,'VECTOR CICLO NEGATIVO')	003011
01187	088*	WRITE(6,11)(A1(I2),I2 = 1,JG)	003011
01187	089*	51 FORMAT(10A,15)	003032
01188	090*	WRITE(6,30)(IR,IN)	003032
01188	091*	30 FORMAT(///,10X,'COSTO',5X,J4)	003042
01189	092*	WRITE(6,62) COT,I=VALOR	003042
01190	093*	62 FORMAT(10X,'COSTO-TOT',F15.3,3X,'VALOR=FLUJO',I9)	003051
01191	094*	WRITE(6,53) IFL	003051
01194	095*	50 FORMAT(///,10X,'CAMBIO FLUJO CON IFL =',4X,I4,///)	003060
01195	096*	20- WRITE(6,203)	003064
01197	097*	203 FORMAT(10I,40A,'** MATRIZ DE DATOS DE LA RED FINAL **',41X)	003064
01197	098*	-*****//,27A,'MARCA',0X	003064

0117 044  
0117 000  
0117 001  
0117 002  
0117 003  
0117 004  
0117 005  
0117 006  
END FOR

```
-INDD*1A*INDD*BX*CAPACIDAD*1A*CS10*5X*FLUJO*//  
-BX*INICIAL*1X*FINAL*//  
DO 10 I=1,N  
WRITE(6,15) MARCAT(I),NDDO(I),NDDOF(I),ICA(I),CCST(I),IFLUJ(I)  
15 FORMAT(29A,12,7A,14,5A,14,8X,16,7A,16,4A,15)  
16 CONTINUE  
RETURN  
END
```

003064  
003064  
003064  
003071  
003104  
003104  
003104  
003147

## 2.1.- COMENTARIOS REFERENTES A LA ESTRUCTURA DEL PROGRAMA

La programación del proceso propuesto para la solución del problema de Flujo Máximo-Costo Mínimo se realizó en el lenguaje de Fortran V y fue probado en el sistema de la máquina UNIVAC 1180.

El programa consta de 5 subrutinas (presentadas en el diagrama de bloque anterior).

A continuación se da una explicación de la parte del proceso que se realiza en cada una de ellas, así como de la forma de obtención de los datos.

### LECTURA DE DATOS.

Un punto importante para el proceso es la representación de la estructura de una red a través de vectores datos ya que a través de estos se especifica claramente la estructura de la red sobre la cual se va a trabajar.

Para lograr esto se asocia a cada vértice un número (en orden ascendente) de tal manera que al vértice  $s$  (entrada) le corresponda un número  $n_s$  y al vértice  $t$  (salida) le corresponda el número  $n_t$ ; de esta manera el conjunto de vértices  $X$  estará representado por un vector  $(n_1, n_2, \dots, n_t)$  en el cual nosotros podemos saber claramente que el vértice  $s$  es  $n_s$  y  $t$  es  $n_t$ . En otras palabras tenemos que a la fuente de una

red se le asocia siempre  $n_1$  y a la salida  $n_t$ .

El primer paso dentro del proceso es la lectura de los datos.

La dimensión de cada vector dato sera determinada por las variables  $n$  y  $m$ , donde  $n$  = número de nodos de la red, y  $m$  = número de arcos de la red.

Los vectores datos son los siguientes:

(dentro del proceso se leen en esta secuencia)

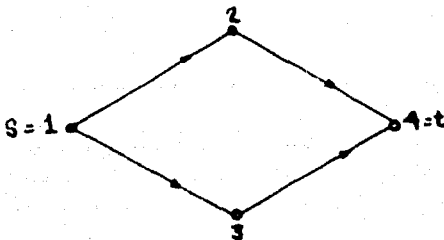
1.- Vector que contiene el conjunto de nodos iniciales de cada arco de la red, su dimensión es =  $m$ .

(se debe dar en orden ascendente)

2).- Vector que contiene el conjunto de nodos finales de cada arco de la red, su dimensión es =  $m$ .

Los 2 vectores anteriores se deben leer en esta secuencia, ya que a través de ellos estamos dando el sentido de orientación correspondiente a cada arco.

Por ejemplo:



1	-----	2
1	-----	3
2	-----	4
3	-----	4
vector 1		vector 2



Así se indica que existe un arco de la forma (1,2),  
(1,3),(2,4),(3,4).

El motivo por el cual los datos se dan de esta manera es con objeto de ahorrar memoria, ya que si se guardan en una matriz, tendremos que todas las entradas de la diagonal no son necesarias como datos para el proceso.

- 3).- Vector que contiene el conjunto de capacidades asociadas a cada arco, su dimensión es  $= m$ .
- 4).- Vector que contiene el conjunto de costos asociados a cada arco, su dimensión  $= m$ .
- 5).- Vector que contiene el conjunto de flujos transportados en cada arco, su dimensión  $= m$ .

## SUBROUTINE FORY

(paso 1 del algoritmo).

Dentro de esta rutina se encuentra el flujo Máximo de (s) a (t) dentro de una red G dada; esto es mediante la programación del algoritmo de Marcaje propuesto por Ford-fulkerson (secc 4.3).

Los datos de entrada que intervienen en esta rutina se leen al principio del proceso y son los siguientes:

## DATOS 1

- |                              |               |
|------------------------------|---------------|
| 1) vector de nodos iniciales | dimensión = m |
| 2) vector de nodos finales   | dimensión = m |
| 3) vector de capacidades     | dimensión = m |
| 4) vector de flujos          | dimensión = m |

Al aplicar la rutina Fory a los datos anteriores obtendremos al final del proceso de esta rutina un patrón de flujo  $f$  que será el flujo máximo dentro de la red, así como el corte mínimo para la misma.

## SUBROUTINE CREA

(paso 2 del algoritmo).

En esta rutina se genera la red incremental  $G^u(\xi)$  respecto al patrón de flujo  $\xi$  obtenido en la rutina FORY, de acuerdo como se indica en la secc 5.2 y asociando un costo a cada arco así:

$$\text{Para todo arco } (x_i^u, y_j^u) \in A_1^u \quad c^u(x_i^u, y_j^u) = c(x_i, y_j)$$

$$\text{Para todo arco } (y_j^u, x_i^u) \in A_2^u \quad c^u(y_j^u, x_i^u) = -c(x_i, y_j)$$

Los datos de entrada que intervienen en esta rutina se leen al principio del proceso a excepción del vector de flujos que se obtiene en la rutina anterior (FORY):

DATOS 2

- 1) Vector de capacidades asociadas a cada arco dimensión = m
- 2) Vector de costos asociados a cada arco dimensión = m
- 3) Vector de flujos asociados a cada arco. dimensión = m

Al final de esta rutina se obtienen los datos por medio de los cuales podemos identificar perfectamente a la red  $G^u(\xi)$ .

## SUBROUTINE CCST

(paso 3 del algoritmo)

Dentro de esta rutina se construye la matriz de Costo  $C$  asociada a la red  $G^u(\xi)$ , y la matriz de caminos mínimos  $\Theta$ , (secc 6.2, 6.3).

Una vez construidas estas dos matrices de dimensión  $n \times n$ , se aplica el algoritmo de Floyd para buscar todos los posibles circuitos negativos existentes en la red  $G^u(\xi)$ .

En el caso de encontrar un circuito negativo; lo localizamos perfectamente en la red  $G^u(\xi)$  mediante la matriz  $\Theta$ , y el proceso continua con la subroutine FLOYD.

Si no es posible encontrar un circuito negativo; entonces la solución ha sido encontrada y el proceso termina.

Los datos de entrada que intervienen en esta rutina se leen al principio del proceso unos, y otros se obtienen de las rutinas anteriores.

DAFOS 3

1) Vector de costos asociados a cada arco

de la red  $G^u(\xi)$                       dimensión =  $kz$  donde  $kz \leq 2m$

NOTA.- La nueva red  $C(\xi)$  tendra a lo más 2 veces el número de arcos de la red  $G$  (secc 5.2)

## SUBROUTINE CCST

(paso 3 del algoritmo)

Dentro de esta rutina se construye la matriz de Costo  $C$  asociada a la red  $G^u(\xi)$ , y la matriz de caminos mínimos  $\theta$ , (secc 6.2,6.3).

Una vez construidas estas dos matrices de dimensión  $n \times n$ , se aplica el algoritmo de Floyd para buscar todos los posibles circuitos negativos existentes en la red  $G^u(\xi)$ .

En el caso de encontrar un circuito negativo; lo localizamos perfectamente en la red  $G^u(\xi)$  mediante la matriz  $\theta$ , y el proceso continua con la subroutine FLOYD.

Si no es posible encontrar un circuito negativo; entonces la solución ha sido encontrada y el proceso termina.

Los datos de entrada que intervienen en esta rutina se leen al principio del proceso unos, y otros se obtienen de las rutinas anteriores.

DAOS 3

1) Vector de costos asociados a cada arco

de la red  $G^u(\xi)$                       dimensión =  $kz$  donde  $kz \leq 2m$

NOTA.- La nueva red  $G(\xi)$  tendra a lo más 2 veces el número de arcos de la red  $G$  (secc 5.2)

## SUBROUTINE FLOYD

(paso 4 del algoritmo).

En esta rutina se calcula la cantidad de flujo  $\delta$  según la ecuación 6.11, para lo cual realiza una búsqueda con el fin de obtener todas las capacidades de los arcos incluidos en el circuito negativo.

$$\text{Una vez conocida } \delta = \min [q^u(x_i^u, y_j^u)] \\ (x_i^u, y_j^u) \in \phi$$

se cambia el flujo a los arcos contenidos en el circuito (en la gráfica G) según lo indica el paso 4 del algoritmo basado en la determinación de circuitos negativos.

Así obtenemos como resultado un nuevo patrón de flujo con un costo menor al costo del patrón anterior.

Los datos necesarios en esta rutina son:

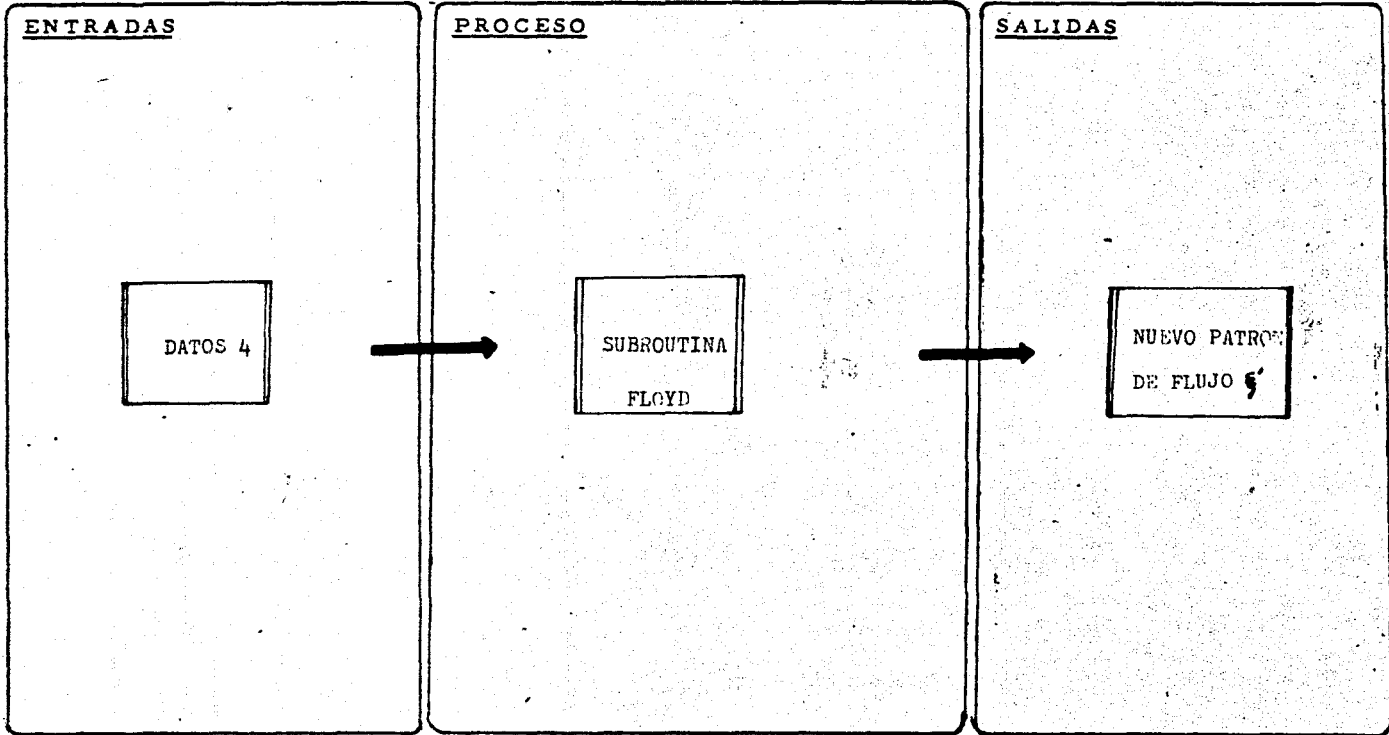
DATOS 4

- 1) Vector de capacidades asociadas a cada arco de la red  $G^u(\xi)$ .
- 2) Vector de flujo asociado a cada arco de la red G.
- 3) Vector indicador del ciclo negativo obtenido en la rutina Cost.

IDENTIFICACION

HOJA PARA DESCRIPCION  
FUNCIONAL DE PROGRAMAS

SISTEMA \_\_\_\_\_  
PROGRAMA SUBROUTINA FLOYD  
FUNCION GENERAL REALIZA EL CAMBIO DE FLUJO EN EL CIRCUITO NEG



OBSERVACIONES: \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

## SUBROUTINE ESCRIB

(paso 5 del algoritmo)

Esta es la rutina de escritura del proceso, en la cual vamos a ir escribiendo los resultados de cada iteración, i.e. vamos a escribir cada patrón de flujo que se va obteniendo para minimizar el costo.

Después de escribir los resultados obtenidos en cada iteración, regresamos con el nuevo patrón de flujo a la subrutina Crea( paso 2 del algoritmo).



## PARAMETROS QUE OPERAN DENTRO DE LA SUBROUTINE FORY

Los parámetros que se utilizan dentro de esta subrutina son los siguientes:

## a) Las variables:

$n$  = número de nodos de la red

$m$  = número de arcos de la red

b) Los arreglos de datos de dimensión  $(1 \times m)$  :

MARCA.- Arreglo que guarda los arcos en los que se envían unidades de flujo.

NODOI.- Arreglo que guarda el conjunto de nodos iniciales de cada arco de la red.

NODOF.- Arreglo que guarda el conjunto de nodos finales de cada arco de la red.

ICAP.- Arreglo que guarda las capacidades asociadas a cada arco de la red.

IFLUJ.- Arreglo que guarda el flujo asociado a cada arco de la red.

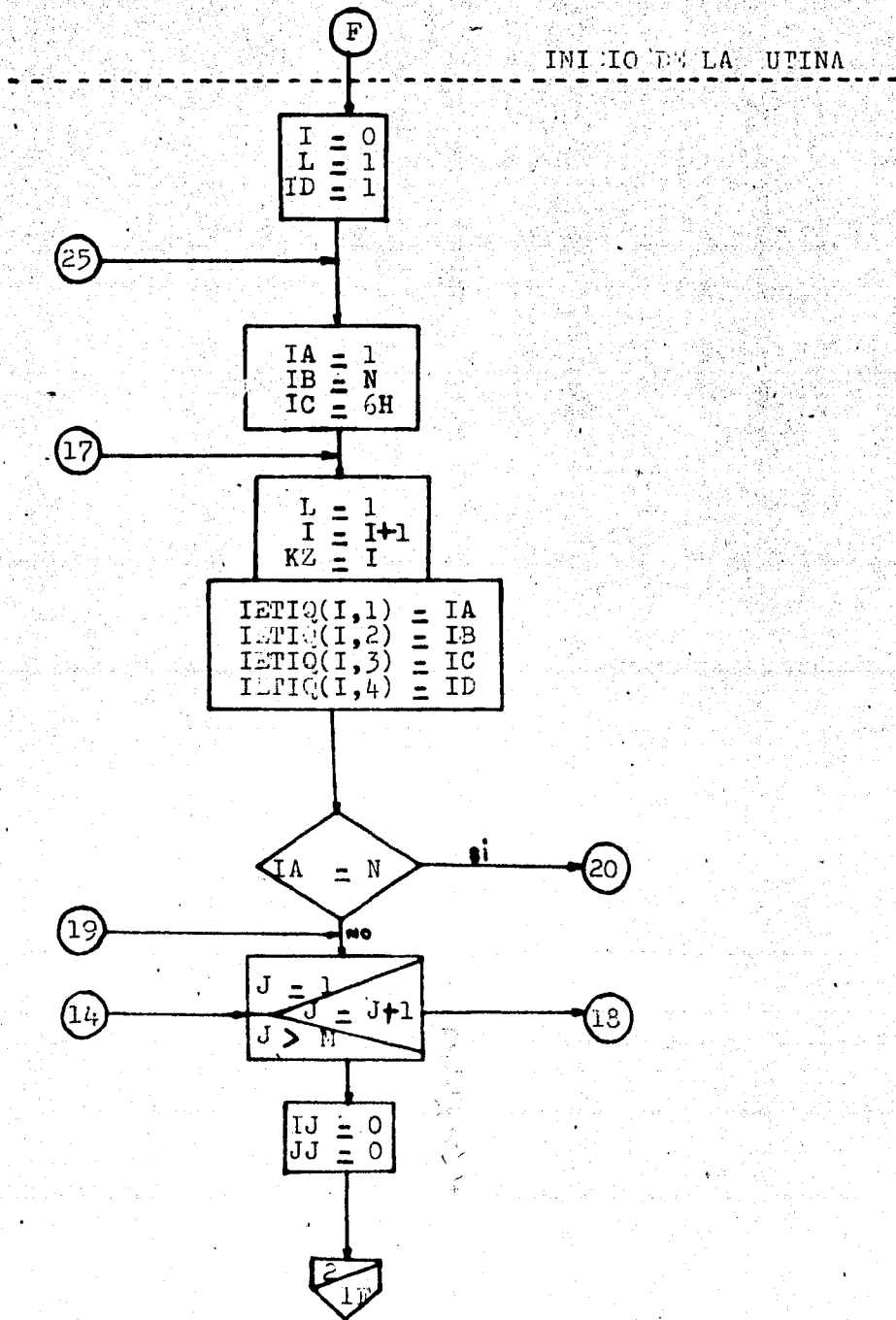
## c) Los arreglos que se generan dentro de esta rutina:

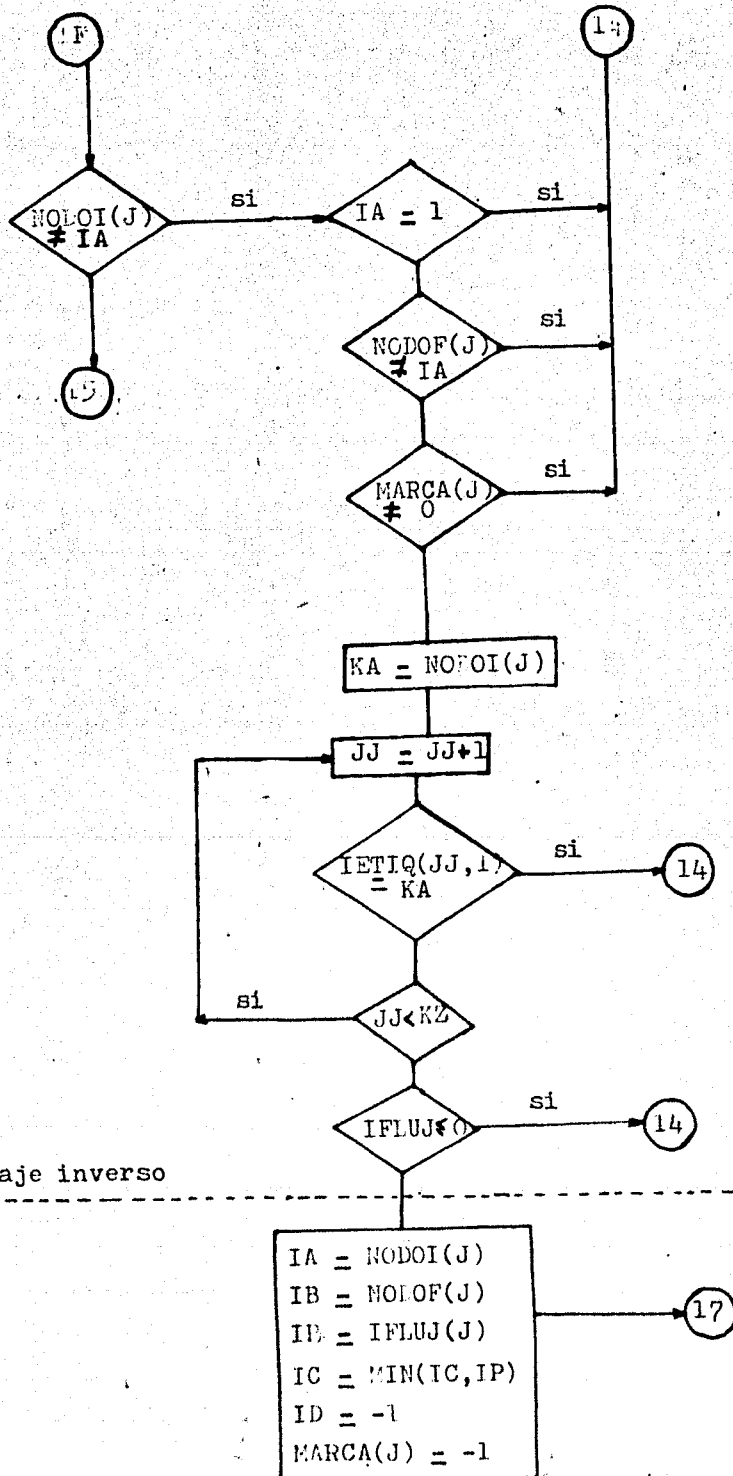
IETIQ.- Arreglo de dimensión  $(n \times 4)$  que guarda el conjunto de nodos marcados (proceso de Ford-Fulkerson) con su etiqueta correspondiente.

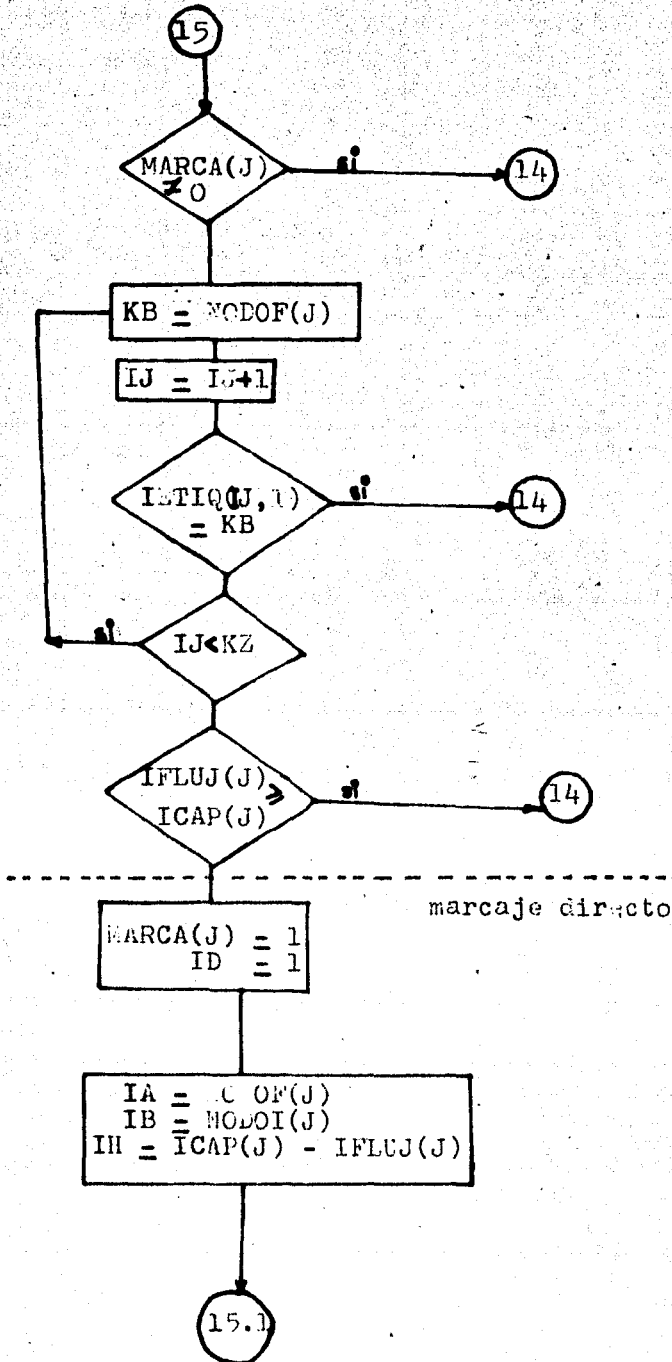
MARCAT.- Arreglo de dimensión  $(1 \times m)$  que guarda el conjunto de arcos que forman la cadena aumenta-flujos (cadena en la que se realiza el cambio de flujo).

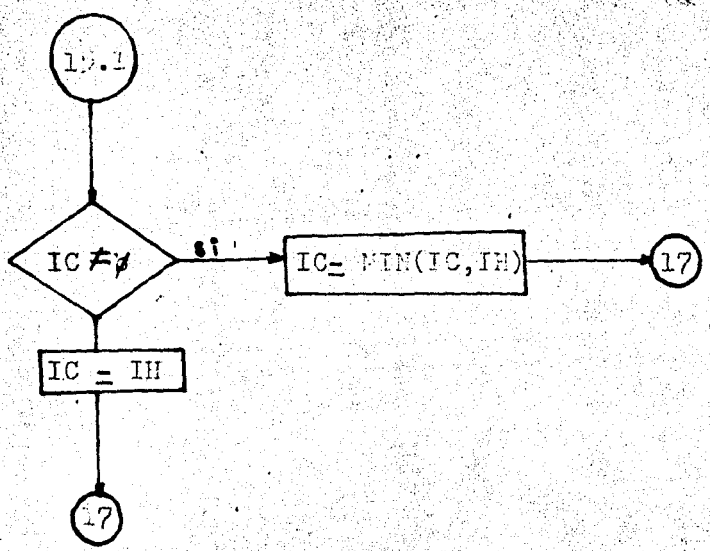
Y la variable  $kz1$  que es el número de nodos etiquetados.

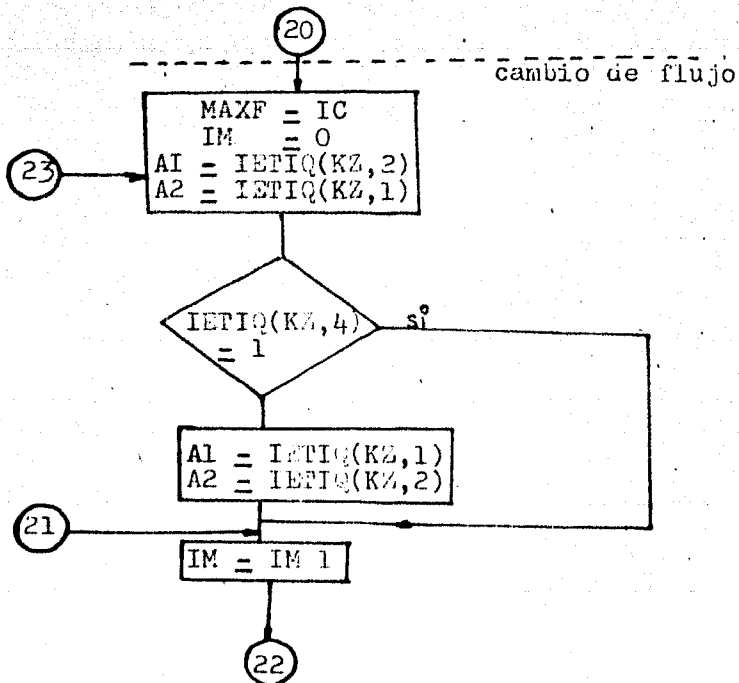
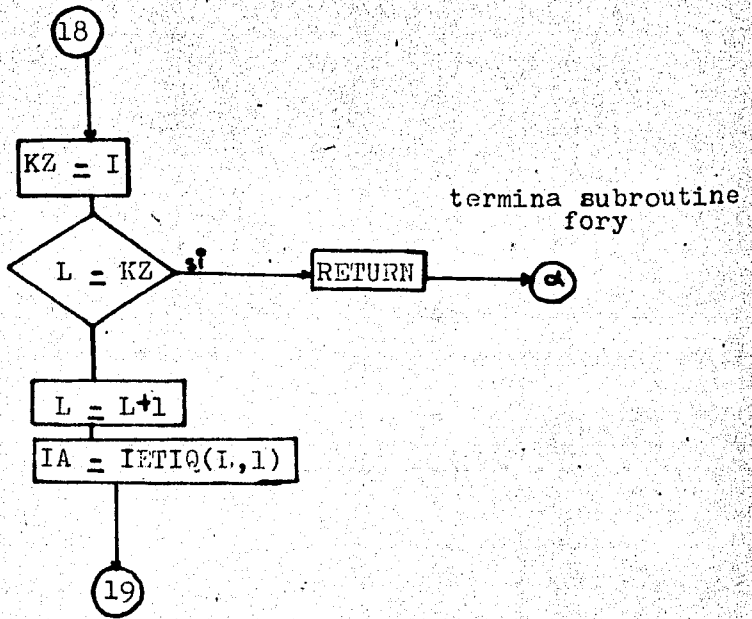
DIAGRAMA DE FLUJO DE LA SUBROUTINE FORY

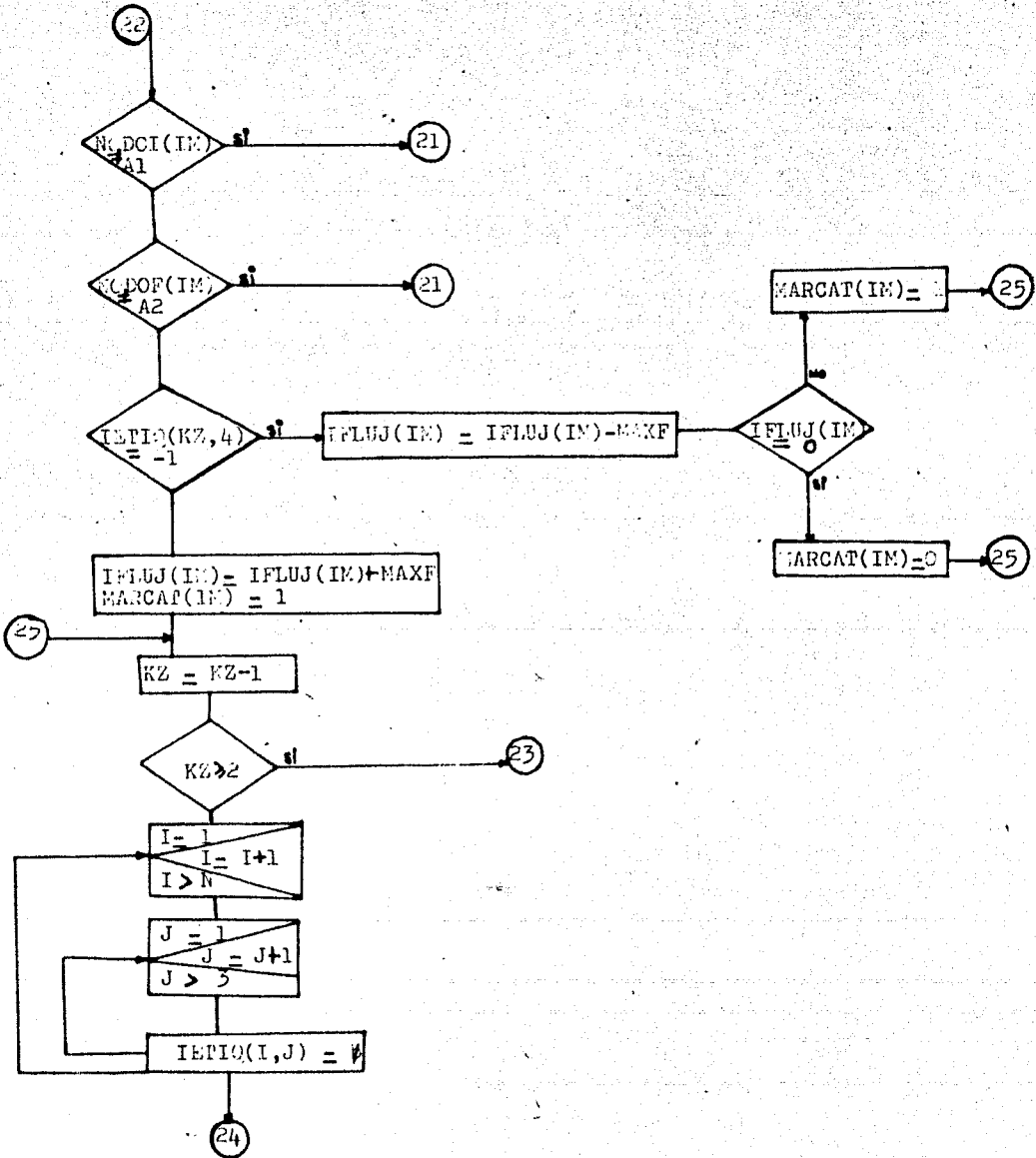


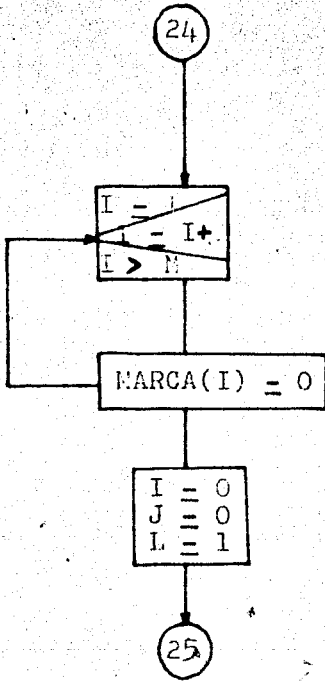














## PARAMETROS QUE OPERAN DENTRO DE LA SUBROUTINE CREA

Los parámetros que se utilizan dentro de esta subrutina son los siguientes:

## a) Las variables

n = (idem)

m = (idem)

ml = 2m es el máximo número de arcos que puede llegar a tener la nueva red generada  $G^u(\xi)$ .

KZ .- Número de arcos que tiene la nueva red  $G^u(\xi)$

(se obtiene su valor dentro de la rutina)  $KZ \leq 2m$

## b) Los arreglos de datos de dimensión (lxm):

MARCA.- (idem)

NODOI.- (idem)

NODOF.- (idem)

ICAP .- (idem)

COSTO.- Arreglo que guarda el costo asociado a cada arco:

IFLUJ.- (idem)

## c) Los arreglos que se generan dentro de esta rutina:

NODOI2.- Arreglo que guarda el conjunto de nodos iniciales de la nueva red  $G^u(\xi)$ , dimensión = 1 x ml.

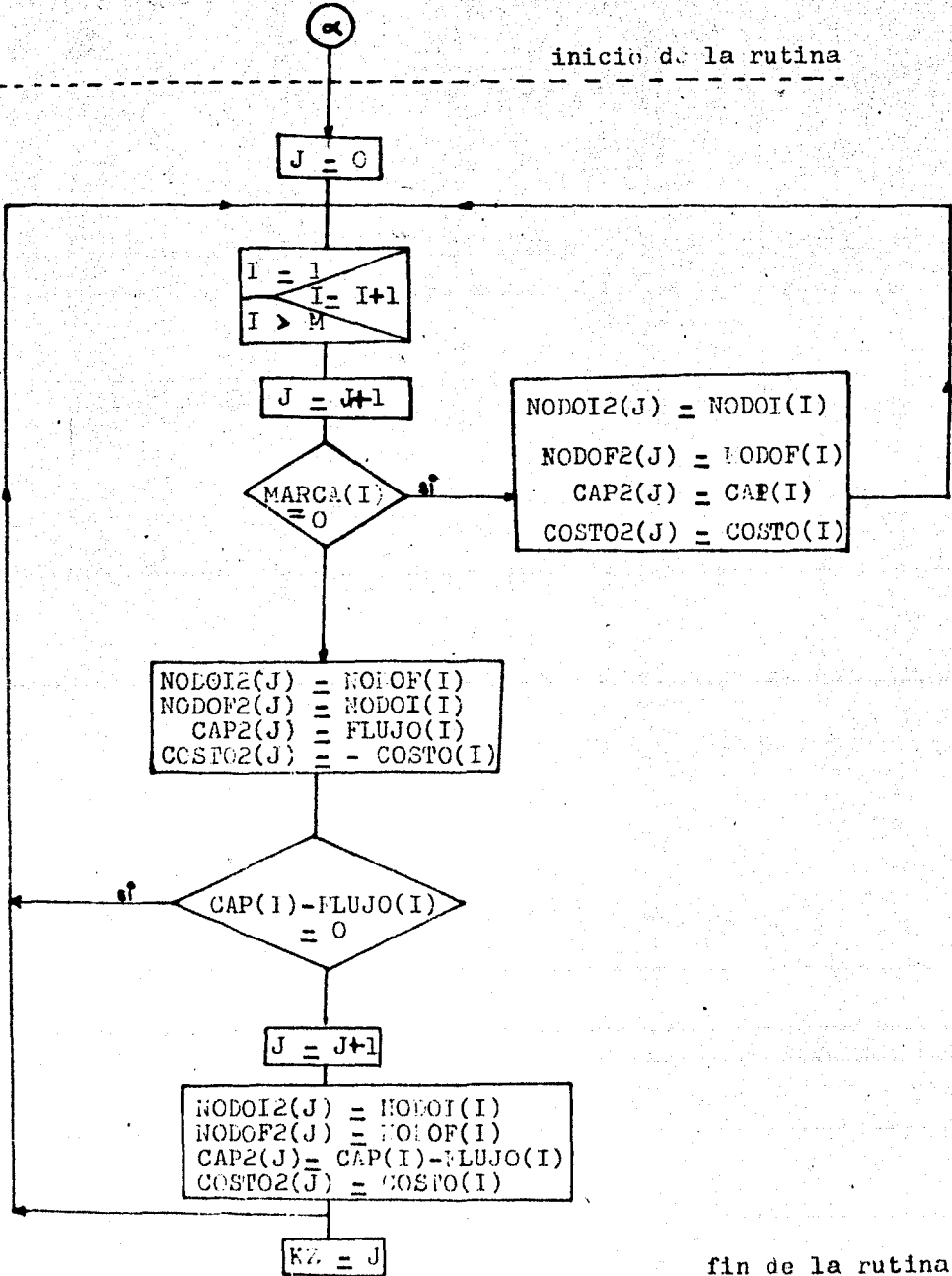
NODOF2.- Arreglo que guarda el conjunto de nodos finales de la nueva red  $G(\xi)$ , dimensión = 1 x ml.

CAP2.- Arreglo que guarda las capacidades asociadas a cada arco de la nueva red  $G^u(\xi)$ , dimensión = 1 x ml.

COSTC2.- Arreglo que guarda los costos asociados a cada ar-  
co de  $G^u(\frac{2}{3})$ , dimensión = 1 x ml.

DIAGRAMA DE FLUJO DE LA SUBROUTINA CREA

inicio de la rutina



fin de la rutina

## PARAMETROS QUE OPERAN DENTRO DE LA SUBROUTINE COST

Los parámetros que operan dentro de esta subroutine son los siguientes:

## a) Las variables:

n.- (idem)

m.- (idem)

ml.- (idem)

KZ.- (idem)

ISW.- variable que va a indicar si existe un circuito o no al salir de esta subroutine.

IR.- Variable que indica el lugar dentro de la matriz de caminos mínimos, en el cual encontraremos el nodo en el que se inicia y termina un circuito negativo.

Las variables ISW, IR se obtienen ambas dentro de esta rutina.

## b) Los arreglos de datos de dimensión (1x1) :

NODOI2.- (idem)

NODOF2.- (idem)

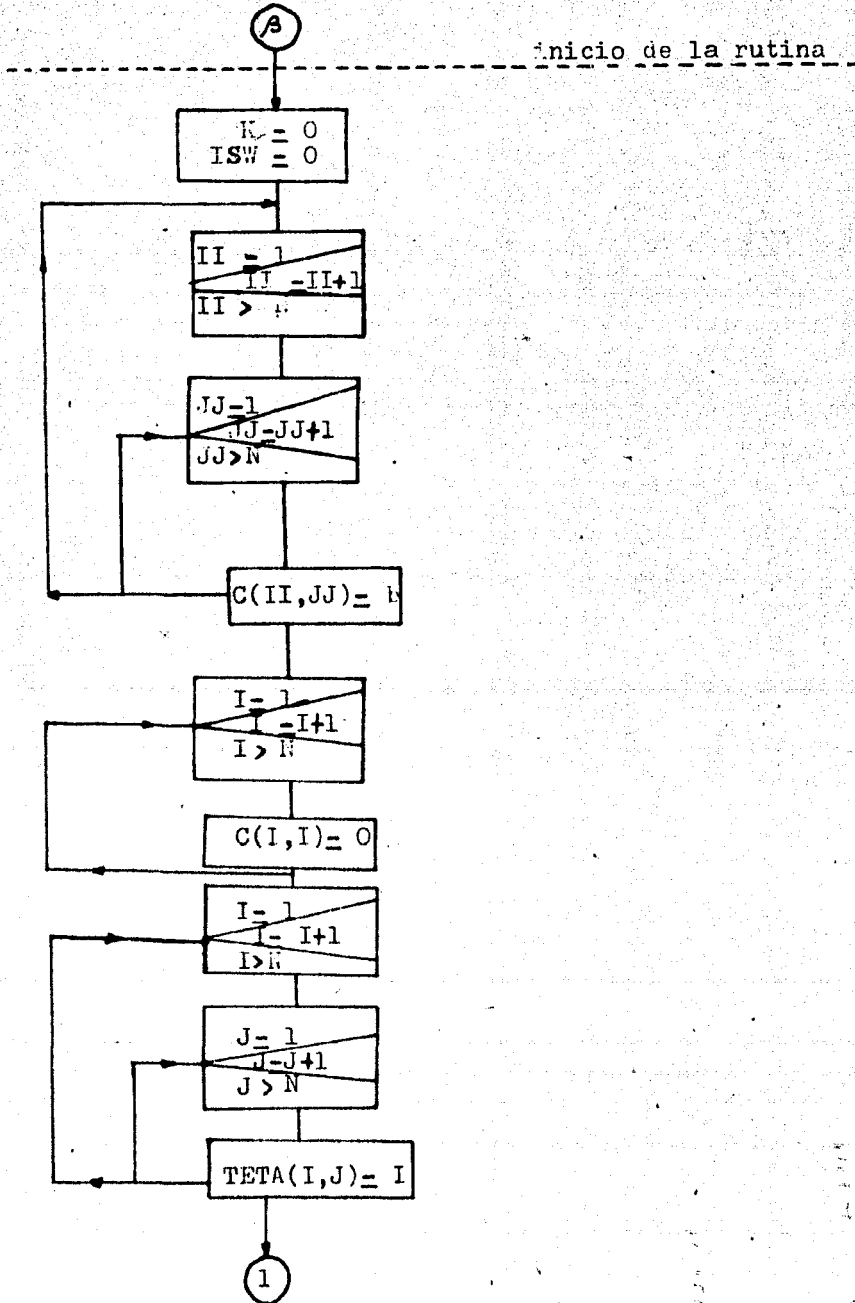
COSTO2.- (idem)

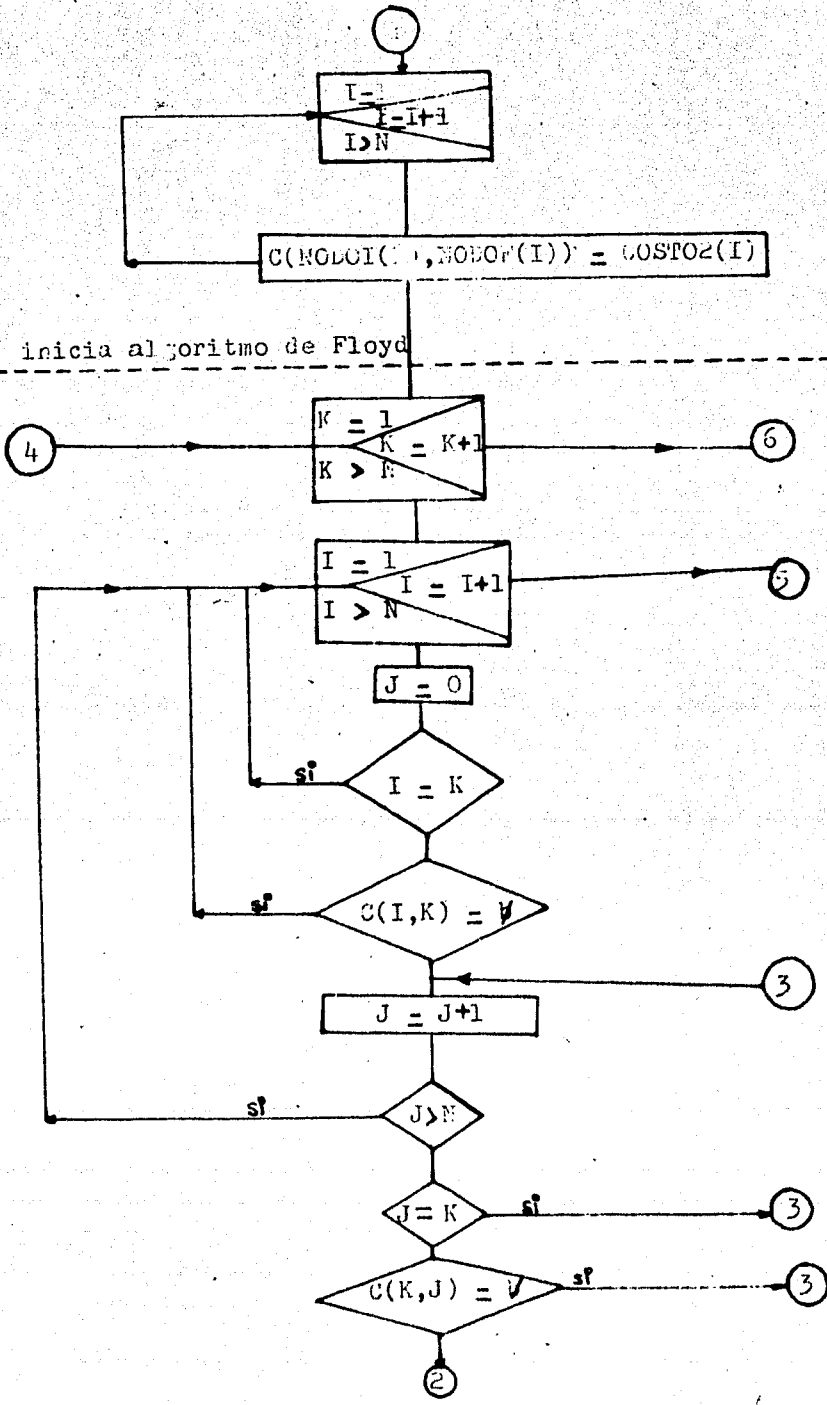
## c) Los arreglos de dimensión (nxn) que se generan dentro de esta rutina:

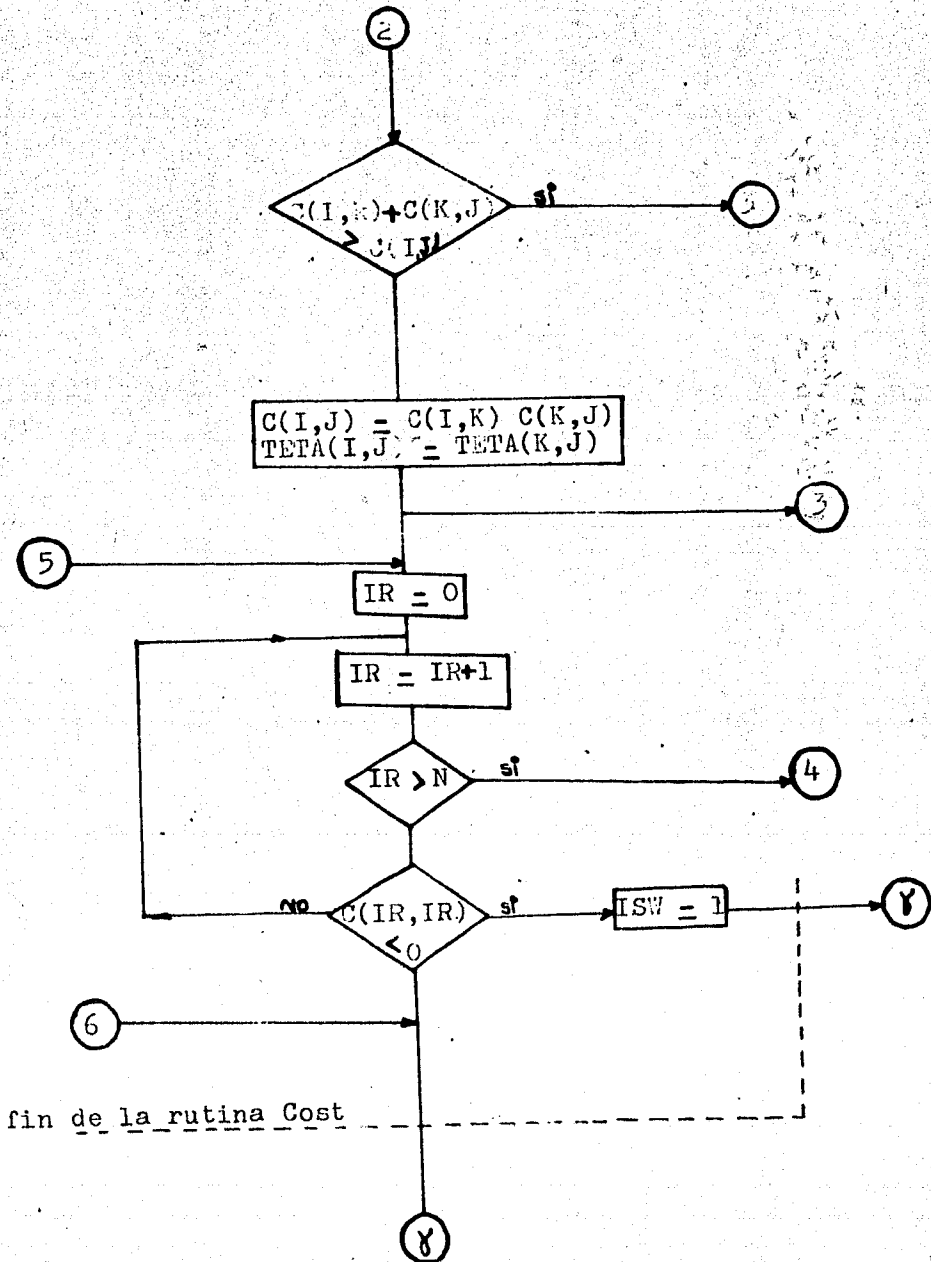
C .- Matriz de costos de la red  $G^u(\xi)$ .

TETA.- Matriz de caminos mínimos.

## DIAGRAMA DE FLUJO DE LA SUBROUTINA COST







## PARAMETROS QUE OPERAN DENTRO DE LA SUBROUTINE FLOYD

Los parámetros que operan dentro de esta subrutina son los siguientes:

## a) Las variables:

n = (idem)

m = (idem)

ml = (idem)

IR = (idem)

KZ = (idem)

JG.- Número de nodos que intervienen en un circuito negativo, el valor de esta variable se obtiene dentro de esta rutina.

IFL.- Cantidad de flujo que se cambia dentro del circuito negativo.

ICOST.- Costo de transportación del patrón de flujo.

IVALOR.- Valor del flujo.

## b) Los arreglos de datos con dimensión (1xm)

NOEOI.- (idem)

NOEOF.- (idem)

ICAP.- (idem)

COSTO.- (idem)

IFLUJ.- (idem)

## b') Los arreglos de datos de dimensión (1 x ml)

NOEOI2.- (idem)

NOEOF2.- (idem)



CAP2 .- (idem)

COSECO2.- (idem)

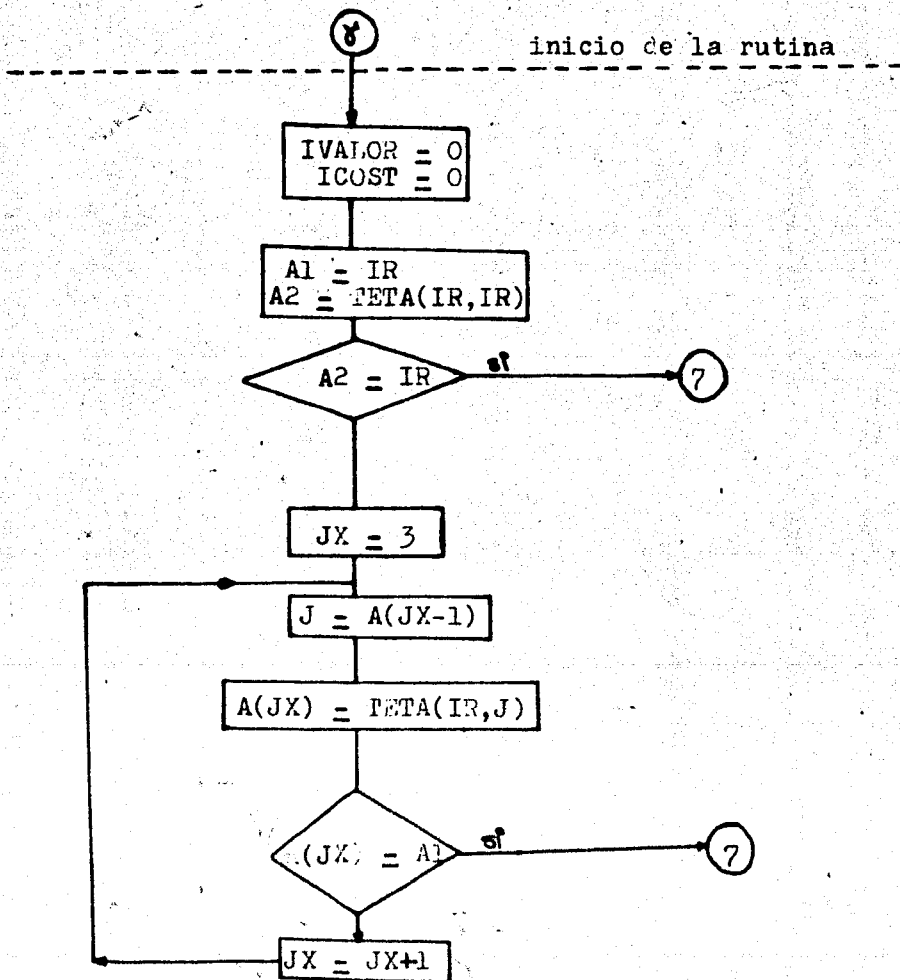
c) Las matrices C y TETA de dimensión (n x n).

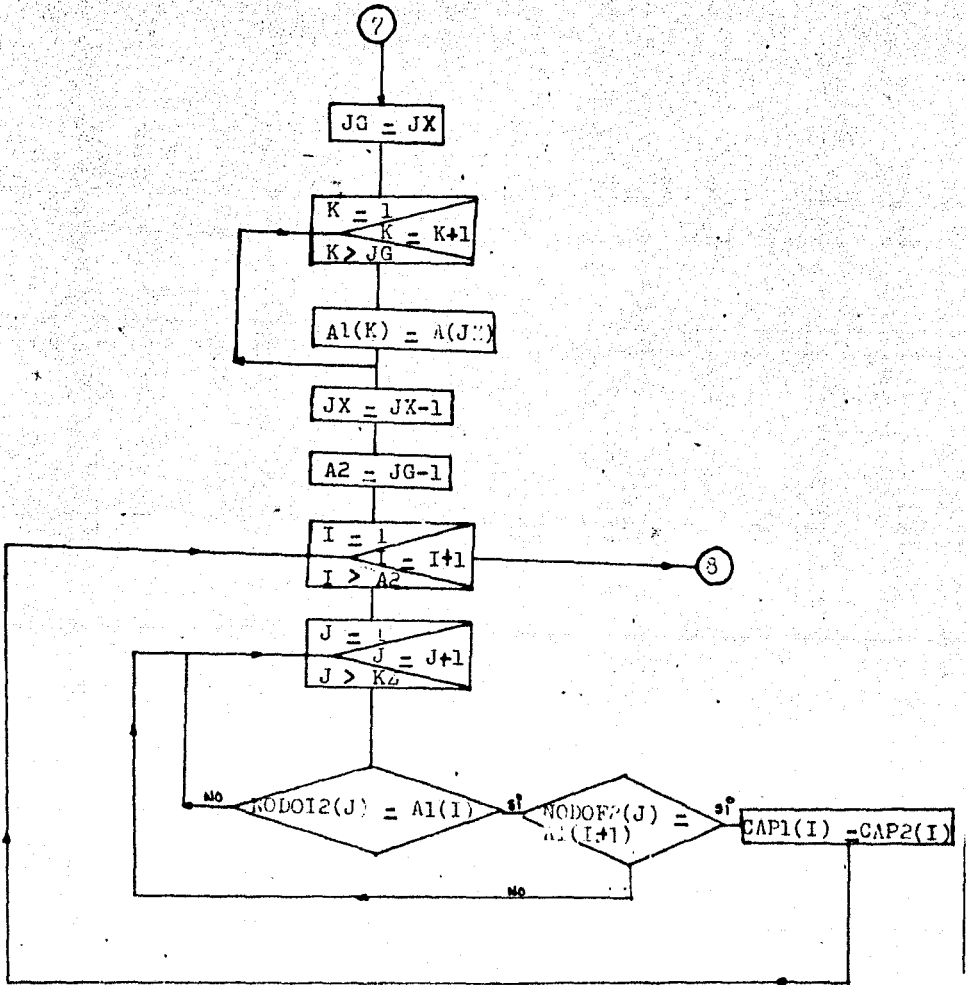
d) Los arreglos que se generan dentro de esta rutina.

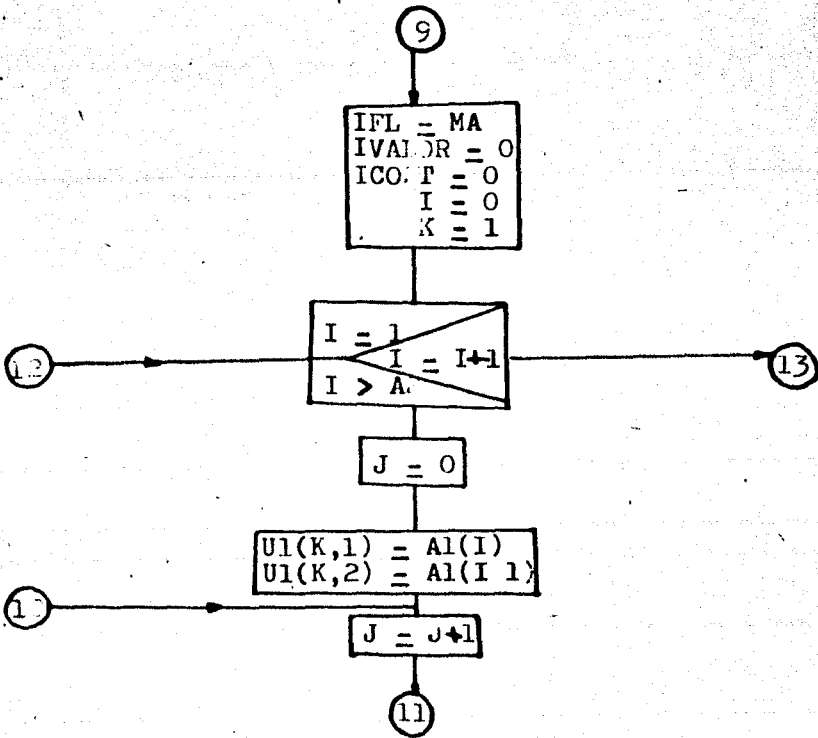
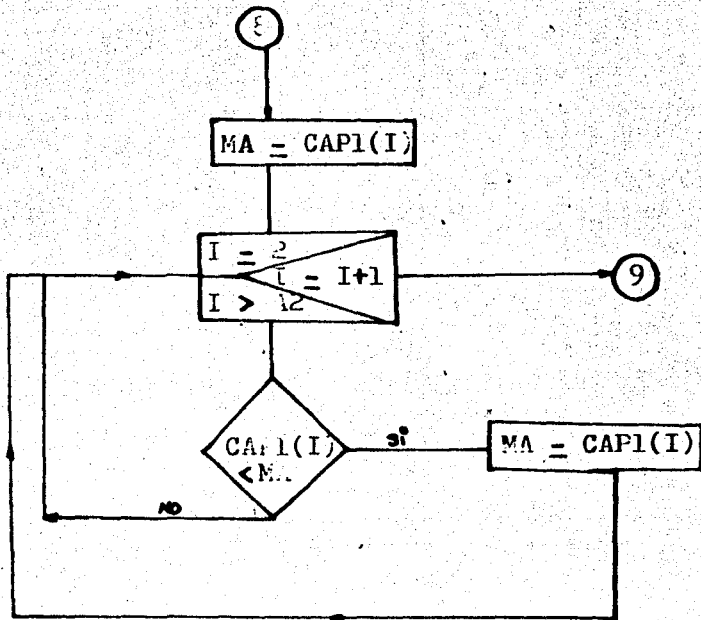
A, A1, U, U1 .- Vectores por medio de los cuales se encuentra y se guarda un ciclo negativo.

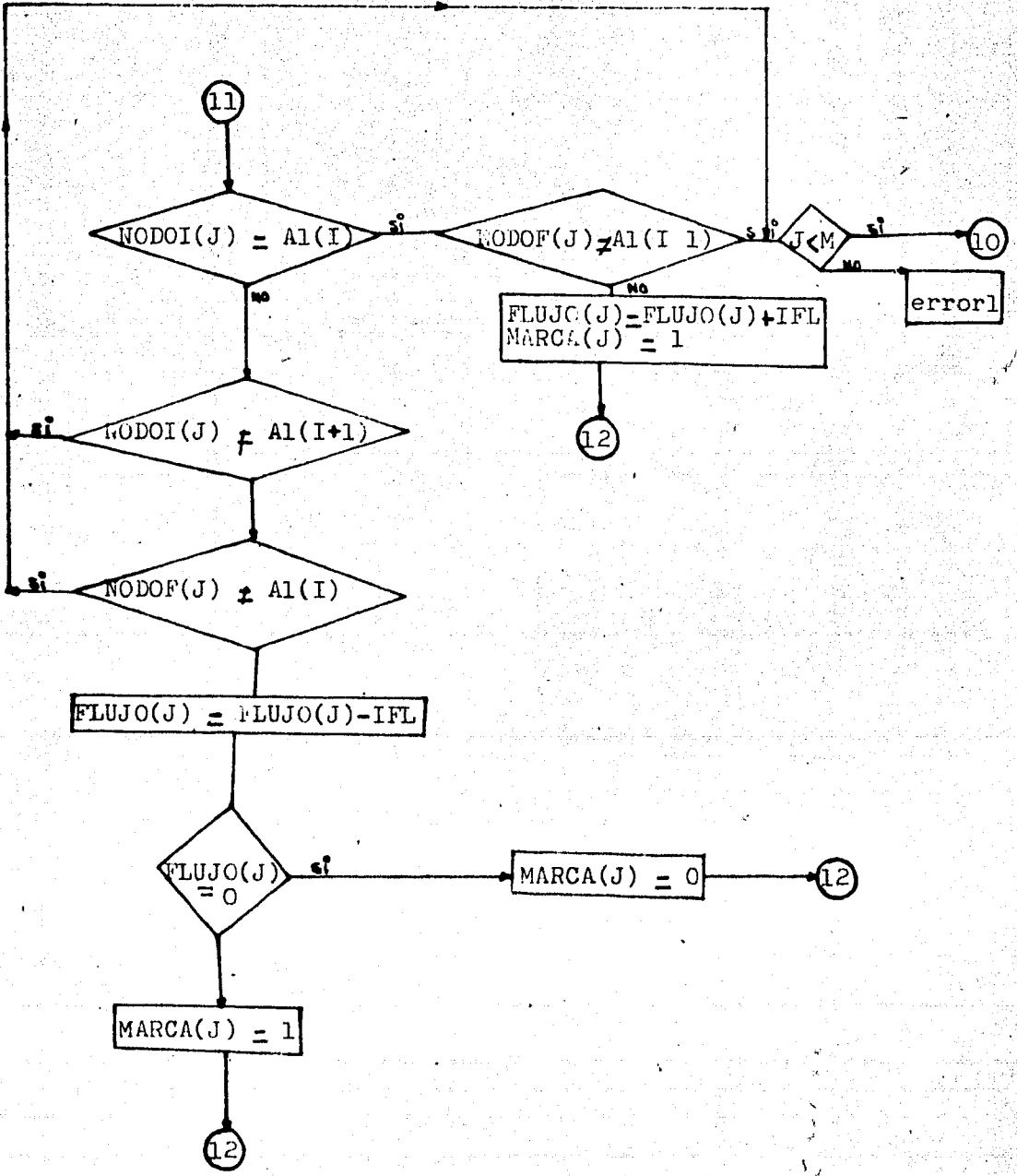
CAP1.- Vector que guarda la capacidad de los arcos contenidos en un circuito negativo.

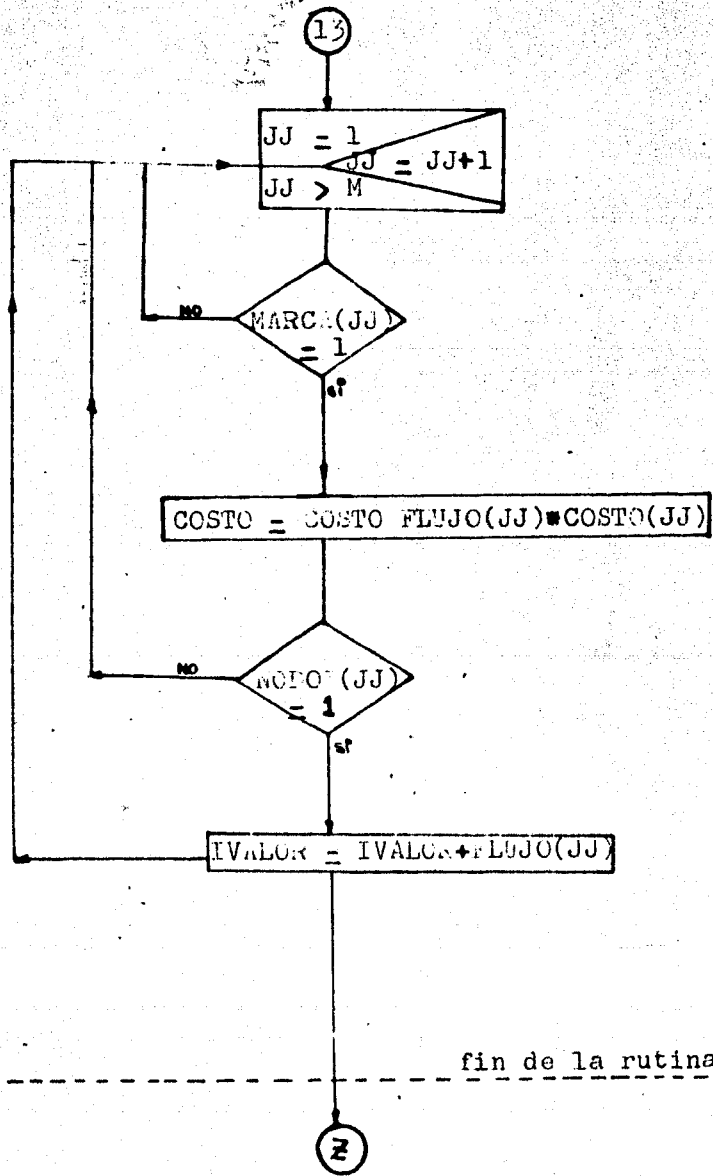
## DIAGRAMA DE FLUJO DE LA SUBROUTINA FLOYD











## PARAMETROS QUE OPERAN EN LA SUBROUTINE ESCRIB

Los parámetros para esta rutina son:

Al.- Arreglo que contiene los nodos que estan dentro de un circuito negativo.

IFL.- Cantidad de flujo que se cambia dentro del circuito negativo.

n .- Número de nodos de la red original

C .- Matriz de Costos

IVALOR.- Valor del flujo Máximo

JG .- Indice que indica la dimensión del arreglo Al

IR .- Lugar de la Matriz C en el cual existe un costo negativo, i.e.  $C(IR,IR) < 0$ .

