



UNIVERSIDAD NACIONAL AUTÓNOMA DE  
MÉXICO

ESCUELA NACIONAL DE ESTUDIOS  
PROFESIONALES ACATLÁN

SISTEMA COMPUTACIONAL PARA LA OBTENCIÓN  
DEL FLUJO MÁXIMO A COSTO MÍNIMO EN UNA RED,  
UTILIZANDO EL MÉTODO DE LAS DESVIACIONES.

TESIS PRESENTADA EN

LA DIVISIÓN DE MATEMÁTICAS E INGENIERÍA

COMO ASPIRANTE AL GRADO DE

LICENCIADO EN MATEMÁTICAS APLICADAS Y COMPUTACIÓN

*Incluye un Diskett de 3 1/2*  
POR

VERÓNICA ALEGRÍA RUIZ

ASESORADA POR

MAT. HÉCTOR ARGUELLES TEJEDA.

MARZO 1996.

TESIS CON  
FALLA DE ORIGEN

TESIS CON  
FALLA DE ORIGEN



Universidad Nacional  
Autónoma de México



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

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

Todo el material contenido en esta tesis 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.

---

## AGRADECIMIENTOS

A Dios por darme fe y esperanza,  
y me permitió vivir y crecer en su palabra,  
por la familia que me dio y por la vida  
que hoy puedo disfrutar con mis seres  
queridos ya que teniendo vida, esperanza  
y a él, que más puedo yo pedir.

A mis padres, por brindarme su cariño  
y comprensión, y brindarme la mejor  
de la herencia que un padre puede  
dar un hijo que es la educación.

A mis hermanos, Saúl, Román y Ulises por compartir  
mis triunfos y derrotas, por estar conmigo y por esos  
momentos maravillosos de mi infancia.

A mis amigos, los cuales siempre me brindaron su amistad y  
comprendieron que en mí existe un ser humano el cual puede tener errores, pero a  
pesar de todo siempre estuvieron conmigo y me exhortaron a dar mi mejor esfuerzo.  
Y sobre todo a Chontal, Aidelé, Veró, Saúl, José Manuel y Elío.

A todos mis maestros por sembrar en mí  
la semilla del conocimiento, todos ellos que  
compartieron su experiencias y conocimiento,  
especialmente a Héctor Argüelles T. y Efraim  
Meza quienes dejaron una huella profunda en mí.

---

---

## AGRADECIMIENTOS

A Dios por darme fe y esperanza,  
y me permitió vivir y creer en su palabra,  
por la familia que me dio y por la vida  
que hoy puedo disfrutar con mis seres  
queridos ya que teniendo vida, esperanza  
y a él, que más puedo yo pedir.

A mis padres, por brindarme su cariño  
y comprensión, y brindarme la mejor  
de la herencia que un padre puede  
dar un hijo que es la educación.

A mis hermanos, Saúl, Román y Ulises por compartir  
mis triunfos y derrotas, por estar conmigo y por esos  
momentos maravillosos de mi infancia.

A mis amigos, los cuales siempre me brindaron su amistad y  
comprendieron que en mí existe un ser humano el cual puede tener errores, pero a  
pesar de todo siempre estuvieron conmigo y me exhortaron a dar mi mejor esfuerzo,  
Y sobre todo a Chontal, Aidaé, Veró, Saúl, José Manuel y Elío.

A todos mis maestros por sembrar en mí  
la semilla del conocimiento, todos ellos que  
compartieron su experiencias y conocimiento,  
especialmente a Héctor Arquielles T. y Efraín  
Meza quienes dejaron una huella profunda en mí.

---

**OBJETIVO:**

El objetivo de este trabajo, es el proporcionar una herramienta computacional para obtener el flujo máximo y el flujo máximo a costo mínimo en una red. Este último utilizando el método de las desviaciones. Ya que este método tiende a ser muy laborioso por el número de cálculos involucrados, se presenta un sistema de cómputo el cual utiliza la metodología de las desviaciones; para que así el alumno tenga apoyo computacional didáctico en la materia de Optimización Combinatoria sobre el análisis de redes.

---

ÍNDICE

INTRODUCCIÓN	vii
<b><i>I. RESEÑA HISTÓRICA Y CONCEPTOS BÁSICOS</i></b>	<b>9</b>
I.1 RESEÑA HISTÓRICA	11
I.2 ELEMENTOS DE TEORÍA DE GRÁFICAS	13
I.3 ELEMENTOS DE GRÁFICAS DIRIGIDAS	15
I.4 REDES ESPECIALES	21
<b><i>II. FLUJO EN REDES</i></b>	<b>25</b>
II.1 DEFINICIÓN	27
II.2 FLUJO MÁXIMO EN UNA RED	27
II.3 MÉTODOS PARA ENCONTRAR EL FLUJO MÁXIMO EN UNA RED	29
II.4 APLICACIONES	34

---

**III. PROBLEMA DE FLUJO MÁXIMO A COSTO MÍNIMO**  
**EN UNA RED** 37

III.1 DEFINICIÓN DEL PROBLEMA 39

III.2 DIFERENTES MÉTODOS DE SOLUCIÓN 44

III.3 ALGORITMO PARA ENCONTRAR LOS  
POTENCIALES 46

III.4 MÉTODO DE LAS DESVIACIONES 46

**IV. OPERACIÓN DEL SISTEMA DE COMPUTO** 51

IV.1 REPRESENTACIÓN DE RESULTADOS 53

IV.2 LISTADO DEL PROGRAMA 54

IV.3 MANUAL DE OPERACIÓN 82

IV.3.1 OBJETIVO DEL PAQUETE. 82

IV.3.2 PASCAL 83

IV.3.3 TURBO PASCAL 84

IV.3.4 REQUERIMIENTOS DE HADWARE 85

IV.3.5 COMO ENTRAR AL PAQUETE. 85

IV.3.6 FORMA DE UTILIZAR EL MENÚ DE

AYUDA 86

---

IV.3.7 FORMA DE PRESENTACIÓN DE RESULTADOS DE EL FLUJO MÁXIMO A COSTO MÍNIMO POR EL MÉTODO DE DESVIACIONES.	87
IV.3.8 FORMA DE UTILIZAR EL MENÚ DE EJEMPLO	89
IV.3.9 FORMA DE UTILIZAR EL MENÚ DE CAPTURA	91
IV.3.10 FORMA DE UTILIZAR EL MENÚ DE MÉTODOS	93
IV.3.11 COMO SALIR DEL PAQUETE	94
IV.4 UTILIZACIÓN DEL SISTEMA	95
<b>V. COMENTARIOS Y CONCLUSIONES</b>	131
GLOSARIO	135
ANEXO A	137
ANEXO B	139
ANEXO C	142
ANEXO D	143
BIBLIOGRAFÍA	149

---

## INTRODUCCIÓN

A través del tiempo el hombre ha utilizado modelos, ya sea gráficos, iconicos, analógicos, físicos o simbólicos para resolver los problemas de su entorno.

Como se sabe la Investigación de Operaciones surge de la necesidad de la administración de herramientas para la solución de problemas en esta área. Dentro de la clasificación de la Investigación de Operaciones se tiene a la Programación Lineal, Programación no Lineal, Programación Entera, etcétera. En este trabajo se ha tomado un caso especial que se da en la Programación Entera, el cual es el relacionado a una red de transporte de la que se desea obtener el flujo máximo a un costo mínimo, esta red tiene un número de vértices finitos y capacidades finitas.

Existen diferentes métodos que se han realizado para dar solución a este problema, ya que puede solucionarse con el método simplex, pero en la década de los 60's se creo un método desarrollado por Ford y Fulkerson en 1962, llamado método de las *desviaciones o fuera de ajuste*; dicho algoritmo construye un flujo, mandando un flujo a los arcos donde el costo es mínimo o reacomodando este, si el flujo existente es mayor a cero, terminando en un número finito de pasos. Además, en otros métodos tenemos que empezar con un flujo cero y se va incrementando unidad por unidad para obtener el máximo flujo, también este método permite capacidades mínimas y máximas en el flujo, como también costos negativos que podrían considerarse como ganancias y en otros algoritmos no permite mas que valores enteros positivos.

Por ser un algoritmo primal dual puede empezarse con una solución la cual no sea optima ni para el primal o el dual, o para ambos. Esto suministra una medida de flexibilidad no disponible con otro algoritmo.

Este método se basa en clasificar los arcos en conformables y no conformables, un arco conformable se define como un arco el cual no excede las capacidades mínimas y máximas de flujo establecidas.

Este método es fácil de entender en clase, pero cuando se tienen redes pequeñas, es decir, de 3 a 5 vértices, porque después el número de calculos se complican si se resuelve manualmente porque es un método muy laborioso pero muy interesante, la complejidad del método hace que este se vuelva confuso y que muchos estudiantes no lo conozcan y si lo conocen no lo utilizan; además, si no se va graficando la solución correspondiente no se ve claramente que tipo de etiquetación se efectúa. Lo anterior da como consecuencia no tener una visualización general de los resultados que se van obteniendo a lo largo de la aplicación de este método.

Por eso nace la inquietud de desarrollar un software, el cual dé solución a las redes que tengan las características para darle la solución por el método de desviaciones, ya que en trabajos previos se ha expuesto este método, pero no existe un sistema el cual nos presente la graficación de la etiquetación.

Este paquete funciona mediante menús, el cual permite al usuario tener un menú de la terminología mas importante de acuerdo a teoría de gráficas relacionada con redes y cuenta con un ejemplo donde se mostrara al usuario como funciona el programa. Este programa además de darnos el flujo máximo nos ira dando los circuitos de cada etapa hasta llegar al flujo máximo y se aplica también la metodología del método de las desviaciones.

Este método se puede aplicar tanto en ingeniería para los flujos en circuitos o flujos fluviales, etcétera; como en otras áreas como la de transporte o comunicaciones que es donde se aplica o donde se ha visto que sirve para algo. En administración también se puede utilizar ya que si se tienen un número de oficinas en el país y se desea mandar el máximo de información y que este pase por las oficinas que son necesarias (es decir, tener el mínimo costo).

**CAPITULO I.**

**RESEÑA HISTORICA Y CONCEPTOS**

**BÁSICOS**

## I. RESEÑA HISTÓRICA Y CONCEPTOS BÁSICOS

### I.1 RESEÑA HISTÓRICA

Dentro de la Investigación de Operaciones se tiene a la Programación Entera, en la cual se encuentra el análisis de redes. El método de las desviaciones es uno de los primeros algoritmos para resolver el problema de flujo máximo a costo mínimo en una red.

En 1954 H. W. Kuhn, resolvió el problema de asignación. Con ayuda de su algoritmo de flujo maximal, que generalizaba el algoritmo de acoplamiento maximal de Kuhn-Ergeváry, L. R. Ford y D. R. Fulkerson fueron capaces de adaptar el método húngaro al problema de transporte simple y al problema de transporte a capacidades. Después se extendió el método al problema de distribución (simple o a capacidades).

Cuando Datzing, Ford y Fulkerson extendieron directamente las ideas fundamentales del método húngaro al problema general de programación lineal, el método se convirtió en "Un algoritmo Primal-Dual para Programas lineales". Por esta razón los algoritmos de redes especiales se denominan frecuentemente métodos primal-dual.

Durante este tiempo, A. Orden había indicado cómo cambiar formalmente un problema de distribución en un problema de transporte a capacidades. Más recientemente, H. Wagner ha dado un método eficaz para cambiar un problema de transporte simple, y D. R. Fulkerson ha dado una síntesis de los resultados de estos dos autores. Pero estas transformaciones desde el punto de vista metodológico, conducen a cálculos considerablemente más largos que el algoritmo directo de Jewell.

Obviamente el problema de flujo con costo mínimo es un programa lineal y se puede resolver de varias formas. Una forma consiste en aplicar el algoritmo simplex primal común. Lo que se busca es la simplificación del método simplex que pueda aplicar directamente a la red sin la necesidad de una tabla simplex.

Finalmente, D. R. Fulkerson ha dado recientemente un método, bajo el nombre de "fuera de ajuste" (out of kilter) que generaliza el de Jewell y el suyo para hacerle aplicable al problema de transporte con mínimos.

También se ha de conceder crédito a J. B. Dennis, cuya tesis de Programación Matemática y Redes eléctricas en 1958 presentaba métodos de solución de redes eléctricas que tienen paralelos muy interesantes con el algoritmo fuera de ajuste (out of kilter).

En realidad, el estado de esta técnica en esa, época era tal que muchas personas empezaron a reconocer la importancia de utilizar relaciones de las separaciones complementarias para resolver problemas con condiciones iniciales arbitrarias, o hacer estudios paramétricos.

El algoritmo de desviaciones fue presentado por vez primera por Fulkerson (1961). Ford y Fulkerson (1962) describe el algoritmo en su libro clásico de *Flows in Networks*. Clasen (1968) da una implementación computacional de este algoritmo. Durbin y Kroehke (1967) y Phillips y Jensen (1974) proveen tutoriales.

Este algoritmo se experimento ampliamente durante los 60's y a principios de los 70's. El problema de flujo máximo a costo mínimo es un problema complejo, por el número de variables involucradas, es estático dado que no se tienen probabilidades asociadas. El algoritmo tiene características que lo hacen útil hoy en día. Entre otras ventajas están:

- Es simple de comprender y por eso es útil para el salón de clases, siempre que la red no sea muy grande.
- Toda la información es descrita por los parámetros de los arcos.
- El algoritmo puede iniciar con cualquier flujo que satisfaga la conservación de flujo. Así el algoritmo es particularmente útil para el análisis de sensibilidad cuando los arcos son cambiados.

Entre las desventajas del algoritmo están:

- Las soluciones no son necesariamente básicas. Esto puede resultar en una convergencia lenta al óptimo.
- Es muy exhaustivo por el análisis de los arcos.
- Los procedimientos de solución pueden requerir numerosos cruces a través de la larga lista, así el algoritmo no es probablemente computable para problemas largos.

Este algoritmo puede empezar con cualquier solución siempre que satisfaga la conservación del flujo, esta solución puede no ser óptima para el primal, ni para el dual.

I.2 ELEMENTOS DE TEORÍA DE GRÁFICAS

Una gráfica  $G=(V,A)$  consiste de un conjunto de objetos llamados vértices  $V$  y un conjunto de líneas  $A$ , relacionados por una función  $\phi$ .

$V = \{v_1, v_2, v_3, \dots, v_n\}$  Conjunto no-vacío de objetos llamados vértices de  $G$ .  
(También son conocidos por puntos ó nodos).

$A = \{e_1, e_2, e_3, \dots, e_n\}$  Es el conjunto de líneas de  $G$ , el cual puede ser igual al conjunto vacío. (También llamadas: arcos, ejes, aristas o segmentos)

$$\phi = A \rightarrow \{(v_i, v_j), v_i, v_j \in V\}$$

Esta relación es la que asocia a cada línea de  $G$ , un par de vértices de  $G$  (no necesariamente distintos) de tal forma que a cada línea  $e_k$  de  $G$  se le asocia un par de vértices  $v_i, v_j$ .

Dos vértices  $v_i$  y  $v_j$  que inciden en una línea en común se dice que son adyacentes, es decir, son vértices finales de la misma línea.

Los vértices finales de una línea, se dicen que son incidentes a esa línea y al contrario.

Dos líneas  $e_k$  y  $e_{k+1}$  que están asociadas con un mismo par de vértices se dice que son linealmente paralelas. (Fig. 1.1)

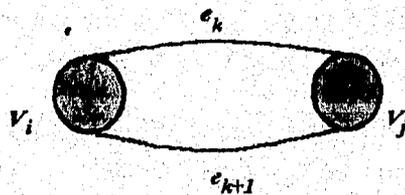


Fig. 1.1

Se llama bucle, rizo o loop de una grafo a un arco cuya extremidad inicial se confunde con la extremidad terminal, es decir, que del vértice  $v_i$  al vértice  $v_j$  existe una línea  $e_k$ , pero  $i=j$ , entonces el vértice extremo coincide con el inicial. (Fig. 1.2)

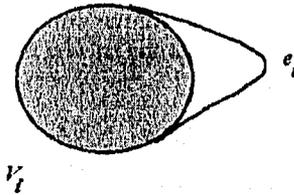


Fig. 1.2

Se le llama grafo o gráfica simple aquella gráfica que no tiene loops ni líneas paralelas, en caso de tener algún elemento de los anteriores se tiene una gráfica general.

Una secuencia de nodos  $v_0, v_1, \dots, v_n$ ,  $n \geq 1$ , es llamado un camino de  $v_0 - v_n$  camino de  $(v_{i-1}, v_i)$   $A, \forall i$  con  $i = \overline{1, n}$ .

El nodo  $v_0$  es llamado origen y el nodo  $v_n$  es llamado destino y los nodos  $v_1, v_2, \dots, v_{n-1}$  son nodos intermedios. Nosotros podemos representar un camino por secuencias de aristas  $\{e_1, e_2, \dots, e_n\}$ ; donde  $n$  es el número de aristas. Un camino es llamado ruta si no existen nodos repetidos.

Una cadena es una sucesión de aristas  $e_i$  tales que una de las extremidades de cada sucesión cadena pertenece a la arista anterior de la sucesión y la otra extremidad es la siguiente.

Un paseo es definido como una sucesión alternante de vértices y líneas de  $G$  donde  $V_1, V_2, \dots, V_n$  son vértices de  $G$ , y  $\{e_1, e_2, \dots, e_k\}$  son aristas de  $G$ , pero además cada arista  $e_k$  tiene extremos al vértice  $v_i$  que le precede. Por ejemplo  $v_1 e_1 v_2 e_3 v_6$  (Figura 1.3).

Una trayectoria simple es un paseo abierto, en el que ningún vértice aparece más de una vez. Un loop puede aparecer en un paseo, pero no en una trayectoria.

Un camino o trayectoria cerrada, es aquella donde los vértices no aparecen más de una vez; (excepto el vértice inicial y final), también es conocido como circuito, ciclo elemental, trayectoria circular o polígono, entonces, un circuito es una trayectoria cerrada sin intersección y esta trayectoria debe de constar de por lo menos de 3 elementos  $e_k$  los cuales están relacionados con elementos del conjunto de vértices  $V$ . Por ejemplo  $v_1 e_1 v_2 e_4 v_4 e_3 v_1$ , ver Fig. 1.4.

Una gráfica se dice que es acíclica si no contiene un ciclo.

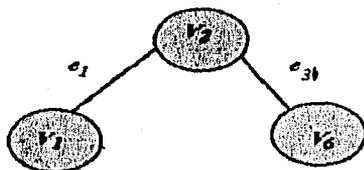


Fig. 1.3

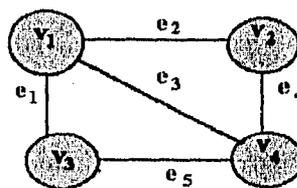


Fig. 1.4

### 1.3 ELEMENTOS DE GRÁFICAS DIRIGIDAS

Una generalización importante de teoría de gráficas, son las gráficas dirigidas.

Una gráfica dirigida ó digráfica  $D=(V,A)$  consiste en una serie finita  $V=\{1,\dots,m\}$  y una serie  $A = \{e_1, e_2, e_3, \dots, e_n\}$ , los elementos de  $V$  están ordenados en subseries de  $V$  de tamaño dos llamados arcos. En una digráfica  $(i,j)$  y  $(j,i)$  son diferentes elementos y nosotros podemos tener uno, ambos o ninguno de estos elementos. En la representación pictórica de una digráfica, las flechas son usadas para indicar un orden.

Muchas situaciones físicas requieren ser modeladas por digráficas, como cuestiones de redes donde es importante el sentido de las líneas.

Dentro de una red de transporte existen elementos de la serie  $V$ , en donde un  $v_i$  representa un centro de producción o fábrica, este elemento  $v_i$  esta unido por un elemento  $e_i$  de la serie  $A$ , a otro elemento  $v_j$  de  $V$ , en donde  $e_i$  representa una línea de comunicación, es mucho más fácil efectuar el análisis de la red cuando se tiene la representación pictórica, es decir la digráfica de la red.

Digráfica simétricas son aquellas en las cuales para cada arista del vértice  $v_i$  al vértice  $v_j$  existe otra del  $v_j$  al  $v_i$ , pero con diferente dirección.

Una red  $N$  es una digráfica  $D$  (la gráfica fundamental de  $N$ ) con dos distintas subseries de vértices,  $x$  e  $y$ , y una función no negativa  $c$  definida en estos arcos que están en la subserie de  $A$ ; las subseries  $x$  e  $y$  se proponen están separadas y diferentes.

Los vértices  $X$  son los orígenes y los  $Y$  son los destinos de  $N$ . La correspondencia de los centros de producción y mercados, respectivamente son los vértices, los vértices que no son ni el origen ni el destino son llamados vértices intermedios, la serie de estos vértices se denota por  $I$ . La función  $c$  es la función de capacidad de  $N$  y este valor en el arco es la capacidad de  $A$ . La capacidad de un arco puede pensarse como la representación del índice máximo de la mercancía que puede ser transportada a lo largo de ésta.

Una red dirigida o una gráfica lineal dirigida  $G=[V;A]$  consiste de una colección de elementos  $x,y$ , en  $V$ , juntos con una subserie  $A$  estos elementos ordenados en pares  $(x,y)$  de los elementos tomados de  $V$ . Esto se supone para todos los elementos de  $N$ . Los elementos de  $N$  son llamados de diferente forma; nodos, vértices o puntos. Los miembros de  $A$  son referidos como arcos uniones, ramas o aristas.

Llamaremos red de transporte  $(N,A)$  a un grafo orientado sin bucles en el que:

a) Asociamos con cada arco  $(v_i, v_j) \in A$

- una capacidad  $c(v_i, v_j) \Rightarrow c_{ij} \geq 0$  que, por ejemplo, fija el valor maximal del caudal de  $v_i$  hacia  $v_j$  en este arco;
- una "penalización"  $(v_i, v_j) \Rightarrow l_{ij} \geq 0$  que según los casos, puede interpretarse como el coste del caudal unitario en el arco  $(v_i, v_j)$  ó como la longitud del arco  $(v_i, v_j)$ , etcétera.

Sea  $V$  un conjunto finito cualesquiera de los elementos  $v_i$ ,  $A$  es el conjunto constituido por algunos pares ordenados  $e_{ij}=(v_i, v_j)$  con  $v_i, v_j \in V$ .

Llamaremos red de transporte (ó simplemente red)  $(V,A)$  a la red definida por los conjuntos  $V$  y  $A$ .

Los elementos  $v_i$  del conjunto  $V$  se llaman vértices o puntos de la red de transporte  $(V,A)$  y los elementos  $e_{ij}=(v_i, v_j)$  del conjunto  $A$ , comunicaciones de la red. Por convenio, diremos que la comunicación  $e_{ij}$  enlaza los puntos  $v_i$  y  $v_j$  comenzando en  $v_i$  y acabando en  $v_j$ . Por consiguiente, una red de transporte significa indicar el conjunto de puntos y las comunicaciones que los unen.

Una red de transporte cualquiera puede representarse gráficamente. Con este fin hace falta asociar cada punto de la red con un punto del plano, enlazando después, cada par de puntos  $(v_i, v_j)$ , que corresponden a la comunicación de la red mediante un segmento orientado que va de  $v_i$  a  $v_j$ . El sentido del segmento esta indicado por una flecha. Siendo  $(v_i, v_j) \neq (v_j, v_i)$ .

Una red de transporte es una gráfica sin bucles. La cual tiene asociado un número  $c(u_i)$ .

Existe uno y solo un vértice  $v_0$  llamado fuente de la red tal que existen 0 líneas que entran a este vértice. Existe uno y solo un vértice  $v_n$  llamado sumidero de la red, del que no salen ninguna línea.

Se dice que una función entera  $\varphi(u)$  definida sobre el conjunto de los arcos de  $v_i$  es el flujo para la red de transporte.

- $\varphi(u) \geq 0 \quad \forall u \in A$

Una gráfica se puede representar de forma pictórica, ya que es una de las más usadas ya que podemos visualizarla y darnos una idea de como esta compuesta, pero para el procesamiento por medio de computadoras es mejor la representación algebraica que se da con las matrices.

### MATRIZ DE INCIDENCIA

Sea  $G$  una gráfica con  $n$  vértices y  $m$  líneas sin loops. Se define una matriz  $A=[a_{ij}]$  de  $n \times e$  cuyos  $n$  renglones corresponden a los  $n$  vértices y las  $e$  columnas corresponden a las aristas.

Al elemento de la matriz se le será asignado (0 ó 1) de acuerdo a lo siguiente:

$a_{ij} = 1$       Si la  $i$ -ésima arista  $e_j$  es incidente con el  $i$ -ésimo vértice  $v_i$

$a_{ij} = 0$       En otro caso

A tal matriz  $A$  se le denomina la matriz vértice -renglón de incidencia o simplemente matriz de incidencia. la matriz  $A$  de una gráfica  $G$  algunas veces también se escribe como  $A(G)$ , como en la Figura 1.5.

La matriz de incidencia contiene solo dos elementos 0 y 1. De modo que es llamada una *matriz binaria* ó *(0-1)-matriz*.

La matriz de incidencia y la gráfica contienen la misma información. Son simplemente dos alternativas para su representación.

### CARACTERÍSTICAS

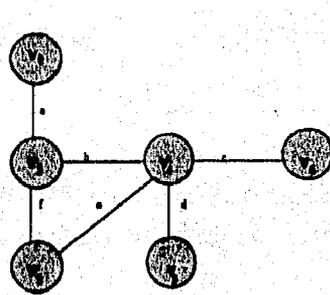
- Dado que toda arista es incidente en exactamente dos vértices, cada columna de  $A$  tiene exactamente dos unos.
- El número de unos en cada renglón es igual al grado de cada vértice.

- Un renglón con ceros, representa un vértice aislado.
- Aristas paralelas en una gráfica produce idénticas columnas en  $A$ .
- Si una gráfica  $G$  es desconectada y consiste en dos componentes  $g_1$  y  $g_2$  la matriz de incidencia  $A(G)$  de una gráfica  $G$  puede ser escrita de la forma de un bloque diagonal.

$$A(G) = \left[ \begin{array}{c|c} A(g_1) & 0 \\ \hline 0 & A(g_2) \end{array} \right]$$

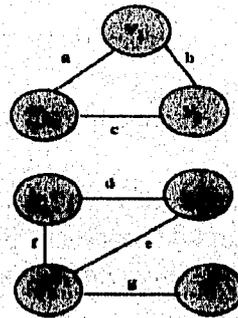
donde  $A(g_1)$  y  $A(g_2)$  son los componentes de la matriz de incidencia de  $g_1$  y  $g_2$ . Esta observación resulta de el hecho de que una arista que no este en  $g_1$  e incidente a los vértices de  $g_2$  y viceversa. Obviamente esta observación es valida para cualquier número de componentes.(Fig. 1.6)

- La permutación de 2 cualesquiera renglones o columnas en una matriz de incidencia simplemente corresponde a re etiquetar los vértices y aristas en la gráfica.



	a	b	c	d	e	f
$V_1$	1	0	0	0	0	0
$V_2$	1	1	0	0	0	1
$V_3$	0	0	0	0	1	1
$V_4$	0	1	1	1	1	0
$V_5$	0	0	0	1	0	0
$V_6$	0	0	1	0	0	0

Fig. 1.5



	a	b	c	d	e	f	g
1	1	1	0	0	0	0	0
2	0	1	1	0	0	0	0
3	1	0	1	0	0	0	0
4	0	0	0	1	0	1	0
5	0	0	0	1	1	0	0
6	0	0	0	0	1	1	1
7	0	0	0	0	0	0	1

Fig. 1.6

**MATRIZ DE ADYACENCIA**

Como una alternativa a la matriz de incidencia en algunas ocasiones es mas conveniente representar una gráfica por esta matriz de adyacencia o matriz de conexión. La matriz de adyacencia de una gráfica  $G$  con  $n$  vértices y sin líneas paralelas es una matriz binaria simétrica de  $n \times n$ ,  $X = \{x_{ij}\}$  definida sobre el anillo de los enteros tal que

$$\begin{aligned} x_{ij} &= 1 && \text{Si existe una línea entre el } i\text{-ésimo y el } j\text{-ésimo vértice} \\ x_{ij} &= 0 && \text{en otro caso} \end{aligned}$$

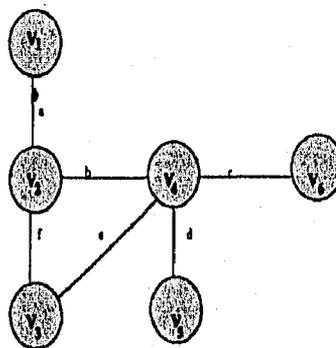
**CARACTERÍSTICAS**

- Las entradas a lo largo de la diagonal principal de  $X$  son todos ceros si y solo sí, la gráfica no contiene loops. Un loop en el  $i$ -ésimo vértice corresponde a  $x_{ii}=1$ . Fig. 1.7
- La definición de una matriz de adyacencia no prevé la existencia de aristas paralelas.
- Si la gráfica  $G$  no contiene loops el grado de los vértices es igual al numero de unos en la correspondiente columna o renglón de  $X$ .
- Una gráfica  $G$  es desconectada en dos componentes  $g_1$  y  $g_2$  si y solo si la matriz de adyacencia  $X(G)$  puede ser particionada de la siguiente forma.

$$X(G) = \left[ \begin{array}{c|c} X(g_1) & 0 \\ \hline 0 & X(g_2) \end{array} \right]$$

- donde  $X(g_1)$  es la matriz de adyacencia del componente  $g_1$  y  $X(g_2)$  es la del componente  $g_2$ , esta partición indica claramente que no existe una arista que pueda unir algún vértice en la subgráfica  $g_1$  con algún vértice de la subgráfica  $g_2$ .
- Dada cualquier matriz binaria simétrica cuadrada  $Q$  de orden  $n$ , se puede construir siempre una gráfica de  $n$  vértices (sin líneas paralelas) tal que  $Q$  es la matriz de adyacencia de  $G$ .
- Una gráfica completa presenta una matriz de adyacencia con ceros en la diagonal principal y unos en el resto.

Una gráfica fuertemente conexa tiene una matriz de adyacencia unitaria.



	$V_1$	$V_2$	$V_3$	$V_4$	$V_5$	$V_6$
$V_1$	0	1	0	0	0	0
$V_2$	1	0	1	1	0	0
$V_3$	0	1	0	1	0	0
$V_4$	0	1	1	0	1	1
$V_5$	0	0	0	1	0	0
$V_6$	0	0	0	1	0	0

Fig. 1.7

Una red tiene las siguiente características:

1. La red es simétrica.
2. No admite loops.
3. No existe líneas paralelas ( es decir, solo hay un canal de comunicación).

## I.3 REDES ESPECIALES

Existen diferentes problemas que se han modelado por medio de redes, continuación se presenta una lista de sistemas en los que se han aplicado:

- 1.- Sistemas de Distribución-Producción.
- 2.- Sistemas Logísticos Militares.
- 3.- Sistemas de Trafico Urbano.
- 4.- Sistemas de Ferrocarriles.
- 5.- Sistemas de Comunicaciones.
- 6.- Sistemas de Redes de Pipas.
- 7.- Sistemas de Ubicación de Sitios.
- 8.- Sistemas de Unión de archivos.
- 9.- Sistemas de Inventarios.
- 10.- Redes electrónicas.

Como se ha dicho anteriormente una red esta compuesta por 2 tipos de series: arcos y nodos. Los arcos por comodidad significan transporte, y los nodos pueden ser interpretados como locaciones o terminales conectadas por los arcos y sirven para que el transporte sea asociado con los arcos. Por lo tanto los arcos pueden representar calles y carreteras en una red de transportación urbana, pipas en una red de distribución de agua, líneas telefónicas en una red de comunicaciones.

Se dice que se tiene una red circulatoria, si hacemos ir un arco o arista del vértice final al vértice origen.

Existen muchos casos especiales de problemas de redes los cuales han sido estudiados y se mencionaran a continuación dando el nombre de estos e indicando la especialización.

1. *Problema de transbordamiento sin capacidad.* Este problema es una especialización del problema en redes en el cual la capacidad máxima del arco es infinita, es decir  $b_{ij} = \infty$  para todo arco que pertenece a la red.

2. *Problema de transporte a capacidades.* Es donde se tiene la restricción de que la cantidad transportada por el camino  $(i,j)$  no debe exceder un valor  $b_{ij}$ , llamado capacidad del camino,  $x_{ij} \leq b_{ij}$ .

Entonces este problema se denomina problema de transporte "a capacidades".

El camino óptimo de  $i$  a  $j$  no es directo, ya que existen varios itinerarios entre los nodos orígenes y destinos que incluyen cruces o puntos de distribución. La red de transporte, en vez de ser una red simple, toma la forma de la figura 1.8. Este problema de distribución (simple o a capacidades).

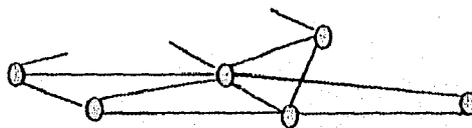


Fig. 1.8

La red es arbitraria, pero la cantidad de mercancía que pasa por un vértice no puede exceder a un valor  $b_j$ , llamado capacidad del vértice. Y la cantidad transportada por un camino  $[i,j]$  ha de ser al menos igual a un valor  $a_j$  y como máximo igual a la capacidad del vértice, es decir:

$$a_j \leq x_j \leq b_j.$$

Este es el problema de transporte (o de distribución) con mínimos.

La mercancía transportada experimenta una transformación durante el camino una unidad que pasa desde  $i$  hasta  $j$  se convierte en  $a_j$  al llegar a  $j$ ; este es el problema de transporte (ó de distribución) con ganancias. Por ejemplo, en el caso de una red simple, el problema es

$$\text{Mínimizar } z = \sum_i \sum_j C_{ij} x_{ij}$$

sujeto a

$$\sum_{j=1}^n x_{ij} = a_i \quad \cdot \quad i = 1, \dots, m$$

$$\sum_{j=1}^m a_{ij} x_{ij} = b_j \quad j = 1, \dots, n$$

$$x_{ij} \geq 0$$

El nombre del problema de transporte viene del significado económico que se le atribuye con más frecuencia a que disponemos en cada uno de los  $i$  orígenes de una cantidad  $a_i$  de un producto que deseamos llevar a  $n$  destinos  $j$ , de manera que se satisfagan las demandas  $b_j$ , el costo unitario del transporte del producto desde el origen  $i$  al destino  $j$  se denota por  $c_{ij}$ , y es directamente proporcional a la cantidad  $x_{ij}$  transportada de  $i$  a  $j$  y se de que el costo total de transporte sea mínimo sin exceder las disponibilidades. La expresión analítica dada más arriba, supone que es posible llevar este producto desde todo origen a un destino  $j$ ; evidentemente podríamos eliminar esta hipótesis impidiendo la condición  $x_{ij} = 0$  para algunos pares  $(i,j)$ .

En lugar de transportar una mercancía sola, se transporta simultáneamente varias por la misma red. Más de una de estas mercancías puede estar disponible en el mismo origen o demanda en el mismo destino para otra. Este es el problema de transporte (o distribución) de múltiples mercancías.

El problema de flujo con costo mínimo se puede originar, por ejemplo, en una red logística en la que hombres y materia mueven varios puntos de la tierra. También puede estar asociado con el movimiento de locomotoras entre puntos en una red de ferrocarriles, con el fin de proporcionar a los trenes energía al menor costo de viaje. Problemas de flujo con costo mínimo en redes ocurren en el análisis y diseño de sistemas de comunicación, sistemas de oleoductos, programación de tanques de guerra, y en muchas otras áreas más.

Para el problema de costo mínimo que se presentará en el capítulo III se considerará la red de transporte (o de distribución) con mínimos.

3.- *Problema de asignación.* Este es un caso especial de un problema de transportación en el cual los requerimientos son igual a 1 para todo  $i=1, \dots, n$ .

4.- *Problema de flujo máximo.* Dada una red con capacidad de arcos  $b_j$  para  $j=1, \dots, m$  el problema de flujo máximo es encontrar el flujo máximo continuo del nodo  $s$  al nodo  $t$ , donde  $s$  y  $t$  pueden ser cualquier par de nodos dados. Más adelante se verá algunos de los diferentes algoritmos que se han hecho para resolver este problema, donde por lo regular el flujo inicial de la red es cero.

5.- *Problema de la ruta más corta.* Dada una red cuyos arcos tienen asignada cantidades  $c_{ij} \geq 0$  para cada vértice  $i,j$  el problema consiste en el camino más corto en la red, entre los nodos  $s$  y  $t$ , donde  $s$  y  $t$  pueden ser un par de nodos cualesquiera, este problema es un caso especial del problema de transbordamiento sin capacidad en el cual se toman  $r_s = 1$  y  $r_t = -1$  y todos los otros requerimientos diferentes a estos vértices son ceros.

**CAPITULO II.**

**FLUJO EN REDES**

## II. FLUJOS EN REDES

### II.1 DEFINICIÓN

Dada una red  $G = [V; A]$  supóngase que a cada arco  $(i, j) \in A$  tiene asociado un número entero no negativo  $c(i, j)$ . Nosotros llamaremos a  $c(i, j)$  la capacidad del arco  $(i, j)$  esto puede considerarse intuitivamente como representación de la cantidad máxima de algunas mercancías que pueden llegar a  $j$ , a partir de  $i$  en una unidad de tiempo. La función  $c$  de  $A$  entera no negativa es la función de capacidad.

Llamaremos flujo estacionario aritmético o más simplemente flujo (aritmético), en la red a una función  $X$  con valores enteros positivos, que hace correspondencia a todo arco  $(i, j) \in A$  un valor  $x(i, j) = x_{ij}$  que satisface las relaciones:

$$0 \leq x(i, j) \leq c_{ij} \quad (i, j) \in A$$

$$x(i, j) - x(k, i) = 0 \quad \forall (k, i) \in A \text{ con } k \in V \quad \forall i \in V$$

$x(i, j) = x_{ij}$  puede considerarse como el valor de una circulación real en el arco  $(i, j)$  con el sentido de  $i$  hacia  $j$ , llamaremos  $x_{ij}$  al valor del flujo aritmético en el arco  $(i, j)$  ó simplemente al flujo en el arco  $(i, j)$ .

Un conjunto de flujos en los arcos de una red define un flujo aritmético en una red únicamente cuando se verifican las relaciones anteriores. La primera relación expresa que el valor del flujo en todo arco es lo sumo igual a la capacidades del arco. La segunda expresa que en todo punto de paso el flujo aritmético "que entra en dicho punto" es igual al flujo aritmético que "sale de dicho punto".

### II.2. FLUJO MÁXIMO EN UNA RED

El problema de la existencia de un flujo no se plantea. En una red cualesquiera *siempre existe* al menos un flujo (el flujo nulo).

Sea una red de transporte cualesquiera  $(V, A)$  y una función de las capacidades  $d(c_{ij}) \geq 0$  definida en el conjunto  $A$ .

Fijemos un cierto par de puntos de la red  $(V,A)$ . Numeremos los puntos del conjunto  $P$  de manera que el primer punto del par fijado (entrada) esta, designado por  $p_1$  y el segundo (salida) por  $p_n$ , donde  $n$  es el número total de los vértices en la red.

Consideremos todos los flujos posibles sobre las trayectorias de  $v_1$  y  $v_n$  compatibles con la función de capacidad antes mencionada. El problema consiste en buscar el conjunto de trayectorias entre ellas con flujo máximo.

A continuación se presentan algunas propiedades simples de los flujos en redes.

Dado que es más cómodo exponer los razonamientos que se presentan en función de flujos aritméticos. Es por esto, que en lo siguiente entenderemos por flujo una función  $x(c_{ij})$ ,  $c_{ij} \in A'$ , que verifica las condiciones

$$x(c_{ij}) = -x(c_{ji}) \quad (1)$$

$$x(c_{ij}) = 0 \text{ para } i \neq 1, N$$

$$c_{ij} \in A_i \quad (2)$$

$$x(c_{ij}) \leq d(c_{ij}),$$

$$c_{ij} \in A' \quad (3)$$

#### Cortadura (Separación del nodo $n$ del nodo 1)

Sea  $W$  cualquier conjunto de nodos en la red tal que  $W$  contiene el nodo inicial (ó 1) pero no al nodo  $n$  o final. Sea  $\bar{W} = V - W$  donde  $V$  es el conjunto de nodos. Entonces  $(W, \bar{W}) \equiv \{(i, j) : i \in W, j \in \bar{W}\}$  se llama una cortadura que separa al nodo  $n$  del nodo 1.

#### Capacidad del corte

Sea  $(W, \bar{W})$  cualquier cortadura en una red  $G$ . Entonces  $c(W, \bar{W}) = \sum_{(i,j) \in (W, \bar{W})} b_{ij}$  se llama la capacidad de la cortadura.

El número de cortaduras posibles en una red de  $n$  nodos, es igual a todas las combinaciones en donde en el conjunto  $W$  se encuentra el vértice origen y en el conjunto  $\bar{W}$  se encuentra el vértice final, es igual a  $2^{n-2}$ .

La cortadura de valor mínimo es el Minimizar(  $c(W, \bar{W})$ ), la cual es igual al flujo máximo.

### II.3 MÉTODOS PARA ENCONTRAR EL FLUJO MÁXIMO EN UNA RED

La idea básica del algoritmo de flujo máximo es encontrar una ruta que vaya del origen al destino y que la capacidad de cada arco en la ruta sea positivo y no sobrepase la capacidad del arco.

Para el problema de flujo máximo, las capacidades mínimas son iguales a cero al igual que los costos por cada unidad del flujo. En este problema al igual que en el problema de flujo máximo a costo mínimo debe existir solo un nodo origen y un nodo destino o final.

La representación matemática del problema es:

$$\text{Minimizar } z = \sum_{i=1}^n \sum_{j=1}^n C_{ij} x_{ij}$$

s.a

$$\sum_{k=1}^n x_{ik} - \sum_{k=1}^n x_{ki} = 0 \quad \forall i \in V$$

$$a_{ij} \leq x_{ij} \leq b_{ij}$$

donde  $a_{ij}$  es la capacidad mínima, y como se dijo anteriormente esta será 0 y  $b_{ij}$  es la capacidad máxima.

Básicamente la mayoría de los autores utilizan el método de Ford y Fulkerson para dar solución al flujo máximo, aunque se presentará también otros donde a diferencia del último no se escoge un flujo cualquiera sino el máximo que se puede mandar de un arco a otro.

ALGORITMO NO.1

El algoritmo que va a continuación, permite, partiendo de un flujo inicial cualesquiera, determinar un flujo de valor igual a la capacidad de un conjunto disjunto, por consiguiente un flujo maximal. El procedimiento es iterativo; a cada iteración, un método inductivo de etiquetación de los vértices permite aumentar el flujo en una cantidad discreta.(Anexo B.1)

El algoritmo se puede traducir indiferentemente como el grafo de la red o sobre la matriz de las capacidades.

El marcado de los vértices de la red se traduce en la matriz por el marcado de una fila. Aquí se enunciará "sobre la red".

Paso 1. Atribuir al vértice inicial la marca  $(-, \infty)$ .

Paso 2. Considerar un vértice ya marcado cualesquiera,  $p_i$ , de marca  $(p_i, \alpha_i)$ . Se marcará  $(p_i, \alpha_j) = \text{minimizar}(\alpha_i, \gamma_{ij})$  a todo vértice  $p_j$  aun no marcados y tal que:  $(p_i, p_j)$  pertenecen a  $V$  y  $\gamma_{ij} > 0$ . Si  $p_j$  se puede marcar de varias maneras, adoptar una cualquiera de estas.

Continuar el marcado hasta que  $p_n$  este marcado o hasta que no sea posible seguir marcando ningún vértice.

Paso 3. Si  $p_n$  esta marcado, modificar el flujo de la manera siguiente:

Determinar, partiendo de  $p_n$ , la secuencia

$$\sigma (p_1, p_2, \dots, p_i, p_j, \dots, p_n)$$

en la que cada vértice es la marca del vértice siguiente. Aumentar en  $\alpha_n$  el flujo algebraico en toda la ruta de esta secuencia, en el sentido de  $p_1$  hacia  $p_n$ , lo que da como consecuencia disminuir en  $\alpha_n$ , las capacidades residuales  $\gamma_{ij}$  de estas rutas.

Entonces, repetir los pasos 1, 2 y 3.

Paso 4. Si no se puede marcar ningún vértice más y  $p_n$  no ha sido marcado, entonces se dice que el flujo es máximo (o que se tiene un flujo maximal).

Observemos que, si  $(\bar{\gamma}_{ij})$  es la matriz de las capacidades residuales en el momento en que se detiene el procedimiento, los flujos de  $p_i$  a  $p_j$  vienen dados por la capacidad del arco menos la cantidad mandada por el arco  $ij$ .

**Demostración.-** La operación 3 transforma un flujo en otro flujo. En efecto, el flujo algebraico en toda ruta  $(p_i, p_j)$  de la cadena  $\sigma$  en una cantidad inferior o igual a  $\gamma_{ij}$  en el sentido de  $p_i$  hacia  $p_j$ , es decir de  $p_i$  hacia  $p_n$ , y disminuye en la misma cantidad en el sentido contrario; ningún otro flujo de la red resulta modificado. Por lo tanto, siguen verificándose las relaciones de flujo aritmético.

Cuando se para el procedimiento, existe un conjunto  $A$  de vértices no marcados;  $A$  define un conjunto disjunto. Por medio del teorema siguiente:

Si existe un flujo algebraico  $f$  y un conjunto disjunto  $\Delta = (\bar{A}, A)$  tales que el valor del flujo sea igual a la capacidad de  $\Delta$ , el flujo es maximal y el conjunto disjunto es un corte de capacidad minimal (o un conjunto disjunto minimal igual a la reunión de un corte minimal y de los arcos de capacidad nula).

**Demostración.-** Toda unidad de flujo que vaya de  $p_0$  a  $p_n$  ocupa al menos un arco de un conjunto disjunto cualesquiera. luego cualesquiera que sea el valor de  $\varphi$  de un flujo  $f$  y la capacidad  $k(\Delta)$  de un conjunto disjunto de  $\Delta$  tenemos:

$$\varphi \leq k(\Delta)$$

De manera formal, se puede escribir;

$$\begin{aligned} \varphi &= f(p_0, P) = f(p_0, P) + f(\bar{A} - \{p_0\}, P), \text{ ya que } f(p_i, P) = 0 \text{ si } i \neq 0, n \\ &= f(\bar{A}, P) = f(\bar{A}, A) \leq k(\bar{A}, A), \text{ ya que } f(p_i, p_j) \leq k_{ij} \end{aligned}$$

Resulta que si existe un par  $(f_0, \Delta_0)$  tal que  $\varphi_0 = k(\Delta_0)$ ,  $f_0$  es un flujo de valor maximal  $\varphi_0$  y  $\Delta_0$  es un conjunto disjunto de capacidad minimal.

Entonces no puede existir subconjunto  $\Delta_1$  de  $\Delta_0$  que difieran de  $\Delta_0$  en arcos de capacidad no nula y tal que todo camino que vaya de  $p_1$  a  $p_n$  encuentre a  $\Delta_1$ ; pues, si existiera  $\Delta_1$ , el flujo no podría exceder del valor  $k(\Delta_1) < k(\Delta_0)$ . Por consiguiente,  $\Delta_0$  solo difiere de un corte en arcos de capacidad nula.

Del valor de  $\varphi$  podemos decir que

$$\varphi = \sum_{(i,j) \in (\bar{A}, A)} f(p_i, p_j) = \sum_{(i,j) \in (\bar{A}, A)} (k_{ij} - \gamma_{ij})$$

y, como  $(i,j) \in (\bar{A}, A)$  entonces  $\bar{\gamma}_{ij} = 0$  (si no  $p_j$  estaría marcado);  $\varphi = k(\bar{A}, A)$

Siendo el flujo igual a la capacidad del conjunto disjunto  $k(\bar{A}, A)$ ,  $\varphi$  es el vector maximal del flujo y  $(\bar{A}, A)$  es, después de suprimir eventualmente los arcos de capacidad nula, un corte de capacidad minimal.

### ALGORITMO NO. 2

**Paso 1.** Hacer pasar un flujo cualesquiera. Este se hace salir del nodo fuente, de nodo a nodo, verificando la propiedad de conservación del flujo. Si debido a las capacidades el flujo compatible con las capacidades de los arcos.

**Paso 2.** Investigar si el flujo es completo. Se necesita verificar esto ya que si el flujo no es completo quiere decir que existe al menos un camino del nodo fuente que llega hasta el nodo final y tal que ninguno de sus arcos este saturado. Se aumenta sucesivamente el flujo de este camino en una unidad hasta lograr que al menos se sature uno de los arcos. Se repite hasta lograr que el flujo sea completo.

**Paso 3.** A partir de un flujo completo marcar los vértices de la red como sigue:

i) Marcar la fuente con el signo (+), si el vértice  $x_i$  está marcado y el  $x_j$  no, entonces:

ii) Marcar el vértice  $x_j$  con el símbolo  $(+x_j)$  si existe un arco  $(x_i, x_j)$  no saturado.

iii) Marcar el vértice  $x_j$  con el símbolo  $(-x_j)$  si existe un arco  $(x_i, x_j)$  con el flujo no nulo.

**Paso 4.** Si este procedimiento llega a marcar el sumidero o nodo final, entonces se considera la cadena que pasa por los vértices marcados con (+) o con (-) y que va de la fuente al sumidero. Si un arco de esta cadena es orientado en el orden indicado por la secuencia de los vértices que forman la cadena, entonces el flujo de dicho arco se aumenta una unidad y en caso contrario se disminuye también en una unidad; así se obtendrá un nuevo flujo, mayor al inicial, y para el cual se sigue satisfaciendo la propiedad conservativa.

nuevo flujo, mayor al inicial, y para el cual se sigue satisfaciendo la propiedad conservativa.

Paso 5 . Se repiten los pasos 3 y 4 hasta lograr que no aparezca ninguna cadena de vértices marcados que vaya de la fuente al sumidero. (Anexo B.2)

### ALGORITMO NO. 3

Sea  $C_{ij}$  la capacidad del arco  $(i,j)$ .

Paso 0. Sea  $F_{ij}$  un flujo realizable factible etiquetemos el vértice origen con  $\{d, \infty\}$  e ir al paso 1

Paso 1. Sea  $Y$  el conjunto de nodos que han recibido una doble etiqueta  $\{k, \gamma\}$  donde  $k$  es el nombre del nodo del cual se llega a  $Y$

$\gamma \in \mathbb{R} \pm \{0\}$  denota la cantidad con la cual se llega .

### MARCAJE DIRECTO

a) Se puede etiquetar un vértice  $j \in Y$  si existe una arista  $(i,j)$  con  $i \in Y$  tal que  $F_{ij} < C_{ij}$ , a  $j$  se le da la etiqueta

$\{i, \text{Mín} \{\gamma, C_{ij} - F_{ij}\}\}$  y se marca la arista  $(i,j)$  con signo (+)

### MARCAJE INVERSO

b) Se puede marcar un vértice  $j \notin Y$  si existe una arista de  $j$  a  $i$  con  $i \in Y$ , tal que el flujo ( $F_{ij} > 0$ ); a  $j$  se le etiqueta con;

$\{i, \text{Mín} \{\gamma, F_{ij}\}\}$  y la arista  $(j,i)$  se marca con el signo (-)

Paso 2. a) Si el sumidero o final ha recibido una etiqueta ir al paso 3.

b) Si no, incluir en  $Y$  el nodo  $j \neq f$  que acaba de ser etiquetado y regresar al paso 2

c) Si ya no se puede etiquetar otro nodo, entonces terminar y el flujo  $F_{ij}$  es el máximo.

Paso 3. Habiendo sido etiquetado el vértice final ( $f$ ) con la etiqueta  $\{k, \gamma\}$  se dice que  $k$  es el antecesor de ( $f$ ) y así se puede remontar de antecesor en antecesor hasta el nodo (1) lo que define un camino de 1 a  $f$  si de este camino se añade la arista ( $f, l$ ) se obtiene un circuito donde

$$T^+ = \{ \text{aristas marcadas con (+)} \} \cup \{ (f, l) \}$$

$$T^- = \{ \text{aristas marcadas con (-)} \}$$

Se define un nuevo flujo  $F'_{ij}$  como sigue:

$$F'_{ij} \begin{cases} F_{ij} + \gamma & \text{si } (i, j) \in T^+ \\ F_{ij} - \gamma & \text{si } (i, j) \in T^- \\ F_{ij} & \text{si } (i, j) \text{ al circuito obtenido} \end{cases}$$

Se borran todas las etiquetas y marcas de las aristas y se regresa al paso 1. (Anexo B.3)

Este último algoritmo se utilizará para el programa, ya que este algoritmo en la primera iteración se efectúa con un flujo nulo, en la que todas las componentes son ceros. Y en cada iteración sucesiva se realiza partiendo del flujo construido durante la iteración precedente, pero este flujo no es estrictamente 1, como se hace en otros algoritmos sino  $k$ . Entonces éste último algoritmo el mejor método a programar, porque el número de iteraciones es menor y llega a converger más rápido.

## II.4 APLICACIONES

### PINTURA Y TAPICERÍA KRUMLEY

La familia Krumley ha pintado casi cada casa en Bula, Texas. La familia es bastante grande y existen 3 cuadrillas de trabajadores disponibles para todos los meses de primavera y verano. Eduard Krumley la cabecera de la compañía, ha sido contratado para 5 trabajos durante esta estación. Cada trabajo puede ser empezado durante las fechas y debe ser completado en o antes de la fecha convenida en el futuro. Krumley tiene gran popularidad en los negocios para ejecutar los trabajos. Los datos de cada uno de los trabajos se dan a continuación.

Trabajo	Fecha de inicio	Fecha de termino	Tiempo requerido (Semanas)
A	Abril 1	Abril 28	2
B	Abril 14	Mayo 12	4
C	Abril 1	Mayo 26	7
D	Abril 21	Mayo 26	3
E	Abril 1	Mayo 26	6

Los trabajos no deben ser hechos continuamente. Sin embargo el Sr. Krumley puso en práctica el asignar cuadrillas semejantes un camino en cualquier semana, estas cuadrillas deben trabajar en un solo trabajo. Además, dos cuadrillas no pueden trabajar en el mismo trabajo al mismo tiempo.

Usando estos datos, el Sr. Krumley quiere conocer si es posible terminar todos los trabajos en las fechas preestablecidas, en la fig. 2.4 se da un tipo de red que representa el problema antes citado, con esta gráfica y utilizando flujo máximo se puede contestar la pregunta del Sr. Krumley, en la gráfica los nodos 1 al 8 representan la ocho semanas que están entre Abril 1 y Mayo 26. Los nodos 9 a 13 representan los trabajos A al E, respectivamente. Los nodos 0 y 14 son el origen y el sumidero del modelo del problema de flujo máximo.

Los arcos originados del nodo cero todos tienen capacidad 3, reflejando el hecho de que existen 3 cuadrillas por semana de trabajo las cuales están disponibles cada semana. Los arcos terminales al nodo 14 todos tienen capacidad igual al número de cuadrillas semanales requeridas para completar el trabajo asociado con el nodo origen de cada arco.

Todos los demás arcos tienen capacidad de 1 unidad y representan la posible asignación de una cuadrilla semanal en un periodo semanal largo especificando el trabajo.

Utilizando la metodología de flujo máximo se puede ver que el flujo máximo es de 22, y todos los trabajos por lo tanto se cumplirán en la fecha preestablecida.

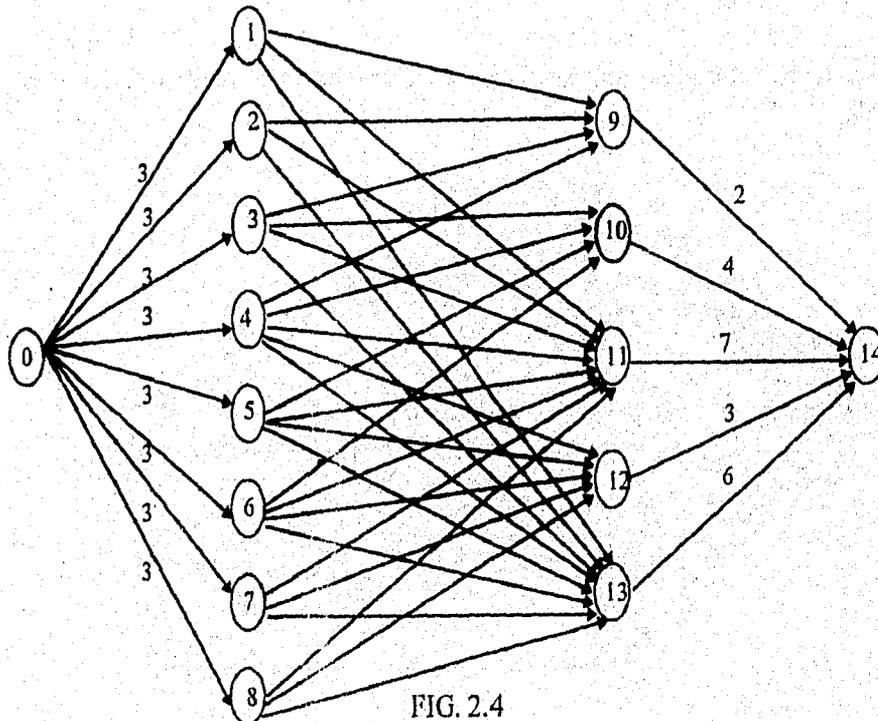


FIG. 2.4

OPERACIÓN CUARENTENA

El servicio de plagas de Estados Unidos ha determinado que la migración anual del insecto de grano esta por empezar otra vez este año. A menos que se tome una medida exhaustiva, los agricultores estén esperando sufrir grandes perdidas económicas. A través de los años, las rutas de migración del insecto de grano ha sido proyectado usando esta información, al servicio le gustaría determinar el camino de mínimo costo para asegurar que cada insecto pueda sufrir una aplicación o dosis del insecticida. Desde los costos del "polvo" de cualquier ruta de migración es directamente proporcional al máximo número de insectos que podría usar la ruta de la representación esquemática de migración que se da en la fig. 2.5, la cual fue dada de acuerdo a la pregunta solicitada. En la figura los nodos son puntos de conexión de los arcos, los cuales representan las rutas de migración respectivamente. Las capacidades del arco son los números máximos de insectos (en miles) esperados a usar la ruta particular. Estas capacidades están basadas en los información de años anteriores.

El sistema de servicios de plagas formuló este problema como flujo máximo y la solución se da con el corte mínimo, esto es, un sitio de ruta de migración de la capacidad mínima total que si se remueve de la red podría prohibir cualquier insecto de alcanzar el nodo 18.

Todas las rutas de migración en el corte mínimo reciben gran cantidad de insecticida.

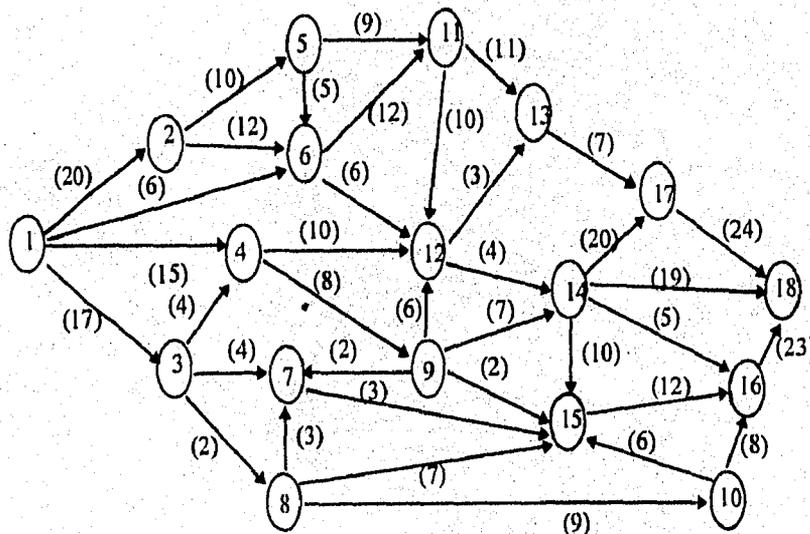


FIG. 2.5

**CAPITULO III**

**PROBLEMA DE FLUJO MÁXIMO A  
COSTO MÍNIMO EN UNA RED**

### III. PROBLEMA DE FLUJO MÁXIMO A COSTO MÍNIMO EN UNA RED

#### III.1 DEFINICIÓN DEL PROBLEMA

En problema de flujo máximo con costo mínimo en una red, se desea embarcar los recursos disponibles a través de la red, para satisfacer la demanda a un costo mínimo. Matemáticamente, este problema se representa de la siguiente manera:  
(Las sumas se toman sobre los arcos existentes):

$$\text{Minimizar } \sum_{j=1}^n \sum_{i=1}^m C_{ij} x_{ij}$$

s. a

$$\sum_{j=1}^m x_{ij} - \sum_{k=1}^m x_{ki} = 0$$

$$i = 1, \dots, m$$

$$x_{ij} \geq 0$$

$$j = 1, \dots, m$$

Las restricciones se llaman las ecuaciones de conservación de flujo o ecuaciones de Kirchhoff e indican que, en la red, no se puede crear ni destruir flujo. En las ecuaciones de conservación,  $\sum_{j=1}^m x_{ij}$  representa el flujo total que sale del nodo  $i$ , mientras que  $\sum_{k=1}^m x_{ki}$  es el flujo total que entra al nodo  $i$ .

Antes de dar el algoritmo referente al método de flujo máximo a costo mínimo, es decir, el problema de distribución debe considerarse lo siguiente:

Considérese una red circulatoria  $G:(V,A)$  con  $V=\{1,2,\dots,n\}$ , y supóngase que se tiene  $0 \leq a_{ij} \leq b_{ij}$ . Son los flujos mínimos y máximos sobre la arista  $(i,j) \in A$  y denotamos por  $C_{ij}$  el costo unitario del flujo que pasa por la arista  $(i,j) \in A$ . El elemento  $x_{ij}$  es el flujo existente en el arco  $(i,j)$ .

El modelo de programación matemática se escribe entonces

$$\text{Min } z = \sum_{(i,j) \in A} C_{ij} x_{ij}$$

s.a.

$$\sum_{j \in S_i} x_{ij} - \sum_{j \in E_i} x_{ji} = 0 \quad \forall i \in V$$

$$a_{ij} \leq x_{ij} \leq b_{ij} \quad (i,j) \in A$$

El modelo anterior es equivalente a

$$\text{Min } z = \sum_{i=1}^n \sum_{j=1}^n C_{ij} x_{ij}$$

s.a.

$$\sum_{i=1}^n \sum_{j=1}^n (x_{ij} - x_{ji}) = 0$$

$$\forall i = 1, \dots, n$$

$$x_{ij} \leq b_{ij} \geq w_{ij}$$

$$a_{ij} \leq x_{ij} \leq b_{ij}$$

$$x_{ij} \leq a_{ij} \geq v_{ij}$$

$$j = 1, \dots, n$$

donde  $a_{ij} = b_{ij} = 0$  si  $(i,j)$  no pertenece  $V$ . La función objetivo anterior, minimiza el costo del flujo mandado por la red.

De tal manera que el problema dual del primal es

$$\text{Máx} \sum_{i=1}^n \sum_{j=1}^n (-w_{ij} b_{ij} + (a_{ij} v_{ij}))$$

s.a.

$$u_i - u_j + v_{ij} - w_{ij} \leq C_{ij}$$

$$v_{ij} \geq 0; w_{ij} \geq 0$$

$$i=1, \dots, n; j=1, \dots, n$$

$u_{ij}$  sin restricción, donde

$$\bar{C}_{ij} = C_{ij} - u_i + u_j$$

ó

$$\text{Min} \sum_{j=1}^n w_{ij} a_{ij} + \sum_{j=1}^n v_{ij} b_{ij}$$

s.a.

$$u_i - u_j + w_{ij} - v_{ij} \geq -c_{ij}$$

$$v_{ij} \geq 0; w_{ij} \geq 0$$

$$i=1, \dots, n; j=1, \dots, n$$

$u_{ij}$  sin restricción

$$\text{donde } \bar{C}_{ij} = C_{ij} + u_i - u_j$$

Si  $x_{ij}$ ,  $u_i$ ,  $v_j$  y  $w_j$  son solución factible a los problemas primal y Dual, respectivamente, entonces estas, soluciones son óptimas si y solo si, se cumple el teorema de las holguras complementarias (complementariedad) esto es si

$$v_{ij} (x_{ij} - a_{ij}) = 0 \dots (1)$$

$$w_{ij} (b_{ij} - x_{ij}) = 0 \dots (2)$$

$$x_{ij}(C_{ij} - w_i + u_j - v_{ij} + w_{ij}) = 0 \dots (3)$$

para cada arista  $(i,j)$  en la gráfica.

Para cada arista  $(i,j)$  las ecuaciones de 1 a 3 se cumplen si satisfacen las siguientes condiciones que son mutuamente excluyentes:

$$x_{ij} = a_{ij}; \quad w_{ij} = 0; \quad v_{ij} = C_{ij} - u_i - u_j \dots (4)$$

$$a_{ij} < x_{ij} < b_{ij} \quad v_{ij} = w_{ij} = C_{ij} - u_i + u_j \dots (5)$$

$$x_{ij} = b_{ij}; \quad v_{ij} = 0; \quad w_{ij} = -C_{ij} + u_i - u_j \dots (6)$$

De las ecuaciones (4) y (6) se tiene que se debe encontrar valores para  $x_{ij}$  (flujo de la red) y para las variables  $u_i$  (Potenciales) y así, obtener los valores de las variables  $v_{ij}$  y  $w_{ij}$ , luego entonces:

$$v_{ij} = \text{Máx} \{0, C_{ij} - u_i + u_j\} \dots (7)$$

$$w_{ij} = \text{Máx} \{0, -C_{ij} + u_i - u_j\} \dots (8)$$

definimos

$$\bar{C}_{ij} = C_{ij} - u_i + u_j \text{ llamado el coeficiente de costo reducido en la arista } (i,j).$$

En estos términos se tiene que dado un flujo  $x_{ij}$  es óptimo si solo si existe un vector de potenciales  $(u_i)$  tal que para cada arista una de las tres siguientes condiciones se cumple

$$\text{L:} \quad \bar{C}_{ij} > 0; \quad x_{ij} = a_{ij}$$

$$\text{B:} \quad \bar{C}_{ij} = 0; \quad a_{ij} \leq x_{ij} \leq b_{ij}$$

$$\text{K:} \quad \bar{C}_{ij} < 0; \quad x_{ij} = b_{ij}$$

CAPITULO III. PROBLEMA DEL FLUJO MAXIMO A COSTO MINIMO EN UNA RED.

Por otra parte podemos implicar para flujos circulatorios (no necesariamente factibles) y vector de potenciales no óptimos, se tiene que un arco se encuentra en alguno de los conjuntos mutuamente excluyentes que son los siguientes:

$$L_1: \quad \bar{C}_{ij} > 0; \quad x_{ij} < a_{ij}$$

$$L_2: \quad \bar{C}_{ij} > 0; \quad x_{ij} > a_{ij}$$

$$B_1: \quad \bar{C}_{ij} = 0; \quad x_{ij} < a_{ij}$$

$$B_2: \quad \bar{C}_{ij} = 0; \quad x_{ij} > b_{ij}$$

$$K_1: \quad \bar{C}_{ij} < 0; \quad x_{ij} < b_{ij}$$

$$K_2: \quad \bar{C}_{ij} < 0; \quad x_{ij} > b_{ij}$$

Un arco no conformable se define como aquel arco que no cumple con las restricciones anteriores, es decir, que el flujo existente en el arco (i,j) es mayor a la capacidad máxima o menor a la capacidad mínima del arco. Si el flujo asociado al arco es mayor o igual a la capacidad mínima, o menor o igual a la capacidad máxima se dice que el arco es conformable.

Se dice que una arista es conformable si se encuentra en alguno de los estados  $L$ ,  $B$  y  $K$ , y las aristas son no conformables si se encuentran en algunos de los estados  $L_1, L_2, B_1, B_2, K_1, K_2$  las condiciones para pasar de un estado no conformable a un estado conformable son las siguientes:

Estado no conformable	Pasar a estado conformable	Si la desviación se hace cero
$L_1$	$L$	$a_{ij} - x_{ij}$
$L_2$	$L$ ó $B$	$\bar{C}_{ij}(x_{ij} - a_{ij})$
$B_1$	$B$	$a_{ij} - x_{ij}$
$B_2$	$B$	$x_{ij} - b_{ij}$
$K_1$	$B$ ó $K$	$\bar{C}_{ij}(x_{ij} - a_{ij})$
$K_2$	$K$	$x_{ij} - b_{ij}$

Los números  $a_{ij} - x_{ij}$ ,  $\bar{C}_{ij}(x_{ij} - a_{ij}), \dots, x_{ij} - b_{ij}$ , relacionados con los estados  $L_1, L_2, \dots, K_2$  respectivamente, se llamarán por definición desviaciones de arcos no conformables; son estrictamente positivos. Todo arco conformable tiene, por definición, una desviación cero.

### III.2 DIFERENTES MÉTODOS DE SOLUCIÓN

Todos los problemas antes citados en redes especiales, se pueden ser solucionados por programación lineal, mediante métodos como el Simplex, Simplex Revisado o Dual. En la práctica, su estructura especial de tales problemas ha inspirado, desde el comienzo, algoritmos más sencillos y eficaces, que son de dos tipos:

1) El primer tipo es exactamente una adaptación del método simplex (algoritmo primal), para un problema de transporte simple. El algoritmo de transporte (también llamado método de "stepping-stone") se puede extender.

2) El segundo tipo se desarrollo inicialmente para problemas del tipo de transporte y después se generalizó para el problema general de programación lineal.

Obviamente el problema de flujo con costo mínimo es un programa lineal y se puede resolver de varias formas. Una forma consiste en aplicar el algoritmo simplex primal común. Lo que se busca es la simplificación del método simplex que pueda aplicar directamente a la red sin la necesidad de una tabla simplex.

A continuación se presenta el algoritmo de etiquetado para el método simplex para redes

Ya sea para cálculos manuales o por computadora, hay formas simplex y convenientes de mantener la información requerida para resolver un problema de flujo con costo mínimo, con cotas inferiores y superiores, mediante el método simplex para redes. Supóngase que con cada nodo  $j \neq n$  se asocia una etiqueta,  $L(j) = (\pm i, \Delta(j))$ , conteniendo dos piezas de información. la segunda componente  $\Delta(j)$ , en  $L(j)$  indica la estimación presente para el valor del cambio de flujo. Si la primera componente  $\pm i$ ,  $L(j)$  indica el nodo anterior en el ciclo a lo largo del cual cambiara el flujo. Si la primera componente en  $L(j)$  es  $+i$ , entonces se añadirá flujo al arco  $(i,j)$ ; por el contrario, si la primera componente  $-i$  entonces se restara flujo del arco  $(j,i)$ . El algoritmo de etiquetado resulta ser el siguiente.

PASO INICIAL

0. Seleccionase una solución básica factible y dense a los  $x_{ij}$  sus valores en la solución. Si no se dispone ya de una solución básica factible, utilícese variables artificiales.

PASO PRINCIPAL

1. calcúlese las variables duales. tómese  $w_m = 0$  si  $w_i$  ya se calculó,  $w_j$  no se ha calculado, y el arco  $(i,j)$  es un arco básico, entonces tómese  $w_j = w_i - c_{ij}$ . Si  $w_i$  ya se calculó,  $w_j$  no se ha calculado y el arco  $(j,i)$  es un arco básico, entonces tómese  $w_j = w_i + c_{ij}$ . Repítase el paso 1 hasta que todas las  $w_i$  se han calculado.

2. En este paso se verifica la optimalidad y en caso de que no sea la solución optima se selecciona la variable entrante. Si cada variable no básica tiene  $x_{ij} = a_{ij}$  y  $z_{ij} - c_{ij} \leq 0$ , o bien,  $x_{ij} = u_{ij}$  y  $z_{ij} - c_{ij} \leq 0$ , deténgase; se ha obtenido la solución optima. En caso contrario, borre todas las etiquetas. Si  $z_{pq} - c_{pq} < 0$  y  $x_{pq} = u_{pq}$ , entonces tómese  $s = p$ ,  $t = q$ ,  $(g,h) = (p,q)$ , y  $L(s) = (-t, x_{pq} - l_{pq})$ ; o bien, si  $z_{pq} - c_{pq} > 0$  y  $x_{pq} = l_{pq}$ , tómese  $s = q$ ,  $t = p$ ,  $(g,h) = (p,q)$  y  $L(s) = (+t, u_{pq} - x_{pq})$ .

3. a) En este paso se determina un ciclo. Si el nodo  $i$  tiene etiqueta, el nodo  $j$  no tiene etiqueta, y el arco  $(i,j)$  es básico, entonces tómese  $L(j) = (+i, \Delta_j)$ , donde  $\Delta_j = \min \{ \Delta_i, u_{ij} - x_{ij} \}$  si  $u_{ij} - x_{ij} < \Delta_i$ , tómese  $(g,h) = (i,j)$ .

b) Si el nodo  $i$  tiene etiqueta, el nodo  $j$  no tiene etiqueta, y el arco  $(j,i)$  es básica, entonces tómese  $L(j) = (-i, \Delta_j)$ , donde  $\Delta_j = \text{Mínimo} \{ \Delta_i, X_{ij} - a_{ij} \}$ . Si  $x_{ij} - a_{ij} < \Delta_i$ , tómese  $(g,h) = (j,i)$ .

c) repetir el paso 3 hasta que el nodo  $t$  este etiquetado.

4. Cambio del flujo a lo largo del ciclo y determina la variable que sale. Sea  $\Delta = \Delta_i$ . Si la primera componente en  $L(t)$  es  $+k$ , entonces súmese  $\Delta$  a  $x_{kt}$ ; en caso contrario, si la primera componente en es  $-k$ , réstese delta de  $x_{kt}$ . Se busca hacia atrás el nodo  $k$  y se repite el proceso hasta que el nodo  $t$  sea alcanzado en el proceso de buscar hacia atrás.

5. Aquí se actualiza la base. Si  $(g,h) = (p,q)$ , se regresa al paso 2. En caso contrario añada  $(p,q)$  a la base, elimine  $(g,h)$  de la base, y repítase el paso número 1.

Como se puede observar tiene gran similitud con el algoritmo que se presentará más adelante, pero la diferencia entre los métodos es que en el anterior se necesita tener una base optima para empezar y se van sacando las variables no optimas, en el otro se puede

empezar con un flujo cualesquiera (es decir, que  $x_{ij}$  sea arbitrario). El otro método utiliza el método de los potenciales donde se va incrementando el valor de estas variables sin que los arcos dejen de ser conformables.

### III.3 ALGORITMO PARA ENCONTRAR LOS POTENCIALES

Para encontrar el valor de cada uno de los potenciales se toma el algoritmo de longitud mínima propuesto por Ford y cuyo problema es el siguiente:

$$\begin{aligned} \text{Máx } z &= u_0 - u_n \\ \text{s.a.} \\ u_i - u_j &\leq a_{ij} \end{aligned} \quad (1)$$

donde  $a_{ij}$  es el costo asociado a cada arco.

Asociar a cada variable  $u_i$  a cada vértice  $v_i$  y asignar a

$$u_0 = 0, u_i = -\infty, i \neq 0.$$

Para todo arco  $(i,j)$  tal que  $u_i - u_j > a_{ij}$ , sustituir  $u_j$  por  $u_i - a_{ij}$

Cuando no se puede hacer ninguna modificación más se tiene  $u_i - u_j > a_{ij}$ , para todo arco  $(i,j)$  y existe, para todo vértice  $p_k \neq p_0$ , al menos un vértice  $p_{k_j}$  tal que  $(p_k, p_{k_j}) \in A$  y que  $u_{k_j} - u_{k_i} = a_{k_j k_i}$ . (2)

Por lo tanto, partiendo de  $p_n$  se puede tomar al menos una secuencia  $(p_n, p_{k_1}, \dots, p_{k_i} = p_0) = \gamma_0$  que es un camino, y sumando las relaciones 2 que corresponden a los arcos de  $\gamma_0$ , se obtiene :

$$0 - u_n = a_{0k_1} + a_{k_1 k_2} + \dots + a_{k_i p_0} = a(\gamma_0)$$

$\gamma_0$  es el camino más corto siendo dicha longitud  $a(\gamma_0) = -u_n$ .

Demostración. Para todo camino

$$\gamma = (p_0, p_{a_1}, \dots, p_{a_i}, \dots, p_n)$$

y para todo programa  $\bar{u}_i$  de 2. Tal que  $\bar{u}_0 = 0$  se tiene  $\bar{u}_{a_i} - \bar{u}_{a_j} \leq a_{a_i a_j}$  o sea sumando miembro a miembro todas las relaciones que corresponden a los arcos de  $\gamma$   $0 - \bar{u}_n \leq a(\gamma)$ .

Así el valor de  $(-u_n)$  asociado a todo programa 1. es como máximo igual a la longitud de todo camino de  $p_0$  a  $p_n$ . Como el algoritmo anterior permite determinar un camino  $\gamma_0$  tal que  $(-\bar{u}_n) = a(\gamma_0)$ , este camino da la longitud minimal.

### III.4 MÉTODO DE LAS DESVIACIONES

Dada la red de transporte, se considera como primer flujo  $x_{ij} = 0$  y un vector de potenciales arbitrario. Dado estas condiciones iniciales se determina si los arcos son o no conformables.

1) Si existe un arco  $(s,t)$  no conformable ir al paso siguiente de otra manera se ha llegado a la solución óptima.

2) Aplique el método de las etiquetas para formar un circuito que contenga  $(s,t)$  sea  $t$  el origen y  $s$  el destino si el arco de  $(s,t)$  esta en alguno de los estados  $L_1, B_1$  ó  $K_1$ , de otra manera sea  $s$  el origen y  $t$  el destino.

En los arcos que pertenecen a  $L_1, B_1$  ó  $K_1$ ; el flujo se aumentará, mientras que en los otros conjuntos el flujo se decrementa para así tener arcos conformables.

2a) Modificación del flujo. Si existe un ciclo que contiene a la arista  $(s,t)$  aumente el flujo en la trayectoria determinada por el método de las etiquetas y el arco  $(s,t)$ , donde  $\alpha$  es igual a:

$$\begin{aligned} \text{Min}[\alpha_s, a_{st} - x_{st}] & \quad \text{si } (s,t) \in L_1 \\ \text{Min}[\alpha_s, b_{st} - x_{st}] & \quad \text{si } (s,t) \in B_1, \text{ ó } K_1 \\ \text{Min}[\alpha_t, b_{st} - x_{st}] & \quad \text{si } (s,t) \in L_2 \text{ ó } B_2 \\ \text{Min}[\alpha_t, x_{st} - b_{st}] & \quad \text{si } (s,t) \in K_2 \end{aligned}$$

Si la arista es conformable entonces regresar al paso 1 de otra manera borre las etiquetas y regrese al paso 2.

2b) No existe ciclo que contenga  $(s,t)$ . Sean  $\bar{I}$  e  $I$  los conjuntos de nodos etiquetados y no etiquetados respectivamente. Entonces, calcular  $\theta$  (ver subrutina de cambio de potenciales).

Si  $\theta$  es igual a  $\infty$  el problema no tiene solución factible de otra manera pasar al paso 3.

3) Definalos nuevos potenciales en los nodos, como:

$$u_i = u_i \text{ si } i \in I;$$

$u_i = u_i + \theta$  si  $i \in \bar{I}$  si  $(s,t)$  no es conformable regresar a 1, de otra manera regrese a 2.

SUBROUTINA 1 MÉTODO DE LAS ETIQUETAS

Denote por 1 el origen y por  $n$  el destino lo que se desea encontrar un ciclo que contenga el arco  $(1, n)$ .

1. Poner la etiqueta  $(-\infty)$  al origen.

2. Se  $\alpha_i$  un vértice etiquetado  $(h, \alpha_i)$  etiquetado con  $(i, \alpha_j)$  cada vértice  $j$  que satisfaga alguna de las siguientes condiciones:

$$\begin{aligned} \lambda; (i, j) \in A, \quad \bar{C}_{ij} > 0; x_{ij} < a_{ij} \\ \mu; (i, j) \in A, \quad \bar{C}_{ij} \leq 0; x_{ij} < b_{ij} \\ \nu; (j, i) \in A, \quad \bar{C}_{ji} \geq 0; x_{ji} > a_{ji} \\ \rho; (j, i) \in A, \quad \bar{C}_{ji} < 0; x_{ji} > b_{ji} \end{aligned}$$

tomando a  $\alpha_j$  es igual a  $\text{Min}[\alpha_i, a_{ij} - x_{ij}]$ ,  $\text{Min}[\alpha_i, b_{ij} - x_{ij}]$ ,  $\text{Min}[\alpha_i, x_{ji} - a_{ji}]$ ,  $\text{Min}[\alpha_i, x_{ji} - b_{ji}]$  según se encuentren en cada uno de los casos anteriores.

3) Si el destino  $n$  es etiquetado considere la trayectoria de 1 a  $n$  en que cada vértice etiquetado indica el vértice sucesor en tal trayectoria y aumente en  $\alpha_n$  el flujo sobre estas aristas en la trayectoria.

4) Si el destino final  $n$  no ha sido etiquetado aplique la subrutina 2.

SUBROUTINA 2. CAMBIO DE POTENCIALES

Denote por  $\bar{I}$  e  $I$  el conjunto de vértices etiquetados y no etiquetados al final de la subrutina de etiquetación.

Defina los siguientes conjuntos

$$\begin{aligned} S_1 &= \{(i, j): i \in \bar{I}, j \in I, \bar{C}_{ij} > 0, x_{ij} \leq b_{ij}\} \\ S_2 &= \{(i, j): i \in I, j \in \bar{I}, \bar{C}_{ij} < 0, x_{ij} \geq a_{ij}\} \end{aligned}$$

Si  $S_1$  y  $S_2 = \emptyset$ , entonces hagamos el valor de  $\theta = \infty$  de otra manera

$$\theta = \text{Min} \left[ \text{Min}_{s_1}(\bar{C}_{ij}), -\text{Min}_{s_2}(\bar{C}_{ij}) \right]$$

Si  $\theta$  es finito entonces los nuevos valores de los potenciales son

$$u'_i = u_i \quad \text{si } i \in I,$$

$$u'_i = u_i + \theta \quad \text{si } i \in \bar{I}$$

Así mismo, la nueva matriz de costos relativos (ó coeficiente de costo reducido) es:

$$\bar{C}'_{ij} = \begin{cases} \bar{C}_{ij} - \theta & i \in \bar{I}, j \in I \\ \bar{C}_{ij} + \theta & i \in I, j \in \bar{I} \\ \bar{C}_{ij} & \text{otra manera} \end{cases}$$

El diagrama de flujo del algoritmo citado se puede observar en el anexo C.

**CAPITULO IV.**

**OPERACIÓN DEL SISTEMA DE**

**COMPUTO**

IV. OPERACIÓN DEL SISTEMA DE COMPUTO

IV.1 REPRESENTACIÓN DE RESULTADOS

Dado que el problema de flujo máximo a costo mínimo resuelto por el método de las desviaciones requiere de un análisis de los arcos para el proceso de etiquetación y el proceso de cambio de potenciales; la programación de este método nos ahorra todo ese análisis y todas las iteraciones. Entonces, el programa presenta un análisis del estado al que pertenece cada uno de los arcos de la red. La tabla que aparece después del análisis de los arcos (Fig. 4.1), es donde se lleva a cabo el proceso de etiquetación, los elementos de las columnas denotados por  $v_i$  y  $v_j$  son los vértices existentes en la red. Los elementos  $u_i$  y  $u_j$  son las variables duales o los potenciales. Estos elementos pueden tener el valor de cero al comenzar el método o ser calculadas con el método desarrollado por Fulkerson que ya ha sido expuesto en el capítulo anterior, el paquete calcula los potenciales por dicho método.

En la intersección de los elementos  $v_i$  y  $v_j$  tales que exista un arco en la red se tendrán 4 elementos que son los que se ven en el recuadro; el primer elemento  $x_{ij}$  es el flujo existente en el arco (i,j), el segundo elemento denota el flujo existente menos la capacidad máxima.

El tercero es el flujo menos la capacidad mínima del arco (i,j) y por último  $\bar{C}_{ij}$  es el coeficiente de costos reducidos o  $\bar{C}_{ij} = C_{ij} - u_i + u_j$  como se expuso en el capítulo anterior

De esta manera son presentados los datos en la tabla.

		$v_1$	$v_2$		$v_i$		$v_n$
		$u_1$	$u_2$		$u_i$		$u_n$
$v_1$	$u_1$						
$v_2$	$u_2$						
					$x_{ij}$	$x_{ij} - b_i$	
					$x_{ij} - a_i$	$\bar{C}_{ij}$	
$v_n$	$u_n$						

Fig. 4.1

## CAPITULO IV. OPERACION DEL SISTEMA DE COMPUTO

### IV.2 LISTADO DEL PROGRAMA

A continuación se presenta el listado del paquete DESVIA, ya que este paquete fue estructurado para redes de 3 a 10 nodos, por eso se señala que tipo de modificaciones se deben hacer en caso de que se desee ampliar el rango de nodos.

```
PROGRAM TESINA; (binN)
USES CRT,DOS,GRAPH,PORTADA,PRINTER,UNIDAD;
TYPE
  MAT =ARRAY [0..10,0..10] OF INTEGER;
  VECTORX =ARRAY [0..10] OF INTEGER;
```

En esta parte se debe cambiar el 10 por el número de nodos que se tengan o se desee que el programa resuelva.

```
VAR
  MAS,ACA,T,YEA,ZETA,MOM,GMOD,P,PAR,RESPU,OPCION,GDRI,N,JA,NN:INTEGER;
  EN,DIG,RIN,ASI,MIN,ERRC,PMAX,LL,IM,IK,K,RES,MM,W,ZAZ,I,LL,DD:INTEGER;
  CMIN,CMAX,ANDO,DDD,KK,OPCI,PP,CEU,JK,LM,TETA,NV,SV,TT2,J,MN:INTEGER;
  CUENTA,OPER,COSTO,TT,D,MINI,TT1,CENX,CENY,KKK,NADA,NI,S,VV:INTEGER;
  MAXCOLOR,RADIO,PASANO,XASP,YASP,MAXX,MAXY:WORD;
  Y,X,NCON,CC,Z,A,B,C,ZZ,AA,FLUI:MAT;
  U,V,BETA,VEC,OAMA:VECTORX;
  OLDEXITPROC:POINTER;
  OP,CH:CHAR;
  HEADER,SI,PATHTODRIVER,OO:STRING;
  POINTS:ARRAY[0..15] OF POINTTYPE;
  PUNTITO:ARRAY[0..15,0..15] OF POINTTYPE;
```

Aquí también se debe cambiar el número de nodos, ya que estas variables son las correspondientes a la graficación de los nodos o vértices.

```
VIEWINFO,VIEWPORT:VIEWPORTTYPE;
RADIANS:REAL;
ARA,SOL,ARAI:TEXT;
SAN,SUN:STRING[12];
LAB:STRING[3];
INGRAPHICSMODE:BOOLEAN;
LABEL 5;
```

```
PROCEDURE SONIER;
BEGIN
  SOUND(50); DELAY (100); NOSOUND;
END;
FUNCTION IN2Str(L:LoNgInt):STRING;
BEGIN
  { Str(L,S); IN2Str := S; }
END; { IN2Str }
FUNCTION RANDCOLOR:WORD;
BEGIN
  RANDCOLOR := RANDOM(MAXCOLOR)+1;
END; { RANDCOLOR }
PROCEDURE DEFAULTCOLORS;
BEGIN
  SetColor(MAXCOLOR);
```

```

END; { DEFAULTCOLORS }
PROCEDURE DRAWBORDER;
BEGIN
  DEFAULTCOLORS; SETLINESTYLE(SolidLN, 0, NORMWidth);
  GetViewSettings(VIEWPORT); with VIEWPORT DO
  RectAngle(0, 0, X2-X1, Y2-Y1);
END; { DrawBORDER }
PROCEDURE FullPORT;
BEGIN
  SetVIEWPORT(0, 0, MAXX, MAXY, CLIPON);
END; { FullPORT }
PROCEDURE STATUSLINE(Msg : STRING);
BEGIN
  FullPORT; DEFAULTCOLORS; SetTextStyle(DefaultFoNt, HORizDir, 1);
  SetTextJustify(CeNterText, TOpText); SETLINESTYLE(SolidLN, 0, NORMWidth);
  SetFillStyle(EmptyFill, 0); Bar(0, MaxY-(TextHeight('M')+4), MaxX, MaxY);
  RectAngle(0, MaxY-(TextHeight('M')+4), MaxX, MaxY);
  OUTTEXTXY(MaxX div 2, MaxY-(TextHeight('M')+2), Msg);
  SetVIEWPORT(1, TextHeight('M')+5, MaxX-1, MaxY-(TextHeight('M')+5), ClipON);
END; { STATUSLINE }
PROCEDURE MAINWINDOW(Header :STRING);
BEGIN
  DEFAULTCOLORS; ClearDevice; SetTextStyle(DefaultFoNt, HORizDir, 1);
  SetTextJustify(CeNterText, TOpText); FullPORT; OUTTEXTXY(MaxX div 2, 2, Header);
  SetVIEWPORT(0, TextHeight('M')+4, MaxX, MaxY-(TextHeight('M')+4), ClipON);
  DrawBORDER; SetVIEWPORT(1, TextHeight('M')+5, MaxX-1, MaxY-(TextHeight('M')+5), ClipON);
END; { MAINWINDOW }
PROCEDURE SALIDA;
VAR
  Z : ARRAY[1..2] OF CHAR;
  SON,B,I,J,D :INTEGER;
  CAR : CHAR;
BEGIN
  STATUSLINE(' PRESIONE [ENTER] PARA CONTINUAR' );
  B:=0; Z[1] := #13;
  REPEAT
    CAR:=READKEY; IF CAR=Z[1] THEN B:=1;
    IF B=0 THEN SONIER;
    UNTIL B=1; EXITPROC := OLDEXITPROC; CLOSEGRAPH;
  END;
  {$F-}{$F+}
  PROCEDURE MYEXITPROC;
  BEGIN
    EXITPROC := OLDEXITPROC; CLOSEGRAPH;
  END; { MyEXITPROC }
  {$F-}
  PROCEDURE INITIALIZE;
  VAR
    INGRAPHICSMODE : BOOLEAN;
    PATHTODRIVER : STRING;
  BEGIN
    DIRECTVIDEO := FALSE; OLDEXITPROC := EXITPROC;
    PATHTODRIVER := ""; REPEAT
      {$IFDEF Use8514}
      GDRI := IBM8514; GMOD := IBM8514H;
      {$ELSE}
      GDRI := DETECT;
      {$ENDIF}
    INITGRAPH(GDRI, GMOD, PATHTODRIVER);
    ERRC := GRAPHRESULT;
    IF ERRC <> GROK THEN
      BEGIN
        Writeln('GRAPHICS ERRO: ', GRAPHERRORMSG(ERRC));
        IF ERRC = GRFILENOTFOUND THEN
          BEGIN
            Writeln('Enter full path to BGI driver or type <Ctrl-Break> to quit:');
          END;
        END;
      END;
  END;

```

## CAPITULO IV. OPERACION DEL SISTEMA DE COMPUTO

```

    READLN(PATHTODRIVER); WRITELN;
  END
  ELSE
    HALT(1);
  END;
  UNTIL ERRC = GROK; RANDOMIZE;
  MAXCOLOR := GETMAXCOLOR;
  MaxX := GETMaxX; MaxY := GETMaxY;
END; { INITIATIZE }
PROCEDURE INICIA;
BEGIN
  PMAX:=N; FILLCHAR(POINTS,SIZEOF(POINTS),0); FILLCHAR(PUNTITO,SIZEOF(PUNTITO),0);
  FILLCHAR(CENX,SIZEOF(CENX),0);
  FILLCHAR(CENY,SIZEOF(CENY),0); FILLCHAR(RADIO,SIZEOF(RADIO),0);
END;
PROCEDURE LINETOPLAY(FLUJ:MAT);
VAR
  I,J:INTEGER;
FUNCTION ADJASP(VALUE : INTEGER) : INTEGER;
BEGIN
  ADJASP := (LONGINT(VALUE) * XASP) div YASP;
END; { AdjAsp }
BEGIN
  INICIA;
  GetAspectRatio(Xasp, Yasp); GetViewSettings(VIEWINFO);
  WITH VIEWINFO DO
  BEGIN
    CENX := ((X2-X1) div 2); CENY := (y2-y1) div 2;
    RADIO := CENY; WHILE (CENY+AdjAsp(RADIO)) < (y2-y1)-20 DO
    INC(RADIO);
  END;
  PASANG := 360 div PMAX;
  FOR I := 0 TO PMAX - 1 DO
  BEGIN
    RadiaNs := (PASANG * I) * PI / 180;
    POINTS[I].X := CENX + ROUND(COS(RADIANS) * RADIO);
    POINTS[I].Y := CENY - AdjAsp(ROUND(SIN(RADIANS) * RADIO));
  END;
  FOR I := 0 TO PMAX - 1 DO
  BEGIN
    FOR J:=0 TO PMAX-1 DO
    BEGIN
      IF Y[I+1,J+1]=1 THEN
      BEGIN
        PUNTITO[I+1,J+1].X:=(POINTS[I].X + POINTS[J].X) div 2;
        PUNTITO[I+1,J+1].Y:=(POINTS[I].Y + POINTS[J].Y) div 2;
      END;
    END;
  END;
  FOR I := 0 TO PMAX - 1 DO
  BEGIN
    FOR J := 0 TO PMAX - 1 DO
    BEGIN
      STR(I+1,LAB); OUTTEXTXY(POINTS[I].X+15, POINTS[I].Y,LAB);
      PIESLICE(POINTS[I].X, POINTS[I].Y-2,0,360,9);
    END;
  END;
  CASE OPCION OF
  1:BEGIN
    IF I<J THEN
    BEGIN
      IF Y[I+1,J+1]=1 THEN
      BEGIN
        MOVETO(POINTS[I].X, POINTS[I].Y); LINETO(POINTS[J].X, POINTS[J].Y);
      END;
    END;
    IF J>I THEN
    BEGIN
      IF Y[J+1,I+1]=1 THEN

```

```

BEGIN
  MOVETO(POINTS[I].X, POINTS[I].Y);
  LINETO(POINTS[J].X, POINTS[J].Y);
  END; END; END; END;
2: BEGIN
  IF I < J THEN
    BEGIN
      IF Y[I+1, J+1]=1 THEN
        BEGIN
          MOVETO(POINTS[I].X, POINTS[I].Y);
          LINETO(POINTS[J].X, POINTS[J].Y);
          IF (I+1) < (N/2) THEN
            OUTTEXTXY(PUNTITO[I+1, J+1].X-2, PUNTITO[I+1, J+1].Y-3, #17);
          IF (I+1) >= (N/2) THEN
            OUTTEXTXY(PUNTITO[I+1, J+1].X+1, PUNTITO[I+1, J+1].Y-3, #16);
          END;
        END;
      IF J > I THEN
        BEGIN
          IF Y[J+1, I+1]=1 THEN
            BEGIN
              MOVETO(POINTS[I].X, POINTS[I].Y);
              LINETO(POINTS[J].X, POINTS[J].Y);
              IF (I+1) >= (N/2) THEN
                OUTTEXTXY(PUNTITO[J+1, I+1].X+3, PUNTITO[J+1, I+1].Y-2, #17);
              IF (I+1) < (N/2) THEN
                OUTTEXTXY(PUNTITO[J+1, I+1].X+2, PUNTITO[J+1, I+1].Y-3, #16);
              END; END; END;
            END;
          END;
        END;
      IF Y[I+1, J+1]=1 THEN
        BEGIN
          STR(M, LAB1); MOVETO(POINTS[I].X, POINTS[I].Y); LINETO(POINTS[J].X, POINTS[J].Y);
          OUTTEXTXY(PUNTITO[I+1, J+1].X+20, PUNTITO[I+1, J+1].Y-3, 'e');
          OUTTEXTXY(PUNTITO[I+1, J+1].X+28, PUNTITO[I+1, J+1].Y, lab1);
          IF (I+1) < (N/2) THEN
            OUTTEXTXY(PUNTITO[I+1, J+1].X-2, PUNTITO[I+1, J+1].Y-3, #17);
          IF (I+1) >= (N/2) THEN
            OUTTEXTXY(PUNTITO[I+1, J+1].X+4, PUNTITO[I+1, J+1].Y-3, #16); M:=M+1; END; END;
        END;
      END;
    END;
  END;
5: BEGIN
  IF I < J THEN
    BEGIN
      IF (FLUJ[I+1, J+1]=1) OR (FLUJ[I+1, J+1]=-1) OR (FLUJ[I+1, J+1]=0) AND (Y[I+1, J+1]=1) THEN
        BEGIN
          MOVETO(POINTS[I].X, POINTS[I].Y);
          IF (FLUJ[I+1, J+1]=1) THEN
            SETLINESTYLE(0, 0, ThickWidth);
          IF (FLUJ[I+1, J+1]=-1) THEN
            SETLINESTYLE(0, 0, NORMWidth);
          IF FLUJ[I+1, J+1]=0 THEN
            SETLINESTYLE(3, 0, NORMWidth);
          LINETO(POINTS[J].X, POINTS[J].Y);
          IF (I+1) < (N/2) THEN
            OUTTEXTXY(PUNTITO[I+1, J+1].X-2, PUNTITO[I+1, J+1].Y-3, #17);
          IF (I+1) >= (N/2) THEN
            OUTTEXTXY(PUNTITO[I+1, J+1].X+2, PUNTITO[I+1, J+1].Y-3, #16);
          END; END;
        END;
      IF J > I THEN
        BEGIN
          IF ((FLUJ[J+1, I+1]=1) OR (FLUJ[J+1, I+1]=-1) OR (FLUJ[J+1, I+1]=0) AND (Y[J+1, I+1]=1) THEN
            BEGIN
              MOVETO(POINTS[I].X, POINTS[I].Y); IF (FLUJ[J+1, I+1]=1) THEN
                SETLINESTYLE(0, 0, ThickWidth); IF (FLUJ[J+1, I+1]=-1) THEN
                SETLINESTYLE(0, 0, NORMWidth); IF FLUJ[J+1, I+1]=0 THEN
                SETLINESTYLE(3, 0, NORMWidth); LINETO(POINTS[J].X, POINTS[J].Y);
              IF (I+1) >= (N/2) THEN
                OUTTEXTXY(PUNTITO[J+1, I+1].X-2, PUNTITO[J+1, I+1].Y-3, #17);
              IF (I+1) < (N/2) THEN
                OUTTEXTXY(PUNTITO[J+1, I+1].X+2, PUNTITO[J+1, I+1].Y-3, #16);
              END; END; END;
            END;
          END;
        END;
      END;
    END;
  END;

```

```

10:BEGIN
  IF I<J THEN
    BEGIN
      IF ((FLUJ[I+1,J+1]=1) OR (FLUJ[I+1,J+1]=-1))AND (Y[I+1,J+1]=1) THEN
        BEGIN
          MOVETO(POINTS[I].X, POINTS[I].Y); LINETO(POINTS[J].X, POINTS[J].Y);
          IF ((I+1) <(N/2)) AND (FLUJ[I+1,J+1]=1) THEN
            BEGIN
              OUTTEXTXY(PUNTITO[I+1,J+1].X-2,PUNTITO[I+1,J+1].Y-3,#17);
              OUTTEXTXY(PUNTITO[I+1,J+1].X+25,PUNTITO[I+1,J+1].Y+2,'(+)');END;
            IF ((I+1) <(N/2)) AND (FLUJ[I+1,J+1]=-1) THEN
              BEGIN
                OUTTEXTXY(PUNTITO[I+1,J+1].X-2,PUNTITO[I+1,J+1].Y-3,#16);
                OUTTEXTXY(PUNTITO[I+1,J+1].X+25,PUNTITO[I+1,J+1].Y+2,'(-)');END;
              IF ((I+1) >= (N/2)) AND (FLUJ[I+1,J+1]=1) THEN
                BEGIN
                  OUTTEXTXY(PUNTITO[I+1,J+1].X-1,PUNTITO[I+1,J+1].Y-3,#16);
                  OUTTEXTXY(PUNTITO[I+1,J+1].X+25,PUNTITO[I+1,J+1].Y+2,'(+)'); END;
                IF ((I+1) >= (N/2)) AND (FLUJ[I+1,J+1]=-1) THEN
                  BEGIN
                    OUTTEXTXY(PUNTITO[I+1,J+1].X-2,PUNTITO[I+1,J+1].Y-3,#17);
                    OUTTEXTXY(PUNTITO[I+1,J+1].X+25,PUNTITO[I+1,J+1].Y+2,'(-)');END;
                  END; END;
            IF (J>I) THEN
              BEGIN
                IF (FLUJ[J+1,I+1]=1) AND (Y[J+1,I+1]=1) THEN
                  BEGIN
                    MOVETO(POINTS[I].X, POINTS[I].Y); LINETO(POINTS[J].X, POINTS[J].Y);
                    IF ((I+1)>(N/2)) AND (FLUJ[J+1,I+1]=1) THEN BEGIN
                      OUTTEXTXY(PUNTITO[J+1,I+1].X-2,PUNTITO[J+1,I+1].Y-5,#17);
                      OUTTEXTXY(PUNTITO[J+1,I+1].X+25,PUNTITO[J+1,I+1].Y-8,'(+)');END;
                    IF ((I+1)>(N/2)) AND (FLUJ[J+1,I+1]=-1) THEN BEGIN
                      OUTTEXTXY(PUNTITO[J+1,I+1].X-2,PUNTITO[J+1,I+1].Y-5,#16);
                      OUTTEXTXY(PUNTITO[J+1,I+1].X+25,PUNTITO[J+1,I+1].Y-8,'(-)');END;
                    IF ((I+1)<=(N/2)) THEN
                      BEGIN
                        OUTTEXTXY(PUNTITO[J+1,I+1].X,PUNTITO[J+1,I+1].Y-3,#16);
                        OUTTEXTXY(PUNTITO[J+1,I+1].X+25,PUNTITO[J+1,I+1].Y-8,'(+)');
                      END;
                    IF ((I+1)<=(N/2)) AND (FLUJ[J+1,I+1]=-1) THEN BEGIN
                      OUTTEXTXY(PUNTITO[J+1,I+1].X-2,PUNTITO[J+1,I+1].Y-5,#17);
                      OUTTEXTXY(PUNTITO[J+1,I+1].X+25,PUNTITO[J+1,I+1].Y-8,'(-)');END;
                    END; END;END;
                  END;END;END;END; END;

```

Este procedimiento es el que grafica las digraficas correspondientes a las redes ya almacenadas o las capturadas por el usuario; este procedimiento también grafica las soluciones a cada iteración de las metodologías de flujo máximo y flujo máximo a costo mínimo.

```

PROCEDURE GRAFICAR(FLUJ:MAT);
BEGIN IF OPCION =1 THEN
  BEGIN
    INITIALIZE;
    MAINWINDOW(' Este grafo esta compuesto por 3 vertices y 3 aristas ');
    LINETOPLAY(FLUJ); SALIDA;END;
  IF (OPCION =2) OR (OPCION= 3) OR (OPCION=4) THEN
    BEGIN
      IF (OPCION=2) THEN
        BEGIN OPCION:=2; RES:=0; END;
      IF (OPCION= 3) OR (OPCION=4) THEN
        BEGIN RES:=OPCION; OPCION:=2; END;
      INITIALIZE; MAINWINDOW(' Grafo asociado ');
      LINETOPLAY(FLUJ);
      IF RES =3 THEN
        BEGIN
          OUTTEXTXY(50,10,'Arco Cap. ');
          OUTTEXTXY(140,10,'Cap. Costo');

```

CAPITULO IV. OPERACION DEL SISTEMA DE COMPUTO

```

OUTTEXTXY(60,20,' Min. '); OUTTEXTXY(120,20,'Max. ');
OUTTEXTXY(45,30,[1,2] 0'); OUTTEXTXY(135,30,'5 5');
OUTTEXTXY(45,40,[1,3] 0'); OUTTEXTXY(135,40,'7 4');
OUTTEXTXY(45,50,[2,4] 3'); OUTTEXTXY(135,50,'4 3');
OUTTEXTXY(45,60,[3,4] 0'); OUTTEXTXY(135,60,'6 8');
OUTTEXTXY(45,70,[4,5] 5'); OUTTEXTXY(135,70,'11 0');
END;
IF RES =4 THEN
BEGIN
OUTTEXTXY(55,20,'Arco Cap. '); OUTTEXTXY(155,20,'Flujo del arco');
OUTTEXTXY(75,30,[1,2] 5 4'); OUTTEXTXY(75,40,[1,3] 7 6');
OUTTEXTXY(75,50,[2,4] 4 4'); OUTTEXTXY(75,60,[3,4] 6 6');
OUTTEXTXY(75,70,[4,5] 10 10'); OUTTEXTXY(75,80,[5,1] 10 10');
END; SALIDA;
END;
IF OPCION=5 THEN
BEGIN
INITIALIZE; MAINWINDOW(' Grafo asociado a la solución' );
LINETOPLAY(FLUJ); OUTTEXTXY(65,10,'Arco Saturado');
OUTTEXTXY(65,20,'X[i,j]=B[i,j]'); SETLINESTYLE(0, 0, ThickWidth);
LINE(35,30,100,30); OUTTEXTXY(40,40,'Arco No');
OUTTEXTXY(110,40,'saturado'); OUTTEXTXY(45,50,'0<x[i,j]>');
OUTTEXTXY(115,50,'=B[i,j]'); SETLINESTYLE(0, 0, NORmWidth);
LINE(30,60,100,60); OUTTEXTXY(50,70,'Arco Nulo');
OUTTEXTXY(45,80,'x[i,j]=0'); SETLINESTYLE(3, 0, NORmWidth);
LINE(30,90,100,90); SALIDA; END;
IF OPCION=10 THEN
BEGIN
INITIALIZE; MAINWINDOW(' Grafo asociado al flujo que forma un ciclo' );
LINETOPLAY(FLUJ); SALIDA;
END;
PROCEDURE VALIDAR(W,VV:INTEGER);
BEGIN
ANDO:=0; OO:= " ";
REPEAT
GOTOXY(VV,W); READLN(OO); VAL(OO,ANDO,P);
IF P=1 THEN
BEGIN
GOTOXY(VV,W); WRITE(' '); COMPONER;
END;
IF (ANDO>=100) OR (ANDO<=-100) THEN
BEGIN
GOTOXY(VV,W); WRITE(' '); COMPONER;
END;
UNTIL (P=0) AND (ANDO<99) AND (ANDO >-99);
GOTOXY(18,22); WRITE(' ');
END;
PROCEDURE VALIDARI(W,VV:INTEGER);
BEGIN
ANDO:=0; OO:= " ";
REPEAT
GOTOXY(W,VV); READLN(OO); VAL(OO,ANDO,P);
IF (P=1) THEN
BEGIN
GOTOXY(W,VV); WRITE(' '); COMPONER;
END;
IF (ANDO>=16) OR (ANDO<3) THEN
BEGIN
GOTOXY(W,VV); WRITE(' '); COMPONER;
END;
UNTIL (P=0) AND (ANDO>=3) AND (ANDO<=15);
GOTOXY(18,22); WRITE(' ');
END;
PROCEDURE IMPRIME(AA:MAT);
BEGIN

```

## CAPITULO IV. OPERACION DEL SISTEMA DE COMPUTO

```

L:=0; KKK:=0;
GOTOXY(8,7);WRITE;
FOR I:=1 TO N DO
WRITE(1:5);
FOR I:=1 TO N DO
BEGIN
L:=10;KKK:=8+1; GOTOXY(6,KKK);WRITE(1:3);
FOR J:=1 TO N DO
BEGIN
GOTOXY(L,KKK); WRITE(AA[I,J]:3); L:=L+5;
END;
WRITELN;
END;
END;
PROCEDURE DATOS;
BEGIN
N:=8; FILLCHAR(Y,SIZEOF(Y),0); FILLCHAR(A,SIZEOF(A),0);
FILLCHAR(C,SIZEOF(C),0); FILLCHAR(B,SIZEOF(B),0);
Y[1,2]:=1; Y[1,3]:=1; Y[2,4]:=1; Y[2,5]:=1; Y[3,2]:=1; Y[3,4]:=1;
Y[3,5]:=1; Y[4,6]:=1; Y[4,7]:=1; Y[5,4]:=1; Y[5,6]:=1; Y[5,7]:=1;
Y[6,8]:=1; Y[7,6]:=1; Y[7,8]:=1; Y[8,1]:=1; A[2,5]:=3; A[4,7]:=5;
a[8,1]:=9; C[2,4]:=4; C[2,5]:=8; C[3,4]:=5; C[3,2]:=4; C[3,5]:=3;
C[4,6]:=5; C[4,7]:=7; C[5,4]:=1; C[5,6]:=2; C[5,7]:=6; C[7,6]:=4;
B[1,2]:=3; B[1,3]:=7; B[2,4]:=5; B[2,5]:=7; B[3,2]:=3; B[3,4]:=4;
B[3,5]:=2; B[4,6]:=7; B[4,7]:=6; B[5,4]:=2; B[5,6]:=3; B[5,7]:=1;
B[6,8]:=5; B[7,6]:=4; B[7,8]:=4; B[8,1]:=9; OPCION:=2;
END;

```

El procedimiento Diime es donde se captura la matriz de adyacencia de la red, así como, los costos involucrados y las capacidades mínimas y máximas, y se almacenan en un archivo el cual es asignado por el usuario.

```

PROCEDURE diime; (**PROCEDIMIENTO QUE PIDE LOS VALORES NECESARIOS**)
BEGIN
GOTOXY(20,22);WRITE('CUAL ES EL NOMBRE DEL ARCHIVO ');
READLN(SUN); ASSIGN(ARA,SUN); VENTA;
REWRITE(ARA); YEA:=0; VV:=72; W:=6;
GOTOXY(5,6); WRITE('INTRODUZCA EL NUMERO DE NODOS EXISTENTES EN LA GRAFICA: ');
VALIDAR(VV,W);N:=ANDO; WRITELN(ARA,N); WRITELN; WRITELN(ARA);
GOTOXY(5,8);WRITELN(' SI NO EXISTE UNA ARCO DIRIGIDO DE "I" A "J", ESCRIBE UN 0;');
GOTOXY(5,9);WRITELN('Y SI EXISTE UN ARCO DIRIGIDO ENTRE ESTOS NODOS ESCRIBE UN 1');
WRITELN; WRITELN;
FOR I:=1 TO N DO
BEGIN
FOR J:=1 TO N DO
BEGIN
IF I>=2 THEN
BEGIN
GOTOXY(10,5+J);WRITE('ARCO ENTRE EL NODO ',I,' Y EL NODO ',J,': ');
W:=5+J; AFIRMAR(YEA,W); Y[I,J]:=YEA;
END;
IF I<2 THEN
BEGIN
GOTOXY(10,11+J);WRITE('ARCO ENTRE EL NODO ',I,' Y EL NODO ',J,': ');
W:=11+J; AFIRMAR(YEA,W); Y[I,J]:=YEA;
END;
WRITELN(ARA,Y[I,J]);
END;
ENTERR; VENTA;
END;
WRITELN(ARA); TT:=0;
FOR I:=1 TO N DO
BEGIN VENTA; VV:=67;
FOR J:=1 TO N DO
BEGIN

```





CAPITULO IV. OPERACION DEL SISTEMA DE COMPUTO

```

WRITE(' V1');
FOR I:= 1 TO N DO
WRITE(V[I]:6, ' '); WRITE(' '); WRITELN;
IF N= 3 THEN
WRITELN(' |');
IF N=4 THEN
WRITELN(' |');
IF N=5 THEN
WRITELN(' |');
IF N=6 THEN
WRITELN(' |');
IF N=7 THEN
WRITELN(' |');
IF N=8 THEN
WRITELN(' |');
IF N=9 THEN
WRITELN(' |');
WRITE(' U1');
FOR I:= 1 TO N DO
BEGIN
WRITE(U[I-1]:6, ' ');
END;
WRITE(' ');
WRITELN;
IF N=3 THEN
BEGIN
WRITELN('V1 U1 |');
WRITELN('é1é11 ');
END;
IF N=4 THEN
BEGIN
WRITELN('V1 U1 |');
WRITELN('é1é11 ');
END;
IF N=5 THEN
BEGIN
WRITELN('V1 U1 |');
WRITELN('é1é11 ');
END;
IF N=6 THEN
BEGIN
WRITELN('V1 U1 |');
WRITELN('é1é11 ');
END;
IF N=7 THEN
BEGIN
WRITELN('V1 U1 |');
WRITELN('é1é11 ');
END;
IF N=8 THEN
BEGIN
WRITELN('V1 U1 |');
WRITELN('é1é11 ');
END;
IF N=9 THEN
BEGIN
WRITELN('V1 U1 |');
WRITELN('é1é11 ');
END;
IF N>9 THEN
BEGIN
WRITELN('V1 U1 |');
WRITELN('é1é11 ');
END;
END;
PROCEDURE IMPRIME1(X:MAT); {IMPRIME LA TABLA*****}

```

## CAPITULO IV. OPERACION DEL SISTEMA DE COMPUTO

```

BEGIN
D:=0; DD:=0; DDD:=0;
IF N <=9 THEN
BEGIN
BEGIN
FOR I:=1 TO N DO
BEGIN
FOR J:=1 TO N DO
BEGIN
IF Y[I,J]=1 THEN
BEGIN
D:=2+(J*7)+D; DD:=5+I*2; DDD:=D+3;
IF D=DDD THEN
DDD:=D+3; GOTOXY(D,DD);WRITE(AA[I,J]:3);
GOTOXY(DDD,DD);WRITE(Z[I,J]:3);
GOTOXY(D,DD+1); WRITE(ZZ[I,J]:3);
GOTOXY(DDD,DD+1); WRITE(CC[I,J]:3);
D:=0; DDD:=3;
END;
END;
END;
END;
D:=0; DD:=0; DDD:=0;
END;
IF N>9 THEN
BEGIN
venta;
GOTOXY(15,5);WRITE('MATRIZ DE FLUJO EXISTENTE XIJ');
IMPRIME(AA); ENTERR; venta; GOTOXY(15,5);WRITE('MATRIZ XIJ-BIJ');
IMPRIME(Z); ENTERR; venta; GOTOXY(15,5);WRITE('MATRIZ XI-AIJ');
IMPRIME(ZZ); ENTERR; venta; GOTOXY(15,5);WRITE('MATRIZ DE COSTOS REDUCIDOS CIJ');
IMPRIME(CC); ENTERR; venta;
venta;
END;
END;
IF N=9 THEN
BEGIN
WRITELN(' É!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!');
TABLA; TABLITA; IMPRIMEI(X);
END;
IF N>9 THEN
BEGIN IMPRIMEI(X);
GOTOXY(5,4);TABLA;
END;
END;
PROCEDURE UU(X:MAT;U:VECTORX);
BEGIN
D:=0; FILLCHAR(Z,SIZEOF(Z),0); FILLCHAR(ZZ,SIZEOF(ZZ),0); FILLCHAR(V,SIZEOF(V),0);
FOR I:=1 TO N DO
V[I]:=1;
FOR I:=1 TO N DO
BEGIN
FOR J:=1 TO N DO
BEGIN
IF Y[I,J]=1 THEN
BEGIN
Z[I,J]:=X[I,J]-B[I,J]; ZZ[I,J]:=X[I,J]-A[I,J];
CC[I,J]:=C[I,J]-U[I-1]+U[J-1];
END;
END;
END;
END;
IF N=3 THEN
BEGIN
WRITELN(' É!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!');
TABLA; TABLITA; IMPRIMEI(X);
END;

```



## CAPITULO IV. OPERACION DEL SISTEMA DE COMPUTO

```
GOTOXY(17,9+1);WRITE('');  
END;
```

El procedimiento que se presenta a continuación es donde se programa el método de las desviaciones.

```
PROCEDURE RETORNO(X,Y,FLUJ:MAT;BETA:VECTORX);  
BEGIN  
VENTA; S:=0;T:=0;NADA:=0; PAR:=0;COSTO:=0; FILLCHAR(NCON,SIZEOF(NCON),0);CUENTA:=0;  
GOTOXY(1,3); WRITE(' A continuaci, #162,n se presenta la tabla donde ');  
GOTOXY(1,4); WRITE(' se muestra a que conjunto pertenece cada arco. ');  
GOTOXY(1,7);  
FOR I:=1 TO N DO  
BEGIN {BEGIN de i}  
FOR J:=1 TO N DO  
BEGIN {BEGIN de j}  
IF Y[I,J]=1 THEN  
BEGIN {BEGIN de arco}  
IF CC[I,J]>0 THEN  
BEGIN {BEGIN de l}  
IF X[I,J]=A[I,J] THEN  
BEGIN  
WRITELN(' El arco ',I,', ',J, ' es Conformable y pertenece a L');  
NCON[I,J]:=0;  
END;  
IF X[I,J]<A[I,J] THEN  
BEGIN  
WRITELN(' El arco ',I,', ',J, ' No es Conformable y pertenece a L1');  
NCON[I,J]:=1;  
END;  
IF X[I,J]>A[I,J] THEN  
BEGIN  
WRITELN(' El arco ',I,', ',J, ' No es Conformable y pertenece a L2');  
NCON[I,J]:=-1;  
END;  
END; {END de l}  
IF CC[I,J] = 0 THEN  
BEGIN {BEGIN de b}  
IF (A[I,J]=X[I,J]) OR ((X[I,J]>A[I,J]) AND (X[I,J]<B[I,J])) OR (X[I,J]=B[I,J]) THEN  
BEGIN  
WRITELN(' El arco ',I,', ',J, ' Es Conformable y pertenece a B');  
NCON[I,J]:=0;  
END;  
IF X[I,J]<A[I,J] THEN  
BEGIN  
WRITELN(' El arco ',I,', ',J, ' No es Conformable y pertenece a B1');  
NCON[I,J]:=2;  
END;  
IF X[I,J]>B[I,J] THEN  
BEGIN  
WRITELN(' El arco ',I,', ',J, ' No es Conformable y pertenece a B2');  
NCON[I,J]:=-2;  
END;  
END; {END de b}  
IF CC[I,J]<0 THEN  
BEGIN {BEGIN de c}  
IF X[I,J]=B[I,J] THEN  
BEGIN  
WRITELN(' El arco ',I,', ',J, ' Es Conformable y pertenece a K');  
NCON[I,J]:=0;  
END;  
IF X[I,J]<B[I,J] THEN  
BEGIN  
WRITELN(' El arco ',I,', ',J, ' No es Conformable y pertenece a K1');  
NCON[I,J]:=3;  
END;  
END;
```

## CAPITULO IV. OPERACION DEL SISTEMA DE COMPUTO

```

IF X[I,J]>B[I,J] THEN
  BEGIN
    WRITELN(' El arco ',I,', ',J,', ' No es Conformable y pertenece a K2');
    NCON[I,J]:=-3;
    END;
  END;{END de c}
END;{END de arco}
END;{END de j}
END;{END de i}
ENTERR;
FOR I:=1 TO N DO
  BEGIN
    FOR J:=1 TO N DO
      BEGIN
        IF Y[I,J]=1 THEN
          BEGIN
            IF NCON[I,J]=1 THEN NADA:=NADA+1;
            IF NCON[I,J]=-1 THEN NADA:=NADA+1;
            IF NCON[I,J]=2 THEN NADA:=NADA+1;
            IF NCON[I,J]=-2 THEN NADA:=NADA+1;
            IF NCON[I,J]=3 THEN NADA:=NADA+1;
            IF NCON[I,J]=-3 THEN NADA:=NADA+1;
            END;
          END;
        END;
      IF NADA>0 THEN
        BEGIN
          VENTA; UU(X,U);
          IF (NCON[N,1]=1) OR (NCON[N,1]=2) OR (NCON[N,1]=3) THEN
            BEGIN
              S:=1; T:=N;
              END;
            IF NCON[N,1]=0 THEN
              BEGIN
                FOR I:=1 TO N DO
                  BEGIN
                    FOR J:=1 TO N DO
                      BEGIN
                        IF ((NCON[I,J]=1) OR (NCON[I,J]=2) OR (NCON[I,J]=3)) THEN
                          BEGIN
                            S:=J; T:=1;
                            END;
                          IF ((NCON[I,J]=-1) OR (NCON[I,J]=-2) OR (NCON[I,J]=-3)) THEN
                            BEGIN
                              S:=1; T:=J;
                              END;
                            END;
                          END;
                        END;
                      END;
                    ETI(N);
                    BEGIN {BEGIN de etiquetacion}
                      FILLCHAR(BETA,SIZEOF(BETA),0); FILLCHAR(FLUJ,SIZEOF(FLUJ),0);
                      BETA[S]:=99; TTI:=99; TT2:=99; TETA:=0; NV:=0; SV:=0;
                      IF (Y[T,S]=1) AND ((BETA[J]=0) OR (BETA[J]=0)) THEN
                        BEGIN
                          IF ((CC[T,S]>=0) AND (X[T,S]<A[T,S])) OR ((CC[T,S]<0) AND (X[T,S]<B[T,S])) THEN
                            IF ((CC[T,S]<=0) AND (X[T,S]>B[T,S])) OR ((CC[T,S]>0) AND (X[T,S]>A[T,S])) THEN
                              BEGIN END;
                            BEGIN {BEGIN principal para empezar etiquetacion}
                              FOR I:=1 TO N DO
                                BEGIN {BEGIN del i}
                                  FOR J:=1 TO N DO
                                    BEGIN {BEGIN de j}
                                      IF ((Y[I,J]=1) AND (BETA[i]>0) AND (BETA[j]=0)) OR ((Y[J,I]=1) AND (BETA[j]=0) AND (BETA[i]=0)) THEN
                                        BEGIN
                                          IF (CC[I,J]>0) AND (X[I,J]<A[I,J]) THEN

```

## CAPITULO IV. OPERACION DEL SISTEMA DE COMPUTO

```

BEGIN
  BETA[J]:=A[I,J]-X[I,J];
  IF BETA[J]>BETA[I] THEN
    BETA[J]:=BETA[I];
    IF (BETA[J]>0) THEN
      BEGIN
        GOTOXY(JA,5+(J*2));WRITELN('I,J,',BETA[J,]);
        FLUJ[I,J]:=1;
      END;
    END;
  IF (CC[I,J]<=0) AND (X[I,J]<B[I,J]) THEN
    BEGIN
      BETA[J]:=B[I,J]-X[I,J];
      IF BETA[J]>BETA[I] THEN
        BETA[J]:=BETA[I];
        IF (BETA[J]>0) THEN
          BEGIN
            GOTOXY(JA,5+(J*2)); WRITELN('I,J,',BETA[J,]);
            FLUJ[I,J]:=1;
          END;
        END;
      IF (I>J) AND ((BETA[J]>0) AND (BETA[I+1]=0)) THEN
        BEGIN
          I:=J; J:=I;
        END;
      END;{END de Y[I,J]}
    END;{END de j}
  END;{END de i}
END;{END de empezar etiquettacion}
END;{END si es Conformable N,I}
RESPU:=0;
IF BETA[T]=0 THEN
  BEGIN
    RESPU:=1;
    FOR I:=1 TO N DO
      BEGIN
        FOR J:=1 TO N DO
          IF (BETA[I]>0) AND (BETA[I]<>BETA[S]) AND (Y[I,J]=Y[I,I]) AND (X[I,I]>0) THEN
            BEGIN RIN:=I; I:=N;END;
          END;
        FOR I:=1 TO N DO
          BEGIN
            IF (BETA[I]>0) AND (I<>S) THEN
              BEGIN
                RIN:=I; I:=N;
              END;
            END;
          FOR J:=1 TO N DO
            BEGIN
              IF (Y[J,RIN]=1) AND (BETA[J]=0) THEN
                BEGIN
                  {<}
                  IF (CC[J,RIN]>=0) AND (X[J,RIN]>A[J,RIN]) THEN
                    BEGIN
                      BETA[J]:=X[J,RIN];
                      IF BETA[J]>BETA[RIN] THEN
                        BETA[J]:=BETA[RIN];
                      IF (BETA[J]>0) THEN
                        BEGIN
                          GOTOXY(JA,5+(J*2));WRITELN('I,RIN,',BETA[J,]);
                          FLUJ[J,RIN]:=-1;
                        END;
                      END;
                    IF (CC[J,RIN]<0) AND (X[J,RIN]>B[J,RIN]) THEN
                      BEGIN
                        BETA[J]:=X[I,RIN];
                        IF BETA[J]>BETA[RIN] THEN

```

```

BETA[J]:=BETA[RIN];
IF BETA[J]<BETA[RIN] THEN
BETA[J]:=BETA[RIN];
IF (BETA[J]>0) THEN
BEGIN
GOTOXY(JA,5+(J*2)); WRITELN('RIN;',BETA[J],');
FLUJ[J,RIN]:=-1;
END;
END;
END; END; END;
ENTERR;
IF BETA[T]>0 THEN
RESPU:=0;
IF ((NCON[T,S]=1) OR (NCON[T,S]=2) OR (NCON[T,S]=3)) THEN
BEGIN
FLUJ[T,S]=1;
END;
IF ((NCON[T,S]=-1) OR (NCON[T,S]=-2) OR (NCON[T,S]=-3)) THEN
BEGIN
FLUJ[T,S]=-1;
END;
OPCION:=10;
FOR I:=1 TO N DO
BEGIN FOR J:=1 TO N DO
BEGIN
IF (FLUJ[I,J]=1) OR (FLUJ[I,J]=-1) THEN
CUENTA:=CUENTA+1;
END; END; GRAFCAR(FLUJ);
IF CUENTA = 0 THEN
RESPU:=1;
VENTA;
CASE RESPU OF
0:BEGIN
GOTOXY(10,3);WRITE('Por favor, si el arco no se encuentra en el circuito ');
GOTOXY(10,4);WRITE('incerta las coordenadas a continuacion. ');
GOTOXY(10,7);WRITE('Cuantos arcos no pertenecen al circuito ? ');
REPEAT;
IM:=7; W:=0; DIG:=6;
DIGITAR(CEU,DIG,IM);
IF (CEU<0) OR (CEU>CUENTA) THEN
COMPONER;
UNTIL (CEU<=CUENTA) AND (CEU>-1);
GOTOXY(10,8);IF CEU >0 THEN
WRITELN('Dame las coordenadas: ');
FOR I:=1 TO CEU DO
BEGIN
CONTAMOS(I);
IF Y[JK,LM]=1 THEN
FLUJ[JK,LM]=0;
IF Y[JK,LM]=0 THEN
BEGIN
COMPONER; CONTAMOS(I);
END;END;
IF ( NCON[T,S]=1) OR (NCON[T,S]=2) THEN
BEGIN
IF (A[T,S]-X[T,S])<=BETA[T] THEN
BETA[T]:=A[T,S]-X[T,S];
IF (A[T,S]-X[T,S])>BETA[T] THEN
WRITELN;
END;
IF (NCON[I,J]=-1) THEN
BEGIN
IF (CC[T,S])*(X[T,S]-B[T,S])<=BETA[T] THEN
BETA[T]:=CC[T,S]*(X[T,S]-A[T,S]);
IF CC[T,S]*(X[T,S]-A[T,S])>BETA[T] THEN

```

## CAPITULO IV. OPERACION DEL SISTEMA DE COMPUTO

```

WRITELN;
END;
IF (NCON[T,S]=-2) OR (NCON[T,S]=-3) THEN
BEGIN
IF (X[T,S]-B[T,S])>BETA[T] THEN
WRITELN;
END;
IF (NCON[T,S]=3) THEN
BEGIN
IF (CC[T,S])*(X[T,S]-B[T,S])<=BETA[T] THEN
BETA[T]=CC[T,S]*(X[T,S]-B[T,S]);
IF CC[T,S]*(X[T,S]-B[T,S])>BETA[T] THEN
WRITELN;
END;
WRITELN;
FOR I:=1 TO N DO
BEGIN
FOR J:=1 TO N DO
BEGIN
IF (Y[I,J]=1) THEN
BEGIN
IF (FLUJ[I,J]=1) AND (BETA[J]>0) THEN
BEGIN
X[I,J]:=X[I,J]+BETA[T];
WRITELN(' x [I,J]=',X[I,J]:4);
END;
IF (FLUJ[I,J]=-1) AND (BETA[J]>0) THEN
BEGIN
X[I,J]:=X[I,J]-BETA[T];
WRITELN(' x [I,J]=',X[I,J]:4);
END;
IF (FLUJ[J,I]=-1) AND (BETA[J]>0) THEN
BEGIN
X[J,I]:=X[J,I]-BETA[T];
END;
END;{END DE Y}
END;{END de j}
END;{END de i}
ENTERR;
FLUJ[S,T]:=1; OPCION:=10;
GRAFICAR(FLUJ);
FILLCHAR(BETA,SIZEOF(BETA),0); FILLCHAR(FLUJ,SIZEOF(FLUJ),0);
BETA[I]:=99;
RETORNO(X,Y,FLUJ,BETA);
END;{DEL OPCION 0}
I:BEGIN
GOTOXY(15,4);WRITE(' CAMBIO DE POTENCIALES');
GOTOXY(10,7);WRITE('Vertices etiquetados (');
FOR I:=1 TO N DO
BEGIN
IF BETA[I]>0 THEN WRITE(' I:2);
END;
WRITE(')'); GOTOXY(10,9);WRITE('Vertices No etiquetados (');
FOR I:=1 TO N DO
BEGIN
IF (BETA[I]<0) OR (BETA[I]=0) THEN WRITE(' I:2);
END;
WRITE(')');
GOTOXY(10,11);WRITE(' SI = { ');
FOR I:=1 TO N DO
BEGIN
FOR J:=1 TO N DO
BEGIN
IF Y[I,J]=1 THEN
BEGIN
IF (BETA[I]>0) AND (BETA[J]=0) AND (CC[I,J]>0) AND (X[I,J]<=B[I,J]) THEN

```

CAPITULO IV. OPERACION DEL SISTEMA DE COMPUTO

```

BEGIN
WRITE('I,J,J,J'); SV:=SV+1;
IF CC[I,J]<TT1 THEN
TT1:=CC[I,J];
END;
END;
END;
END;
WRITE(' '); GOTOXY(10,13);WRITE(' S2 = { ');
FOR I:=1 TO N DO
BEGIN
FOR J:=1 TO N DO
BEGIN
IF Y[I,J]=1 THEN
BEGIN
IF (BETA[I]=0) AND (BETA[J]>0) AND (CC[I,J]<0) AND (X[I,J]>=A[I,J]) THEN
BEGIN
WRITE('I,J,J,J '); NV:=NV+1;
IF (CC[I,J]*(-1)<TT2 THEN
TT2:=CC[I,J]*(-1);
END;
END;
END;
END;
WRITE(' ');
IF SV=0 THEN
BEGIN
GOTOXY(10,14);WRITE('El conjunto S1 es vacio');END;
IF NV=0 THEN
BEGIN
GOTOXY(10,15);WRITE('El conjunto S2 es vacio');
END;
IF (SV=0) AND (NV=0) THEN
BEGIN
GOTOXY(10,16);WRITE('EL PROBLEMA NO TIENE SOLUCION');
END;
IF SV>0 THEN
BEGIN
GOTOXY(10,17);WRITE('El minimo del conjunto S1 es :',TT1:4);
END;
IF NV>0 THEN
BEGIN
GOTOXY(10,18);WRITE('El minimo del conjunto S2 es :',TT2:4);
END;
IF (NV>0) AND (SV>0) THEN
BEGIN
IF TT1>TT2 THEN
TETA:=TT2;
IF TT1<=TT2 THEN
TETA:=TT1;
END;
IF (NV>0) OR (SV>0) THEN
BEGIN
IF (NV>0) AND (SV=0) THEN
TETA:=TT2;
IF (SV>0) AND (NV=0) THEN
TETA:=TT1;
END;
IF ((NV>0) AND (SV>0)) OR ((NV>0) OR (SV=0)) OR ((NV=0) OR (SV>0)) THEN
BEGIN GOTOXY(10,20);WRITELN('El valor de Teta es igual a: ', TETA:4);
END;
IF ((NV=0) AND (SV=0)) THEN
BEGIN GOTOXY(10,20);WRITELN('El valor de Teta es igual a: ',#236);
END;
ENTERR;
FOR I:=1 TO N DO

```



```

I:INTEGER;
LABEL 5,10;
BEGIN
ACA:=0; VENTA;GOTOXY(15,4);WRITE('VALORES DE GAMA E INCREMENTO DEL FLUJO');
GOTOXY(5,5);WRITELN(' la corte inicial es ',CMAX,' y la corte final es ',A[N,1]);
GOTOXY(1,7);
10:BEGIN
K:=1; FILLCHAR(FLUJ,SIZEOF(FLUJ),0);FILLCHAR(GAMA,SIZEOF(GAMA),0);
FILLCHAR(KK,SIZEOF(KK),0); GAMA[1]:=99; GAMA[0]:=99;
5:BEGIN
ACA:=0;
FOR I:= 1 TO N DO
BEGIN
ASI:=B[K,i]-X[K,i];
IF (Y[K,i]=1) AND (ACA<=ASI) AND (K<>I) THEN
BEGIN
ACA:= B[K,i]-X[K,i];
KK:=i; END;
IF (X[KK,N]=B[KK,N]) AND (Y[KK,N]=1) THEN
BEGIN
KK:=KKK;
END;
KKK:=KK;
END;
GAMA[KKK]:=B[K,KKK]-X[K,KKK];
IF K<>KKK THEN
BEGIN
IF (GAMA[KKK]>= GAMA[K]) AND (GAMA[KKK]>0) THEN
BEGIN
GAMA[KKK]=GAMA[K];
WRITELN(' GAMA[',KKK,'] : ',GAMA[KKK]);
FLUJ[K,KKK]:=1;
END;
IF (GAMA[KKK]<GAMA[K]) AND (GAMA[KKK]>0) THEN
BEGIN
WRITELN(' GAMA[',KKK,'] : ',GAMA[KKK]);
FLUJ[K,KKK]:=1;
END;
K:=KKK;
END;
END;
IF (K<N) AND (GAMA[KKK]>0) THEN GOTO 5;
IF (K=N) AND (GAMA[KKK]>0) THEN
BEGIN
FOR I:=1 TO N DO
BEGIN
FOR J:=1 TO N DO
BEGIN
IF (Y[I,J]= 1) THEN
BEGIN
IF (FLUJ[I,J]= 1) AND (GAMA[J]<0) THEN
BEGIN
X[I,J]:=X[I,J]+ GAMA[N];
WRITELN(' x [',I,',',J,'] : ',X[I,J]);
END;
IF (FLUJ[I,J]= -1) AND (GAMA[J]<0) THEN
BEGIN
X[I,J]:=X[I,J] - GAMA[N];
END;
END;
END;
END;
X[N,1]:=X[N,1]+GAMA[N];
ENTERR; FLUJ[N,1]:=1;
OPCION:=10; GRAFICAR(FLUJ);
END;

```

## CAPITULO IV. OPERACION DEL SISTEMA DE COMPUTO

```

IF (X[N,1]<B[N,1]) AND (GAMA[KKK]>0) THEN
BEGIN
VENTA; GOTOXY(15,4); WRITE('VALORES DE GAMA E INCREMENTO DEL FLUJO');
GOTOXY(1,7); GOTO 10;
END;
END;
IF (X[N,1]=B[N,1]) OR (GAMA[KKK]=0) THEN
BEGIN
IF ((KKK=K) AND (GAMA[KKK]=0)) THEN
BEGIN
WRITELN; WRITELN(' No fue posible llegar a sumidero ');
ENTERR;
END;
BEGIN
VENTA; GOTOXY(15,4); WRITELN('El Flujo M aximo es: ',x[N,1]);
IMPRIME(X); GOTOXY(7,20);
WRITE('Como deseas que se llame el archivo de resultados ');GOTOXY(65,20);
READLN(SUN); ASSIGN(SOL,SUN); FILLCHAR(FLUJ,SIZEOF(FLUJ),0);REWRITE(SOL);
WRITELN(SOL,N);
FOR I:=1 TO N DO
BEGIN
FOR J:=1 TO N DO
BEGIN
IF X[I,J]>0 THEN
BEGIN
IF X[I,J]=B[I,J] THEN FLUJ[I,J]:=1;
IF (X[I,J]>0) AND (X[I,J]<B[I,J]) THEN FLUJ[I,J]:=-1;
END;
WRITELN(SOL,X[I,J]);WRITELN(SOL,B[I,J]);WRITELN(SOL,Y[I,J]);
END;
END;
ENTERR; CLOSE(SOL);OPCION:=5;
GRAFICAR(FLUJ);END;
END;
END;
PROCEDURE FLUJITO;

```

Este procedimiento calcula el flujo máximo de la red que se escoga.

```

BEGIN
I:=0; FILLCHAR(X,SIZEOF(X),0); FILLCHAR(FLUJ,SIZEOF(FLUJ),0); FILLCHAR(GAMA,SIZEOF(GAMA),0);
FILLCHAR(KK,SIZEOF(KK),0); GAMA[1]:=99; GAMA[0]:=99; CMIN:=0; CMAX:=0; K:=0; ACA:=0;
FOR J:=1 TO N DO
BEGIN
IF (Y[1,J]=1) OR (Y[J,N]=1) THEN
BEGIN
CMIN:=B[1,J]+CMIN; CMAX:=B[J,N]+CMAX;
IF ACA<=B[1,J] THEN
BEGIN
ACA:=B[1,J]; K:=J; END;
END; END;
IF CMIN<=CMAX THEN
A[N,1]:=CMIN;
IF CMIN>CMAX THEN
A[N,1]:=CMAX;
B[N,1]:=A[N,1];
K:=1;
CAL_K(K,KK,KKK,GAMA);
END;
PROCEDURE INICIO;
BEGIN
FILLCHAR(I,SIZEOF(I),0);FILLCHAR(J,SIZEOF(J),0);FILLCHAR(x,SIZEOF(x),0);
FILLCHAR(a,SIZEOF(a),0);FILLCHAR(b,SIZEOF(b),0);FILLCHAR(C,SIZEOF(C),0);
FILLCHAR(CC,SIZEOF(CC),0);FILLCHAR(u,SIZEOF(u),0);FILLCHAR(AA,SIZEOF(AA),0);
KK:=0;NN:=0; Nada:=0;RES:=0;CMIN:=0; CMAX:=0; K:=0;ACA:=0;
END;
PROCEDURE GRAFI_CA;

```

```
PROCEDURE LLECE;
BEGIN
FILLCHAR(Y,SIZEOF(Y),0);
FILLCHAR(FLUJ,SIZEOF(FLUJ),0);
FILLCHAR(x,SIZEOF(x),0);
END;
PROCEDURE GRAFI_CA;
BEGIN
VENTA; AYUD;
ENTERR; N:=3; Y[1,2]:=1; Y[2,3]:=1;
Y[3,1]:=1; GRAFICAR(FLUJ);
END;
PROCEDURE PASEO;
BEGIN
VENTA; PASS; ENTERR;
N:=3; Y[1,2]:=1; Y[2,3]:=1; GRAFICAR(FLUJ); END;
PROCEDURE CIRCUITO; {*****DEFINICION *****}
BEGIN VENTA; CIRC;
ENTERR; FILLCHAR(y,SIZEOF(y),0); N:=4; Y[1,2]:=1;
Y[2,3]:=1; Y[3,1]:=1; Y[3,4]:=1;
GRAFICAR(fluj);
END;

PROCEDURE DIGRAFI_CA; {*****DEFINICION *****}
BEGIN
VENTA; DIGR; ENTERR;
FILLCHAR(y,SIZEOF(y),0); N:=5; Y[1,2]:=1;
Y[1,3]:=1; Y[2,4]:=1; Y[3,2]:=1;
Y[3,4]:=1; Y[4,5]:=1;
GRAFICAR(fluj);
END;

PROCEDURE RED; {***** DEFINICION *****}
BEGIN
VENTA; REDD; ENTERR; N:=5;
FILLCHAR(y,SIZEOF(y),0); Y[1,2]:=1;
Y[1,3]:=1; Y[2,4]:=1;
Y[3,2]:=1; Y[3,4]:=1; Y[4,5]:=1;
GRAFICAR(fluj);
END;

PROCEDURE FLU_EJE;
BEGIN
VENTA; FLU; ENTERR; N:=5;
FILLCHAR(y,SIZEOF(y),0); Y[1,2]:=1; Y[1,3]:=1;
Y[2,4]:=1; Y[3,2]:=1; Y[3,4]:=1;
Y[4,5]:=1; Y[5,1]:=1;
GRAFICAR(FLUJ);
END;

PROCEDURE IMPMAT;
BEGIN
VENTA; PRINT1; IMPRIME(C); ENVENTA;
PRINT2; IMPRIME(A);
ENVENTA; GOTOXY(10,2);
PRINT3; IMPRIME(B); ENTERR;
END;

PROCEDURE CAP;
BEGIN
IF (SUN="") OR (SUN=SAN) THEN
SUN:=SAN; INICIO;
ASSIGN(ARA,SUN);
RESET(ARA);
READ(ARA,N);
READLN(ARA); INICIO;
FOR I:=1 TO N DO
```



## CAPITULO IV. OPERACION DEL SISTEMA DE COMPUTO

```

3:BEGIN C[I,J]:=AA[I,J]; END;
END;
END;
END; END;
IF N=0 THEN BEGIN
GOTOXY(10,21);WRITELN('El Número de vértices de tu archivo es Nulo');
END; ENTERR;
END;
PROCEDURE FUNCION;
BEGIN
D:=7; VENTA;GOTOXY(15,2); WRITE(' MODELO PRIMAL DE LA RED ');
GOTOXY(5,4);WRITE(' MIN Z =');
FOR I:=1 TO N-1 DO
BEGIN
FOR J:=1 TO N DO
BEGIN
IF Y[I,J]= 1 THEN
WRITE('+',C[I,J]);C(X,I,J);
END;
END;
READLN; WRITELN(' s.a. ');
FOR I:=1 TO N DO
BEGIN
FOR J:=1 TO N DO
BEGIN
IF Y[I,J]=1 THEN
BEGIN
WRITE(' X',I,J);
FOR L:=1 TO N DO
BEGIN
IF (Y[L,J]=1) AND (L<>I) THEN
WRITE(' + X',L,J);
END;
FOR K:=1 TO N DO
BEGIN
IF Y[J,K]=1 THEN
WRITE(' - X',J,K);
END;
WRITE(' = 0'); D:=D+1; WRITELN;
IF D=21 Then
BEGIN
ENTERR; VENTA; D:=3; GOTOXY(1,3);
END; END;
END;
END;
D:=D+1;
FOR I:=1 TO N DO
BEGIN
FOR J:=1 TO N DO
BEGIN
IF Y[I,J]=1 THEN
BEGIN
GOTOXY(10,D); WRITE('X',I,J,'>=',A[I,J];2);
GOTOXY(30,D); WRITE(' - X',I,J,'>=',B[I,J];2); D:=D+1;
END;
IF D= 23 THEN
BEGIN
ENTERR; VENTA; D:=3;
END;
END; END;
GOTOXY(20,D); WRITE(' X(I,J) i Z'); ENTERR;
END;
PROCEDURE METODOS;
BEGIN (***** BEGIN PRINCIPAL***** )
VENTA;I:=0;ASSIGN(ARA,SUN); GOTOXY(23,4);
WRITE(' MENU DE METODOS ');GOTOXY(15,8);
WRITE(' ');GOTOXY(15,10);WRITE(' 1.- FLUJO MAXIMO DE LA RED ');
GOTOXY(15,12);WRITE(' 2.- DIGRAFICA DE LA RED ');
GOTOXY(15,14);WRITE(' 3.- FLUJO MAXIMO A COSTO MINIMO EN UNA RED ');

```

## CAPITULO IV. OPERACION DEL SISTEMA DE COMPUTO

```
GOTOXY(15,16);WRITE(' 4.- MODELO PRIMAL ');
GOTOXY(15,18);WRITE(' 5.- MENU PRINCIPAL ');
GOTOXY(14,20);WRITE(' DAME LA OPCION DESEADA : ');
REPEAT
DIG:=2;
DIGITAR(OPCION,DIG,J);
IF (OPCION<1) OR (OPCION>5) THEN
COMPONER;
UNTIL (OPCION<6) AND (OPCION>0);
WRITELN;
ENTERR;
CASE OPCION OF
1:BEGIN
INICIO; GOTOXY(22,22);WRITE('CUAL ES EL NOMBRE DEL ARCHIVO ');
READLN(SUN); CAP; FLUJITO; METODOS; END;
2:BEGIN
INICIO; GOTOXY(22,22);WRITE('CUAL ES EL NOMBRE DEL ARCHIVO ');
READLN(SUN); CAP; GRAFICAR(FLUJ); METODOS; END;
3:BEGIN
INICIO; GOTOXY(22,22);WRITE('CUAL ES EL NOMBRE DEL ARCHIVO ');
READLN(SUN); CAP;
FOR J:=1 TO N do
BEGIN
IF (Y[1,J]=1) OR (Y[J,N]=1) THEN
BEGIN
CMIN:=B[1,J]+CMIN; CMAX:=b[j,N]+CMAX;
IF ACA<=B[1,J] THEN
BEGIN
ACA:=B[1,J]; K:=j; END;
END; END;
IF CMIN<=CMAX THEN
A[N,1]:=CMIN;
IF CMIN>CMAX THEN
A[N,1]:=CMAX; B[N,1]:=a[N,1];
VENTA; GOTOXY(15,6); WRITE('VALORES DE U');
GOTOXY(10,9); WRITE(' 0. Ui GENERADO POR EL PROGRAMA ');
GOTOXY(10,11); WRITE(' 1. Ui =0 PARA TODO I ');
GOTOXY(10,14); WRITE(' QUE POTENCIALES DESEAS : ');
W:=14; AFIRMAR1(P,W);
IF P=1 THEN
FILLCHAR(U,SIZEOF(U),0);
IF P=0 THEN
POTENCIALES;
ENTERR;
POTENCIALES;
RETORNO(X,Y,FLUJ,BETA); METODOS;
END;
4:BEGIN
GOTOXY(22,22);WRITE('CUAL ES EL NOMBRE DEL ARCHIVO ');
READLN(SUN); CAP; FUNCION; METODOS; END;
5:BEGIN
END;
END;
PROCEDURE MODIFICACION;
BEGIN
VENTA; GOTOXY(10,5);WRITELN(' MENU DE MODIFICACIONES ');
GOTOXY(10,10);WRITELN('1. MATRIZ DE CAPACIDADES MINIMAS ');
GOTOXY(10,12);WRITELN('2. MATRIZ DE CAPACIDADES MAXIMAS ');
GOTOXY(10,14);WRITELN('3. MATRIZ DE COSTOS ');
GOTOXY(10,16);WRITELN('4. MENU DE CAPTURA ');
GOTOXY(10,20);WRITE('CUAL ES LA MATRIZ QUE DESEAS MODIFICAR : ');
DIG:=1; W:=0; REPEAT;
DIGITAR(OPER,DIG,W);
IF (OPCION<1) OR (OPCION>4) THEN
COMPONER;
UNTIL (OPCION<5) AND (OPCION>0);
CASE OPER OF
1: BEGIN MODI; MODIFICACION; END;
```

```

PROCEDURE LLECE;
BEGIN
FILLCHAR(Y,SIZEOF(Y),0);
FILLCHAR(FLUJ,SIZEOF(FLUJ),0);
FILLCHAR(x,SIZEOF(x),0);
END;
PROCEDURE GRAFI_CA;
BEGIN
VENTA; AYUD;
ENTERR; N:=3; Y[1,2]:=1;Y[2,3]:=1;
Y[3,1]:=1;GRAFICAR(FLUJ);
END;
PROCEDURE PASEO;
BEGIN
VENTA;PASS;ENTERR;
N:=3; Y[1,2]:=1;Y[2,3]:=1; GRAFICAR(FLUJ);END;
PROCEDURE CIRCUITO;{*****DEFINICION *****}
BEGIN VENTA; CIRC;
ENTERR;FILLCHAR(y,SIZEOF(y),0);N:=4;Y[1,2]:=1;
Y[2,3]:=1;Y[3,1]:=1; Y[3,4]:=1;
GRAFICAR(fluj);
END;

PROCEDURE DIGRAFI_CA;{*****DEFINICION *****}
BEGIN
VENTA;DIGR; ENTERR;
FILLCHAR(y,SIZEOF(y),0);N:=5;Y[1,2]:=1;
Y[1,3]:=1;Y[2,4]:=1;Y[3,2]:=1;
Y[3,4]:=1; Y[4,5]:=1;
GRAFICAR(fluj);
END;

PROCEDURE RED;{***** DEFINICION *****}
BEGIN
VENTA;REDD;ENTERR;N:=5;
FILLCHAR(y,SIZEOF(y),0);Y[1,2]:=1;
Y[1,3]:=1; Y[2,4]:=1;
Y[3,2]:=1; Y[3,4]:=1; Y[4,5]:=1;
GRAFICAR(fluj);
END;

PROCEDURE FLU_EJE;
BEGIN
VENTA;FLU;ENTERR;N:=5;
FILLCHAR(y,SIZEOF(y),0);Y[1,2]:=1; Y[1,3]:=1;
Y[2,4]:=1;Y[3,2]:=1; Y[3,4]:=1;
Y[4,5]:=1;Y[5,1]:=1;
GRAFICAR(FLUJ);
END;

PROCEDURE IMPMAT;
BEGIN
VENTA;PRINT1;IMPRIME(C); ENVENTA;
PRINT2;IMPRIME(A);
ENVENTA;GOTOXY(10,2);
PRINT3;IMPRIME(B); ENTERR;
END;

PROCEDURE CAP;
BEGIN
IF (SUN=") OR (SUN=SAN) THEN
SUN:=SAN; INICIO;
ASSIGN(ARA,SUN);
RESET(ARA);
READ(ARA,N);
READLN(ARA); INICIO;
FOR I:=1 TO N DO

```

ESTA TRABAJO NO DEBE  
SALIR DE LA BIBLIOTECA

## CAPITULO IV. OPERACION DEL SISTEMA DE COMPUTO

```
REPEAT
DIG:=5; DIGITAR(OPCION,DIG,J);
IF (OPCION<1) OR (OPCION>6) THEN
COMPONER;
UNTIL (OPCION<7) AND (OPCION>0);
WRITELN;ENTERR;
CASE OPCION OF
1:BEGIN
  DATOS; GRAFICAR(FLUJ); EJEMPLO; END;
2:BEGIN
  DATOS; FUNCION; EJEMPLO; END;
3:BEGIN
  INICIO; DATOS; FLUJO; EJEMPLO; END;
4:BEGIN
  INICIO; DATOS;
  VENTA; GOTOXY(15,6); WRITE('VALORES DE U');
  GOTOXY(10,9); WRITE(' 0. U: GENERADO POR EL PROGRAMA. ');
  GOTOXY(10,11); WRITE(' 1. U: =0 PARA TODO L');
  GOTOXY(10,14); WRITE(' QUE POTENCIALES DESEAS: ');
  W:=14; AFIRMAR(P,W);
  IF P=1 THEN
    FILLCHAR(U,SIZEOF(U),0);
  IF P=0 THEN
    POTENCIALES;
    ENTERR;
    RETORNO(X,Y); EJEMPLO;
  END;
5:BEGIN
  INICIO; DATOS; VENTA;
  GOTOXY(10,2); WRITE('UUUUUUUUUUUUUUUUUUUUUU');
  GOTOXY(10,3); WRITE(' MATRIZ DE COSTOS');
  GOTOXY(10,4); WRITE('UUUUUUUUUUUUUUUUUUUUUU');
  IMPRIME(C); ENVENTA;
  GOTOXY(10,2); WRITE('UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU');
  GOTOXY(10,3); WRITE(' MATRIZ DE CAPACIDADES MINIMAS');
  GOTOXY(10,4); WRITE('UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU');
  IMPRIME(A); ENVENTA;
  GOTOXY(10,2); WRITE('UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU');
  GOTOXY(10,3); WRITE(' MATRIZ DE CAPACIDADES MAXIMAS');
  GOTOXY(10,4); WRITE('UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU');
  IMPRIME(B); ENTERR; EJEMPLO;
  END;
6:BEGIN END;
  END;
  END;
  PROCEDURE AYUDA:(MENU DE AYUDA)
  BEGIN
  VENTA; J:=0; LLECE;
  GOTOXY(23,5); WRITE(' MENU DE TERMINOLOGIA ');
  GOTOXY(15,7); WRITE(' 1.- DEFINICION DE GRAFICA');
  GOTOXY(15,9); WRITE(' 2.- DEFINICION DE DIGRAFICA');
  GOTOXY(15,11); WRITE(' 3.- DEFINICION DE PASEO');
  GOTOXY(15,13); WRITE(' 4.- DEFINICION DE CIRCUITO');
  GOTOXY(15,15); WRITE(' 5.- DEFINICION DE RED ');
  GOTOXY(15,17); WRITE(' 6.- DEFINICION DE FLUJO ');
  GOTOXY(15,19); WRITE(' 7.- MENU PRINCIPAL ');
  GOTOXY(14,20); WRITE(' DAME LA OPCION DESEADA: ');
  REPEAT
  DIG:=7; DIGITAR(OPCION,DIG,J);
  IF (OPCION<1) OR (OPCION>7) THEN
  COMPONER;
  UNTIL (OPCION<8) AND (OPCION>0);
  WRITELN;ENTERR;
  CASE OPCION OF
  1:BEGIN
    GRAFI_CA; AYUDA; END;
  2:BEGIN
    DIGRAFI_CA; AYUDA; END;
  3:BEGIN
    PASEO; AYUDA; END;
  4:BEGIN
```

```
DIGRAFI_CA;
AYUDA;
END;
3:BEGIN
  RED;
  AYUDA;
  END;
4:BEGIN
  FLU_EJE;
  AYUDA;
  END;
5:BEGIN
  END;
END;
END;
PROCEDURE MENU_PRIN ;
BEGIN
  VENTA;
  INICIO;
  GOTOXY(28,5);
  WRITE(' MENU PRINCIPAL ');
  GOTOXY(15,11);WRITE(' 1. MENU DE TERMINOLOGIA ');
  GOTOXY(15,13);WRITE(' 2. MENU DE EJEMPLO ');
  GOTOXY(15,15);WRITE(' 3. MENU DE METODOS ');
  GOTOXY(15,17);WRITE(' 4. SALIDA ');
  GOTOXY(19,20);WRITE('DAME LA OPCION DESEADA: ');
  REPEAT
  DIG:=1; DIGITAR(OPCI,DIG,J);
  IF (OPCI<1) OR (OPCI>4) THEN
  COMPOSER;
  UNTIL (OPCI<5) AND (OPCI>0);
  WRITELN;
  ENTERR;
  CASE OPCIOF
  1:BEGIN
    AYUDA;
    MENU_PRIN; END;
  2:BEGIN
    EJEMPLO;
    MENU_PRIN; END;
  3:BEGIN
    CAPTURA;
    MENU_PRIN; END;
  4:BEGIN
    END;
  END;
  END;
  END;
```

Existen dos unidades que son llamadas por el programa desvia las cuales son Portada y Unidad, la primera presenta la portada de MAC y el objetivo del paquete, la segunda sirve para validar los datos y poner color a las ventanas, las cuales se encuentran en el Anexo D.

### IV.3 MANUAL DEL OPERACIÓN

#### IV.3.1 OBJETIVO DEL PAQUETE:

El objetivo de este paquete, es introducir al usuario en el manejo del software **DESVIA**. El cual se ha desarrollado para la materia de Optimización combinatoria, enfocándose al método o algoritmo de desviaciones. este es para la ayuda del estudiante y contiene un menú de terminología como gráfica, red, digráfica, flujo, paseo y circuito. Ya que, estos conceptos son de suma importancia para el tema que se trata en el trabajo, éste se complementa con gráficas. En él, los algoritmos programados son los de flujo máximo y el de desviaciones; que el algoritmo de flujo máximo se ve también en la materia de teoría de gráficas, y se puede utilizar como material didáctico para la exposición de este tema, esto porque se van teniendo las gráficas de cada una de las soluciones, antes de encontrar el flujo máximo. este algoritmo se describe en el Capítulo II. El algoritmo de desviaciones es una metodología muy interesante dado que se desea conseguir un flujo máximo en la red a un costo mínimo y es lo que hace esta metodología, pero este algoritmo puede tender a ser muy tedioso manualmente por el número de iteraciones, antes de obtener el resultado óptimo, por eso, se ha programado, lo cual da al estudiante y profesor una reducción de tiempo y una comprensión mejor del tema. el paquete incluye un programa de prueba el cual introduce al usuario sobre los métodos o algoritmos ya citados.

### IV.3.2 PASCAL

El lenguaje **PASCAL** estándar presenta una serie de características que lo hacen idóneo como primero o segundo lenguaje de programación, desde el punto de vista del aprendizaje y como lenguaje de propósito general para la resolución de aplicaciones de todo tipo: gestión, científicas, ingeniería, etc.. Destacamos en este apartado las características más sobresalientes:

- \* Lenguaje excelente para el aprendizaje de programación.
- \* Lenguaje de propósito general.
- \* Lenguaje procedural (imperativo, orientado a órdenes).
- \* Lenguaje estructurado (soporta las estructuras básicas **while**, **for** y **repeat**; no necesita el uso de la sentencia **goto**).
- \* Lenguaje recursivo( soporta la recursividad: propiedad de llamarse así mismo una función o un procedimiento).
- \* Gran riqueza de tipos ( fuertemente tipeado), con tipo de datos simples y estructurados así como datos definidos por el usuario.
- \* Producción por los compiladores de programas ejecutables rápidos y eficientes.
- \* Facilidad para realizar programación modular debido a la posibilidad de diseñar subprogramas o módulos del tipo procedimiento y función.

### IV.3.3 TURBO PASCAL

En 1983, la firma Borland International lanzó al mercado un compilador de Pascal denominado Turbo Pascal (versión 1.0) por el notable precio de \$49.95 (dólares USA), y a modo experimental. La versión fue un éxito ya no solamente funcionaba, sino que compilaba y corría (ejecutaba) los programas más rápido que cualquier compilador de Pascal existente en el mercado.

La razón fundamental del éxito inicial de Turbo Pascal, no sólo residía en su bajo precio, sino que ofrecía al mismo tiempo un entorno de programación (editor/compilador) y numerosas utilidades inexistentes en otras versiones de Pascal como gráficos, colores, gestor de archivo, etc. La versión 1.0 prácticamente tuvo una corta vida y pronto aparecieron las versiones 2.0 y 3.0, esencialmente esta última se popularizó notablemente y ha sido el primero o segundo lenguaje de muchos profesionistas actuales.

No contento con el éxito, Borland lanzó a finales de 1987 la versión 4.0, la cual conservaba todas las peculiaridades de 3.0 con la que era compatible, pero ofrecía un entorno de programación más potente y en líneas generales mejoraba considerablemente el producto. Sin embargo, Borland ha seguido perfeccionando su compilador, y en 1988 lanzó la versión 5.0 y durante el año 1989 ha sacado una nueva versión, la 5.5, que sigue las nuevas filosofías del campo de la programación: **programación orientada a objetos**, y que decir de los grandes del mundo informático es la programación de futuro.

Turbo Pascal es sin lugar a dudas una herramienta de programación potente, fácil de utilizar e idónea para el aprendizaje de profesionales o aficionados a la programación, ya que ha añadido a todas las ventajas de Pascal deseadas; anteriormente las propias del entorno de programación aquí citada.

### IV. 3.4 REQUERIMIENTOS DE HADWARE

- Sistema Operativo versión 3.01 o posterior.
- CPU XT ó 8086, AT ó 80286, ó posteriores.
- 640 Kb en memoria RAM
- Cualquier monitor ya que el programa en caso de no tenerlo manda un mensaje de error al usuario.
- Unidad de disco flexible de 3 ½ pulgadas.

### IV.3. 5 COMO ENTRAR AL PAQUETE.

Para entrar en este paquete se deberá introducir el disquete que contiene el paquete de Desvia en A , y se deberá teclear *Desvia*, como se muestra:

*>Desvia*

y pulsar [ENTER], para continuar.

Este paquete fue realizado en la versión 5.5 de Turbo Pascal.

Lo primero que aparece en pantalla es el logo de MAC, y después el objetivo del paquete, no se debe pulsar ninguna tecla. A continuación, se desplegará en pantalla el **MENÚ PRINCIPAL**, en el cual el usuario debe especificar cual es la opción que desea.

El *Menú Principal* tiene el siguiente contenido:

*1.- Menú de Terminología*

*2.- Menú de Ejemplo*

*3.- Menú de Métodos*

*4.- Salida*

Y se le pide al usuario que introduzca la opción deseada, la cual debe ser del 1 al 4; cualquier otra opción dada (caracteres, o números diferentes del 1 al 4), no será aceptada y aparece un mensaje donde se pide que se introduzca la tecla correcta.

Si el usuario desea ver la opción de **Menú de Términos**, deberá teclear 1 y se le pedirá que pulse [ENTER], entonces se desplegará el Sub-menú del Términos, el cual contiene diferentes definiciones muy útiles.

Si el usuario desea ver la opción de **Menú de Ejemplo**, deberá teclear 2 y pulsar [ENTER], y se desplegará este Sub-menú, del cual se dará su descripción más adelante.

Si el usuario desea ver la opción de **Menú de Métodos**, deberá teclear 3 y teclear [ENTER], y se desplegará este Sub-menú, en el cual se captura los datos de la red y se hacen las modificaciones necesarias, después de esto se puede pasar al Menú de Métodos para poder ver gráfica, matrices de capacidades y costos, así como el flujo máximo y el flujo máximo a costo mínimo.

Si el usuario desea **SALIR** deberá teclear 4 y [ENTER], y saldrá del paquete.

### IV.3.6 FORMA DE UTILIZAR EL MENÚ DE AYUDA

Este Menú tiene el siguiente contenido:

- 1. Definición de Gráfica.**
- 2. Definición de Digráfica.**
- 3. Definición de Paseo**
- 4. Definición de Circuito**
- 5. Definición de Red.**
- 6. Definición de Flujo.**
- 7. Menú Principal.**

Todas las definiciones van acompañadas por una gráfica asociada, para ilustrar mejor a cada una de las definiciones.

Cualquier opción diferente de 1 a 5 no será aceptada, y aparecerá en pantalla que se presione la tecla correcta, si está fuera de este rango.

Para ver una definición específica, el usuario debe dar la opción deseada la cual es de 1 a 4 y pulsar [ENTER]. Así aparecerá la definición correspondiente a la opción, después se debe pulsar nuevamente [ENTER] para así poder ver el grafo que corresponde a la definición y para regresar al Menú de Terminología se debe pulsar [ENTER].

Si no se desea ver ninguna definición y se quiere volver al Menú principal se debe elegir la opción 5.

#### **IV.3.7 FORMA DE PRESENTACIÓN DE RESULTADOS DEL FLUJO MÁXIMO A COSTO MÍNIMO POR EL MÉTODO DE DESVIACIONES.**

Como este método es iterativo, a continuación se presenta la forma en que aparecen los resultados.

1. Este método, presenta un análisis de los arcos, proporcionando información sobre a que conjunto pertenece cada uno de ellos. A continuación, se debe teclear [ENTER], como se pide en pantalla.
2. Después se presentará la tabla para los cálculos asociados y en el costado derecho de la tabla aparecerán la etiquetación correspondiente, de acuerdo a la expuesta en el capítulo III.

#### CAPITULO IV. OPERACION DEL SISTEMA DE COMPUTO

---

3. A continuación se presenta la gráfica asociada a la etiquetación, aquí se debe poner muchísima atención y ver si existe un circuito o no; si existe un circuito, anotar que arcos no pertenecen al circuito, ya que en la siguiente pantalla se le pedirán para aumentar el flujo o cambiar potenciales, después que se ha observado y anotado los datos correspondientes se debe hacer lo que se pide en pantalla.
4. Si existe un circuito, el paquete pedirá el número de arcos que no están en el circuito, los cuales son los que anoto el usuario de la gráfica anterior a esta pantalla, el usuario debe dar solamente el número de arcos que no pertenecen al circuito y pulsar [ENTER], cualquier otra opción no será aceptada. Después se piden las coordenadas de los arcos, estas deben ser dadas de la siguiente manera, si por ejemplo, existen 2 arcos que no pertenecen y son [3,4] y [5,7] se debe pulsar 3 y [ENTER] aparecerá una coma y se después se debe pulsar 4 y [ENTER] nuevamente, después 5, [ENTER] y aparece una coma y se pulsa 7 acompañado con [ENTER], y se habrá acabado, después se incrementa el flujo y se debe pulsar nuevamente ENTER, a continuación se presenta el circuito al que se le incrementara el flujo.
5. Si no existe un circuito esto ocasionará que se calculen unos nuevos potenciales, de acuerdo a la subrutina de cambio de potenciales que se presenta en el capítulo III.

Dados todos estos pasos se vuelven a repetir, a partir del 1 hasta que todos los arcos son conformables.

Cuando todos los arcos son conformables se presenta la matriz X la cual contiene el flujo que existe entre cada uno de los arcos, también se presenta el costo asociado a este flujo y la gráfica asociada a la matriz; en dicha gráfica si el arco esta saturado el arco tiene una línea más gruesa, si el arco tiene un flujo mayor a cero, pero menor a la capacidad máxima tiene una arista delgada, y si el arco tiene un flujo igual a cero se tendrá un arco punteado. El usuario puede guardar los resultados en un archivo que el mismo asigne con

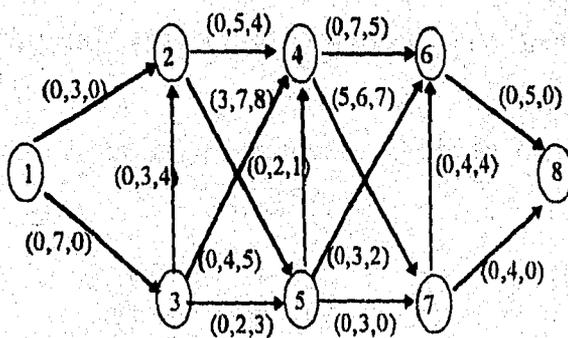
cualquier extensión, e indicando el driver donde se va a grabar el archivo, el cual no debe de sobrepasar de 12 caracteres.

### IV.3.8 FORMA DE UTILIZAR EL MENÚ DE EJEMPLO

Este Menú tiene el siguiente contenido:

1. *Gráfica asociada a la red.*
2. *Modelo Matemático Primal*
3. *Flujo máximo.*
4. *Flujo máximo a costo mínimo.*
5. *Matriz de costos, y de capacidades.*
6. *Menú Principal.*

En este menú ya se tiene una red de 8 nodos y es la siguiente:



El paquete nos proporciona una digráfica de la red, de acuerdo a su matriz de Adyacencia dando la opción 1 y se le pide al usuario que pulse [ENTER] para continuar.

#### CAPITULO IV. OPERACION DEL SISTEMA DE COMPUTO

---

En la opción 2 se podrá ver el modelo matemático en su forma primal de la red de transporte.

Como se puede observar la red original tiene 3 valores entre paréntesis, los cuales corresponden a la capacidad mínima de flujo, la capacidad máxima de flujo, y el costo por unidad de flujo del arco  $(i,j)$ ; los cuales se pueden observar en la opción 5, que es la que contiene todos los datos anteriores.

En la opción 3, se le aplica a la red, el algoritmo de flujo máximo; en esta opción se debe solo seguir las instrucciones en pantalla.

Si el usuario desea ver como se aplica a esta red el método de las desviaciones debe elegir la opción 4, y seguir las instrucciones en pantalla.

Para ver como los resultados se presentan, ver el apartado anterior, donde se dice como se aparecerán.

Para esta red. Se tienen 7 iteraciones y para cada iteración se debe dar lo siguiente:

Iteración No.	Número de arcos	Arcos que no pertenecen al circuito
1	3	(1,3) , (2,4) , y (4,7)
2		
3	2	(4,7) y (3,2)

## CAPITULO IV. OPERACION DEL SISTEMA DE COMPUTO

---

Iteración No.	Número de arcos	Arcos que no pertenecen al circuito
4	1	
5	2	(4,6) y (3,2)
6		
7	0	

Entonces se dará la matriz X que es el flujo que existe en cada uno de los arcos, el costo mínimo y el flujo máximo, el cual es 102 y 9 respectivamente. Pulsando ENTER, se presenta la gráfica de cuyos arcos son mayores a cero y después se regresará al Menú de Ejemplo.

Si el usuario desea regresar al Menú principal debe elegir la opción 5. Entonces se desplegará el Menú principal.

### **IV.3.9 FORMA DE UTILIZAR EL MENÚ DE CAPTURA**

Este menú tiene los siguiente:

- 1. Lectura de Datos**
- 2. Modificación de Datos**
- 3. Impresión de Matriz de costos y de capacidades.**
- 4. Menú de Métodos**

Primero, el usuario debe elegir la opción 1 en donde se le pide al usuario que inserte un disco para grabar el archivo de los datos que serán capturados. La captura de los datos debe hacerse de la siguiente manera:

Se debe introducir el número de nodos existentes en la red, el cual debe ser mayor o igual a 3 y teclear [ENTER]. A continuación se pide al usuario que inserte un 1 ó 0 para saber si existe o no una línea entre el arco (i,j). Cualquier otra opción diferente a las anteriores no será aceptada.

Después, de acuerdo a la matriz anterior, se pide al usuario que introduzca los valores de capacidades de los arcos, así como los costos asociados. (Estos valores pueden ser negativos, ceros o positivos, cualquier carácter diferente de 0 a 9 y - , no se aceptará).

Si el usuario cometió algún error en la captura de los datos debe de elegir la opción 2, esta opción pedirá el nombre del archivo que se desea modificar y permite al usuario cambiar los datos siguientes:

- 1. Matriz de capacidades mínimas**
- 2. Matriz de capacidades máximas**
- 3. Matriz de costos**
- 4. Menú de captura**

En las opciones de 1 a la 3 de este sub menú se pide el número de cambios a realizar y el nuevo valor para cambiar, después de insertar los valores se debe pulsar [ENTER].

La opción 4 nos regresa al Menú de captura.

Si se desea ver la impresión de las matrices se debe pulsar 3 y se mostrarán.

Si el usuario ya tiene un archivo donde ha capturado su matriz, y esta seguro de que sus datos están correctos, entonces debe escoger la opción 5 que nos lleva al Menú de Métodos.

#### IV.3.10 FORMA DE UTILIZAR EL MENÚ DE MÉTODOS

Este Menú tiene el siguiente contenido:

- 1. Flujo máximo.*
- 2. Gráfica asociada a la red.*
- 3. Flujo máximo a costo mínimo.*
- 4. Modelo Matemático Primal.*
- 5. Menú Principal.*

Si se desea ver el flujo máximo de la red capturada, el usuario debe pulsar 1 y dar el nombre del archivo donde capturo sus datos y seguir las indicaciones en pantalla.

Si el usuario desea ver la digráfica asociada a una red específica, debe elegir la opción 2 e introducir el nombre del archivo asociado al de la digráfica que se desea.

Si el usuario desea ver la solución del problema de flujo máximo a costo mínimo por el método de las desviaciones debe escoger la opción 3 y dar el nombre del archivo y seguir las indicaciones que se dieron en el apartado VI.

Si se desea ver el modelo matemático de alguna red, se debe escoger la opción 4 e insertar el nombre del archivo donde fue capturada dicha red, después de esto solo se deben seguir las indicaciones en pantalla.

Y con 5, se vuelve al Menú principal.

#### **IV.3.11 COMO SALIR DEL PAQUETE.**

Si el usuario desea salir del paquete y se encuentra en el Menú principal, solamente debe teclear 4 y [ENTER]. En caso de que no se encuentre el usuario en el menú principal deberá primero regresar al Menú principal y después escoger la opción de salida.

IV.3 UTILIZACION DEL SISTEMA

A continuación se presentan ejemplos de redes que han sido resueltos por el paquete descrito anteriormente.

Este primer ejemplo tiene 7 vértices, y se presenta las matrices asociadas, gráfica y las iteraciones al aplicar el algoritmo de flujo máximo y el flujo máximo a costo mínimo. Este ejemplo se resolvió tomando los potenciales que genera el programa.

CAPTURA DE DATOS

- 1.- LECTURA DE DATOS
- 2.- MODIFICACIÓN DE DATOS
- 3.- IMPRESIÓN DE MATRIZ DE COSTOS Y DE CAPACIDADES
- 4.- MATRIZ DE RESULTADOS
- 5.- MENÚ DE MÉTODOS

DAME LA OPCIÓN DESEADA : 3

CUAL ES EL NOMBRE DEL ARCHIVO : EJER1

PRESIONE [ENTER] PARA CONTINUAR

MATRIZ DE COSTOS

	1	2	3	4	5	6	7
1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0

PRESIONE [ENTER] PARA CONTINUAR

MATRIZ DE CAPACIDADES MINIMAS

	1	2	3	4	5	6	7
1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0

PRESIONE [ENTER] PARA CONTINUAR

MATRIZ DE CAPACIDADES MAXIMAS

	1	2	3	4	5	6	7
1	0	6	7	0	0	0	0
2	0	0	1	3	4	0	0
3	0	0	0	2	0	5	0
4	0	0	0	0	1	2	0
5	0	0	0	0	0	0	5
6	0	0	0	0	2	0	4
7	9	0	0	0	0	0	0

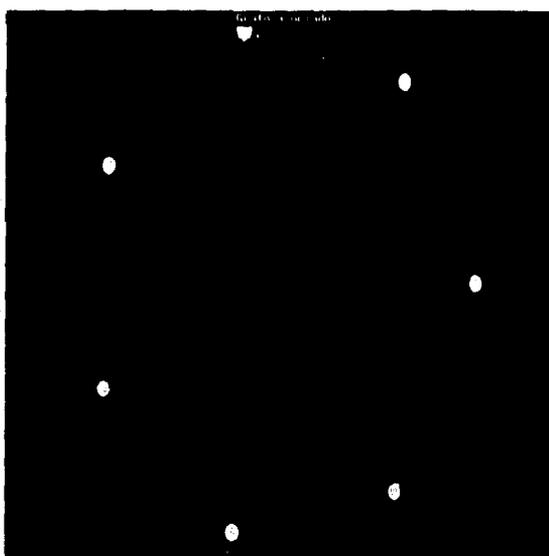
PRESIONE [ENTER] PARA CONTINUAR

MENU DE METODOS

- 1.- FLUJO MAXIMO DE LA RED
- 2.- DIGRAFICA DE LA RED
- 3.- FLUJO MAXIMO A COSTO MINIMO EN UNA RED
- 4.- MODELO PRIMAL
- 5.- MENU PRINCIPAL

DAME LA OPCION DESEADA : 2

CUAL ES EL NOMBRE DEL ARCHIVO : EJER1  
PRESIONE [ENTER] PARA CONTINUAR



MENU DE METODOS

- 1.- FLUJO MAXIMO DE LA RED
- 2.- DIGRAFICA DE LA RED
- 3.- FLUJO MAXIMO A COSTO MINIMO EN UNA RED
- 4.- MODELO PRIMAL
- 5.- MENU PRINCIPAL

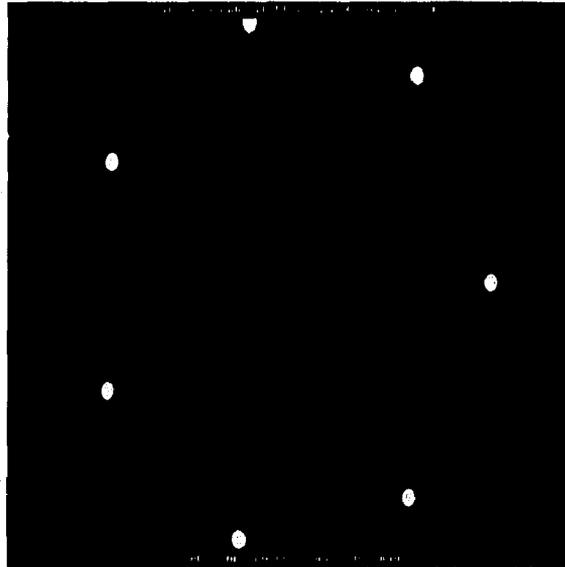
DAME LA OPCION DESEADA : 1

CUAL ES EL NOMBRE DEL ARCHIVO :EJER1  
PRESIONE [ENTER] PARA CONTINUAR

VALORES DE GAMA E INCREMENTO DEL FLUJO  
la corte inicial es 9 y la corte final es 9

GAMA[3] : 7  
GAMA[6] : 5  
GAMA[7] : 4  
x [1,3] 4  
x [3,6] 4  
x [6,7] 4

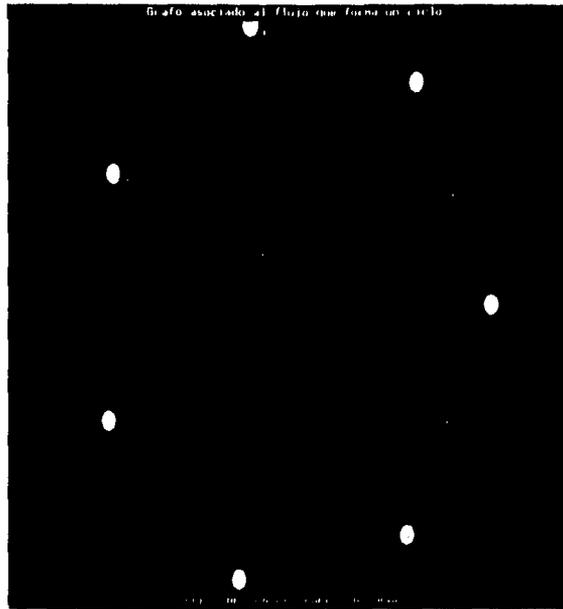
PRESIONE [ENTER] PARA CONTINUAR



VALORES DE GAMA E INCREMENTO DEL FLUJO

GAMA[2]: 6  
GAMA[5]: 4  
GAMA[7]: 4  
x [1,2] 4  
x [2,5] 4  
x [5,7] 4

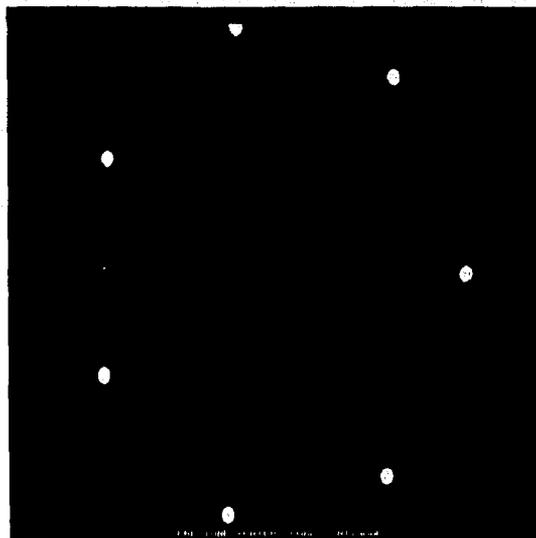
PRESIONE [ENTER] PARA CONTINUAR



VALORES DE GAMA E INCREMENTO DEL FLUJO

```
GAMA[3]: 3  
GAMA[4]: 2  
GAMA[5]: 1  
GAMA[7]: 1  
x [1,3] 5  
x [3,4] 1  
x [4,5] 1  
x [5,7] 5
```

PRESIONE [ENTER] PARA CONTINUAR

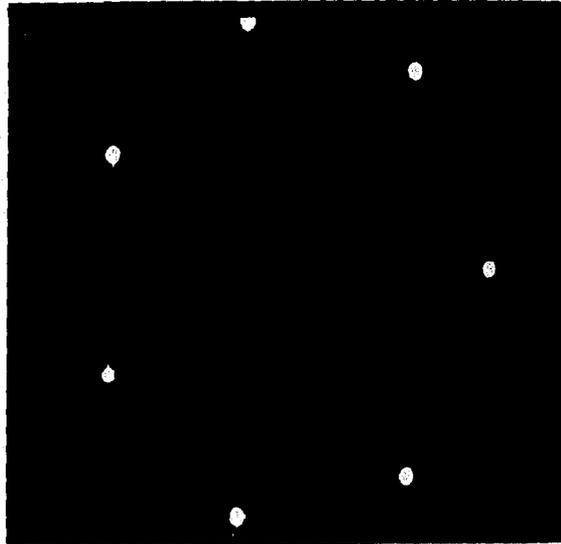


El Flujo Maximo es: 9

1	2	3	4	5	6	7
1	0	4	5	0	0	0
2	0	0	0	0	4	0
3	0	0	0	1	0	4
4	0	0	0	0	1	0
5	0	0	0	0	0	5
6	0	0	0	0	0	4
7	9	0	0	0	0	0

Como deseas que se llame el archivo de resultados : RES11

PRESIONE [ENTER] PARA CONTINUAR



MENU DE METODOS

- 1.- FLUJO MAXIMO DE LA RED
- 2.- DIGRAFICA DE LA RED
- 3.- FLUJO MAXIMO A COSTO MINIMO EN UNA RED
- 4.- MODELO PRIMAL
- 5.- MENU PRINCIPAL

DAME LA OPCION DESEADA : 3

CUAL ES EL NOMBRE DEL ARCHIVO :EJER1

PRESIONE [ENTER] PARA CONTINUAR

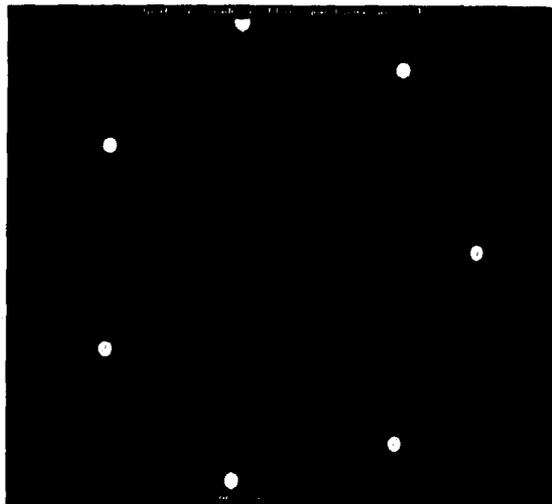
A continuación se presenta la tabla donde se muestra a que conjunto pertenece cada arco.

- El arco 1, 2 Es Conformable y pertenece a B
- El arco 1, 3 Es Conformable y pertenece a B
- El arco 2, 3 Es Conformable y pertenece a B
- El arco 2, 4 Es Conformable y pertenece a B
- El arco 2, 5 Es Conformable y pertenece a B
- El arco 3, 4 Es Conformable y pertenece a B
- El arco 3, 6 Es Conformable y pertenece a B
- El arco 4, 5 Es Conformable y pertenece a B
- El arco 4, 6 Es Conformable y pertenece a B
- El arco 5, 7 Es Conformable y pertenece a B
- El arco 6, 5 Es Conformable y pertenece a B
- El arco 6, 7 Es Conformable y pertenece a B
- El arco 7, 1 No es Conformable y pertenece a B1

PRESIONE [ENTER] PARA CONTINUAR

VI	1	2	3	4	5	6	7
1	0		0-6	0-7			[-,∞]
2	0		0 0	0 0			[1,6]
3	0			0-1	0-3	0-4	[1,7]
4	0			0 0	0 0		[2,3]
5	0				0-1	0-2	[2,4]
6	0				0 0	0 0	[3,5]
7	0	0-9			0-2	0-4	[5,4]
		-9 0			0 0	0 0	

PRESIONE [ENTER] PARA CONTINUAR



Por favor, si el arco no se encuentra en el circuito  
incerta las coordenadas a continuación.

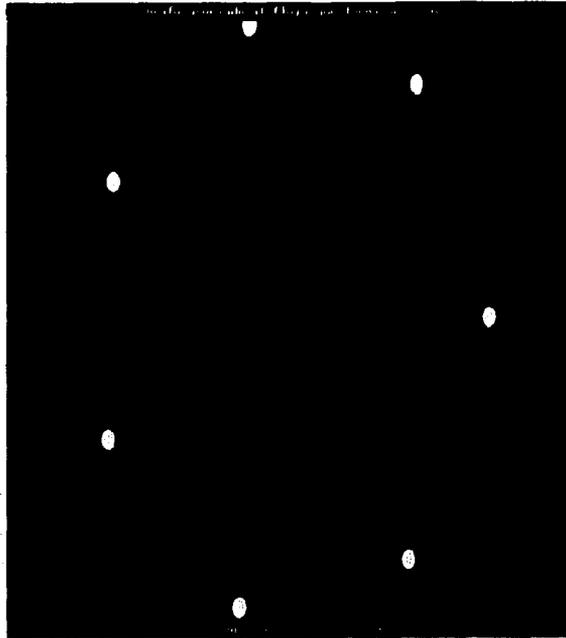
¿Cuántos arcos no pertenecen al circuito?      3  
Dame las coordenadas

[ 1 ,3 ]  
[ 3 ,6 ]  
[ 2 ,4 ]

x [1,2] 4  
x [2,5] 4  
x [5,7] 4  
x [7,1] 4

PRESIONE [ENTER] PARA CONTINUAR

CAPITULO IV. OPERACION DEL SISTEMA DE COMPUTO



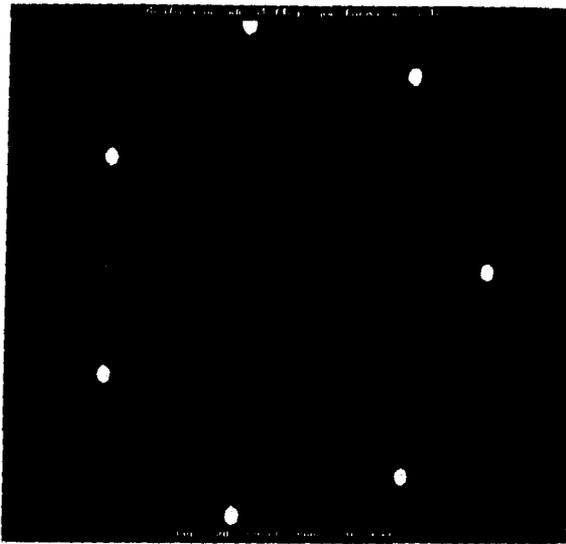
A continuación se presenta la tabla donde se muestra a que conjunto pertenece cada arco.

- El arco 1, 2 Es Conformable y pertenece a B
- El arco 1, 3 Es Conformable y pertenece a B
- El arco 2, 3 Es Conformable y pertenece a B
- El arco 2, 4 Es Conformable y pertenece a B
- El arco 2, 5 Es Conformable y pertenece a B
- El arco 3, 4 Es Conformable y pertenece a B
- El arco 3, 6 Es Conformable y pertenece a B
- El arco 4, 5 Es Conformable y pertenece a B
- El arco 4, 6 Es Conformable y pertenece a B
- El arco 5, 7 Es Conformable y pertenece a B
- El arco 6, 5 Es Conformable y pertenece a B
- El arco 6, 7 Es Conformable y pertenece a B
- El arco 7, 1 No es Conformable y pertenece a B1

PRESIONE [ENTER] PARA CONTINUAR

	VI	1	2	3	4	5	6	7
VI UI	UI	0	0	0	0	0	0	0
1	0		4-2	0-7				[-,∞]
			4	0	0			
2	0			0-1	0-3	0-4		[1,2]
				0	0	0	0	
3	0				0-2		0-5	[1,7]
					0	0	0	
4	0					0-1	0-2	[2,2]
						0	0	
5	0						4	[4,1]
							4	
6	0					0-2	0-4	[3,5]
						0	0	
7	0		4-5					[5,1]
			-5	0				

PRESIONE [ENTER] PARA CONTINUAR



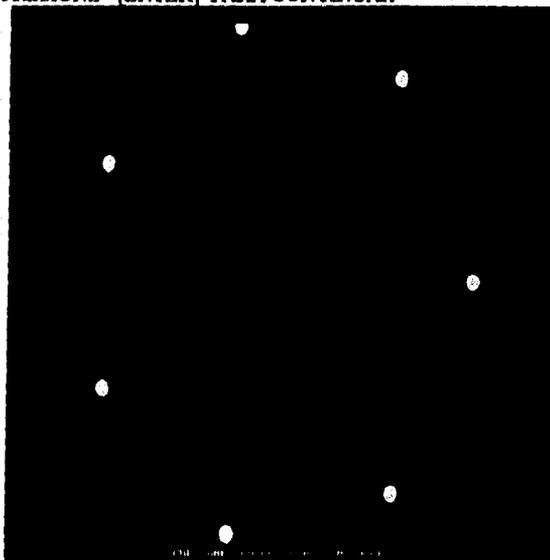
Por favor, si el arco no se encuentra en el circuito  
incerta las coordenadas a continuación.

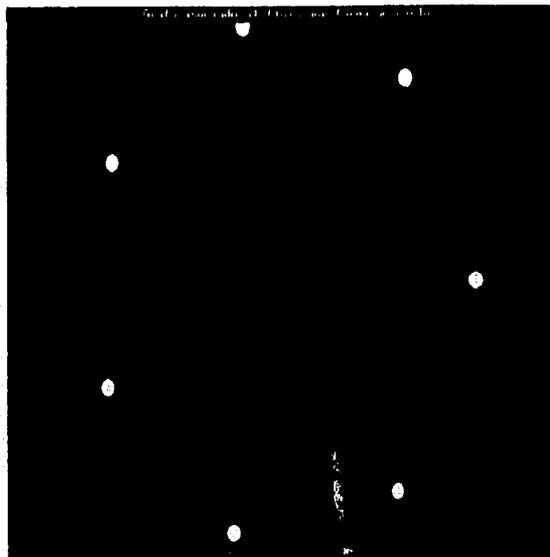
¿Cuántos arcos no pertenecen al circuito?      2  
Dame las coordenadas

[ 1 ,3 ]  
[ 3 ,6 ]

x [1,2] 5  
x [2,4] 1  
x [4,5] 1  
x [5,7] 5  
x [7,1] 5

PRESIONE [ENTER] PARA CONTINUAR





A continuación se presenta la tabla donde se muestra a que conjunto pertenece cada arco.

- El arco 1, 2 Es Conformable y pertenece a B
- El arco 1, 3 Es Conformable y pertenece a B
- El arco 2, 3 Es Conformable y pertenece a B
- El arco 2, 4 Es Conformable y pertenece a B
- El arco 2, 5 Es Conformable y pertenece a B
- El arco 3, 4 Es Conformable y pertenece a B
- El arco 3, 6 Es Conformable y pertenece a B
- El arco 4, 5 Es Conformable y pertenece a B
- El arco 4, 6 Es Conformable y pertenece a B
- El arco 5, 7 Es Conformable y pertenece a B
- El arco 6, 5 Es Conformable y pertenece a B
- El arco 6, 7 Es Conformable y pertenece a B
- El arco 7, 1 No es Conformable y pertenece a B1

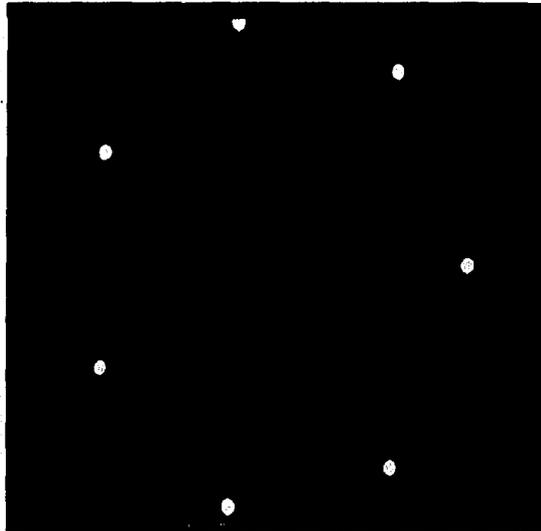
PRESIONE [ENTER] PARA CONTINUAR

V:	1	2	3	4	5	6	7
VI UI	0	0	0	0	0	0	0
1 0		5 -1	0 -7				[-,∞]
		5 0	0 0				
2 0			0 -1	1 -2	4 -0		[1,1]
			0 0	1 0	4 0		
3 0				0 -2	0 -5		[1,7]
				0 0	0 0		
4 0					1 0	0 -2*	[2,1]
					1 0	0 0	
5 0						5 0	[6,2]
						5 0	
6 0					0 -2	0 -4	[3,5]
					0 0	0 0	
7 0	5 -4						[6,4]
	-4 0						

CAPITULO IV. OPERACION DEL SISTEMA DE COMPUTO

---

PRESIONE [ENTER] PARA CONTINUAR

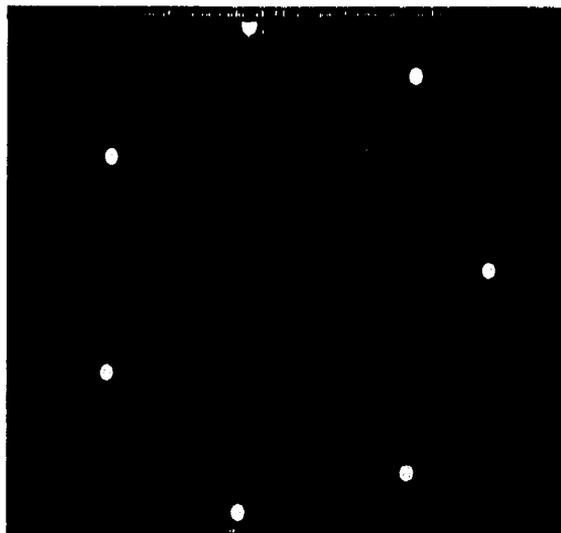


Por favor, si el arco no se encuentra en el circuito  
incerta las coordenadas a continuación.

¿Cuantos arcos no pertenecen al circuito?            3  
Dane las coordenadas

[ 1,2 ]  
[ 2,4 ]  
[ 6,5 ]  
x [1,3] 4  
x [3,6] 4  
x [6,7] 4  
x [7,1] 9

PRESIONE [ENTER] PARA CONTINUAR



A continuación se presenta la tabla donde se muestra a que conjunto pertenece cada arco.

- El arco 1, 2 Es Conformable y pertenece a B
- El arco 1, 3 Es Conformable y pertenece a B
- El arco 2, 3 Es Conformable y pertenece a B
- El arco 2, 4 Es Conformable y pertenece a B
- El arco 2, 5 Es Conformable y pertenece a B
- El arco 3, 4 Es Conformable y pertenece a B
- El arco 3, 6 Es Conformable y pertenece a B
- El arco 4, 5 Es Conformable y pertenece a B
- El arco 4, 6 Es Conformable y pertenece a B
- El arco 5, 7 Es Conformable y pertenece a B
- El arco 6, 5 Es Conformable y pertenece a B
- El arco 6, 7 Es Conformable y pertenece a B
- El arco 7, 1 Es Conformable y pertenece a B

PRESIONE [ENTER] PARA CONTINUAR

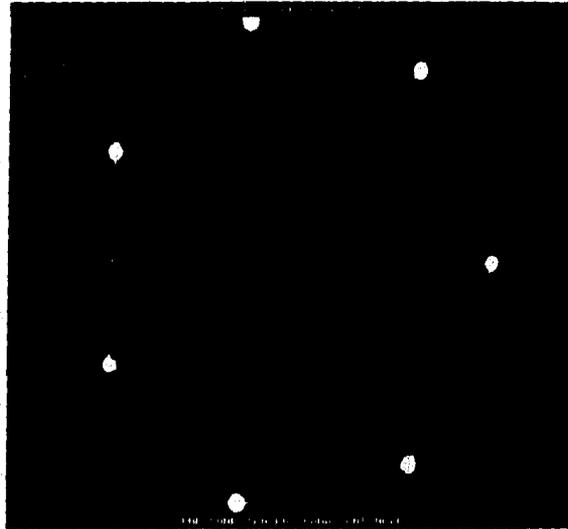
SE HA OBTENIDO LA SOLUCION OPTIMA

	1	2	3	4	5	6	7
1	0	5	4	0	0	0	0
2	0	0	0	1	4	0	0
3	0	0	0	0	0	4	0
4	0	0	0	0	1	0	0
5	0	0	0	0	0	0	5
6	0	0	0	0	0	0	4
7	9	0	0	0	0	0	0

Como deseas que se llame el archivo de resultados : RES12

EL COSTO DEL FLUJO MAXIMO A COSTO MINIMO DE LA RED ES: 0

PRESIONE [ENTER] PARA CONTINUAR



#### MENU DE METODOS

- 1.- FLUJO MAXIMO DE LA RED
- 2.- DIGRAFICA DE LA RED
- 3.- FLUJO MAXIMO A COSTO MINIMO EN UNA RED
- 4.- MODELO PRIMAL
- 5.- MENU PRINCIPAL

DAME LA OPCION DESEADA : 4

CUAL ES EL NOMBRE DEL ARCHIVO :EJER1

PRESIONE [ENTER] PARA CONTINUAR

#### MODELO PRIMAL DE LA RED

$$\text{MIN } Z = +(0)(X12) + (0)(X13) + (0)(X23) + (0)(X24) + (0)(X25) + (0)(X34) + (0)(X$$

$$X_{45} + X_{25} + X_{65} - X_{57} = 0$$

$$X_{46} + X_{36} - X_{65} - X_{67} = 0$$

$$X_{57} + X_{67} - X_{71} = 0$$

$$X_{65} + X_{25} + X_{45} - X_{57} = 0$$

$$X_{67} + X_{57} - X_{71} = 0$$

$$X_{71} - X_{12} - X_{13} = 0$$

$$X_{12} \geq 0 \quad -X_{12} \leq -6$$

$$X_{13} \geq 0 \quad -X_{13} \leq -7$$

PRESSIONE [ENTER] PARA CONTINUAR

$$X_{23} \geq 0 \quad -X_{23} \leq -1$$

$$X_{24} \geq 0 \quad -X_{24} \leq -3$$

$$X_{25} \geq 0 \quad -X_{25} \leq -4$$

$$X_{34} \geq 0 \quad -X_{34} \leq -2$$

$$X_{36} \geq 0 \quad -X_{36} \leq -5$$

$$X_{45} \geq 0 \quad -X_{45} \leq -1$$

$$X_{46} \geq 0 \quad -X_{46} \leq -2$$

$$X_{57} \geq 0 \quad -X_{57} \leq -5$$

$$X_{65} \geq 0 \quad -X_{65} \leq -2$$

$$X_{67} \geq 0 \quad -X_{67} \leq -4$$

$$X_{71} \geq 0 \quad -X_{71} \leq -9$$

$$X_{(I,J)} \in \mathbb{Z}$$

PRESSIONE [ENTER] PARA CONTINUAR

El siguiente ejemplo tiene 4 vértices.

CAPTURA DE DATOS

- 1.- LECTURA DE DATOS
- 2.- MODIFICACION DE DATOS
- 3.- IMPRESION DE MATRIZ DE COSTOS Y DE CAPACIDADES
- 4.- MATRIZ DE RESULTADOS
- 5.- MENU DE METODOS

DAME LA OPCION DESEADA : 3

CUAL ES EL NOMBRE DEL ARCHIVO : EJER2

PRESIONE [ENTER] PARA CONTINUAR

---

MATRIZ DE COSTOS

---

	1	2	3	4
1	0	2	5	0
2	0	0	-3	0
3	0	0	0	-1
4	0	0	0	0

PRESIONE [ENTER] PARA CONTINUAR

---

MATRIZ DE CAPACIDADES MINIMAS

---

	1	2	3	4
1	0	0	1	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0

PRESIONE [ENTER] PARA CONTINUAR

MATRIZ DE CAPACIDADES MAXIMAS

1 2 3 4  
1 0 2 2 0  
2 0 0 6 0  
3 0 0 0 5  
4 3 0 0 0

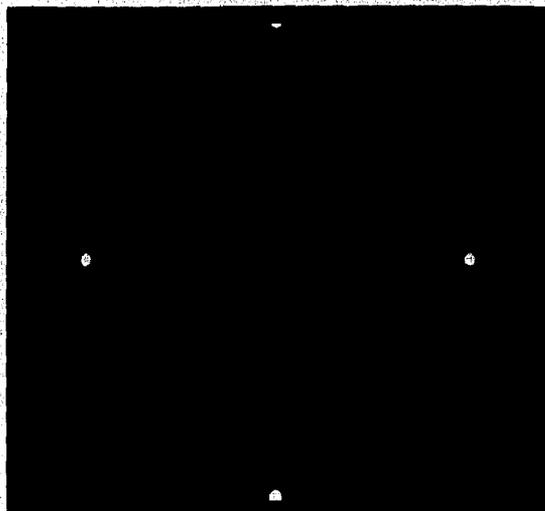
PRESIONE [ENTER] PARA CONTINUAR

MENU DE METODOS

- 1.- FLUJO MAXIMO DE LA RED
- 2.- DIGRAFICA DE LA RED
- 3.- FLUJO MAXIMO A COSTO MINIMO EN UNA RED
- 4.- MODELO PRIMAL
- 5.- MENU PRINCIPAL

DAME LA OPCION DESEADA : 2

CUAL ES EL NOMBRE DEL ARCHIVO :EJER2  
PRESIONE [ENTER] PARA CONTINUAR



MENU DE METODOS

- 1.- FLUJO MAXIMO DE LA RED
- 2.- DIGRAFICA DE LA RED
- 3.- FLUJO MAXIMO A COSTO MINIMO EN UNA RED
- 4.- MODELO PRIMAL
- 5.- MENU PRINCIPAL

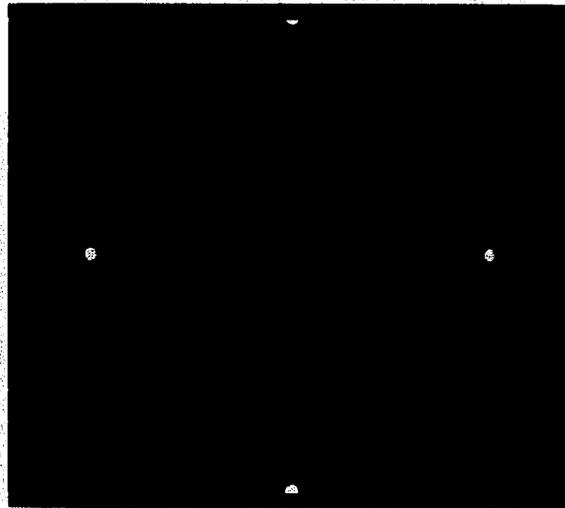
DAME LA OPCION DESEADA : 1

CUAL ES EL NOMBRE DEL ARCHIVO :EJER2  
PRESIONE [ENTER] PARA CONTINUAR

VALORES DE GAMA E INCREMENTO DEL FLUJO  
la corte inicial es 5 y la corte final es 4

GAMA[3] : 2  
GAMA[4] : 2  
x [1,3] 2  
x [3,4] 2

PRESIONE [ENTER] PARA CONTINUAR

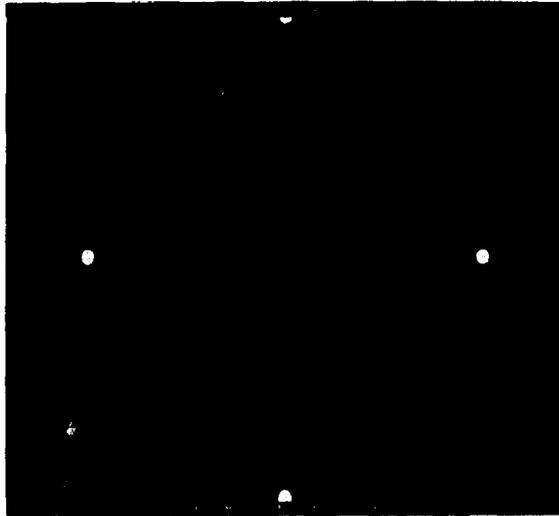


VALORES DE GAMA E INCREMENTO DEL FLUJO

GAMA[2] : 2  
GAMA[3] : 2  
GAMA[4] : 2

x [1,2] 2  
x [2,3] 2  
x [3,4] 4

PRESIONE [ENTER] PARA CONTINUAR

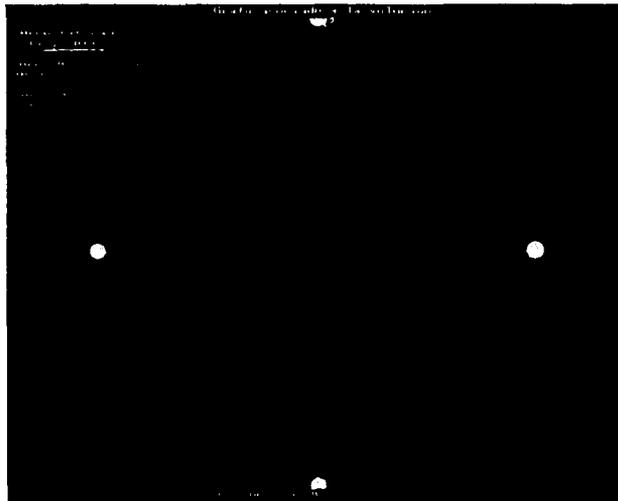


El Flujo Maximo es: 4

	1	2	3	4	
1	0	2	2	0	
2	0	0	2	0	
3	0	0	0	4	
4	4	0	0	0	

Como deseas que se llame el archivo de resultados : RES21

PRESIONE [ENTER] PARA CONTINUAR



**MENU DE METODOS**

- 1.- FLUJO MAXIMO DE LA RED
- 2.- DIGRAFICA DE LA RED
- 3.- FLUJO MAXIMO A COSTO MINIMO EN UNA RED
- 4.- MODELO PRIMAL
- 5.- MENU PRINCIPAL

DAME LA OPCION DESEADA : 3

CUAL ES EL NOMBRE DEL ARCHIVO :EJER2  
PRESIONE [ENTER] PARA CONTINUAR

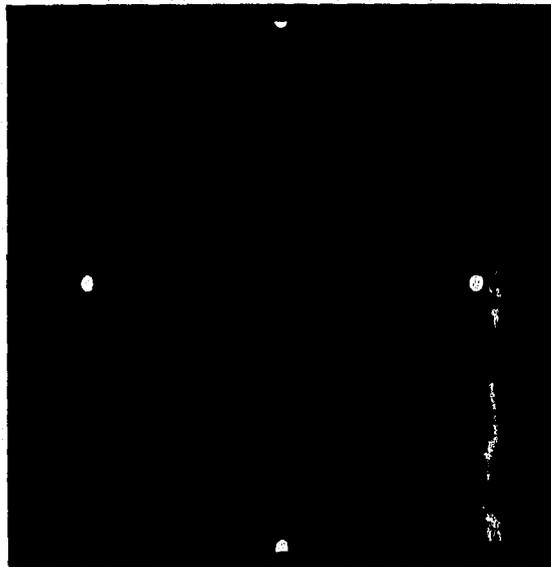
A continuación se presenta la tabla donde se muestra a que conjunto pertenece cada arco.

- El arco 1, 2 Es Conformable y pertenece a B
- El arco 1, 3 No es Conformable y pertenece a B1
- El arco 2, 3 Es Conformable y pertenece a B
- El arco 3, 4 Es Conformable y pertenece a B
- El arco 4, 1 No es Conformable y pertenece a B1

PRESIONE [ENTER] PARA CONTINUAR

	VI	1	2	3	4	
VI	UI	UI	2	-2	1	2
1	2	0-2	0-2			$[-, \infty]$
		0-2	-1	4		
2	-2		0-6			$[1, 2]$
			0	0		
3	1			0-5		$[1, 1]$
				0	0	
4	2	0-4				$[3, 1]$
		-4	0			

PRESIONE [ENTER] PARA CONTINUAR



Por favor, si el arco no se encuentra en el circuito  
incerta las coordenadas a continuación.

¿Cuántos arcos no pertenecen al circuito?      1  
Dame las coordenadas

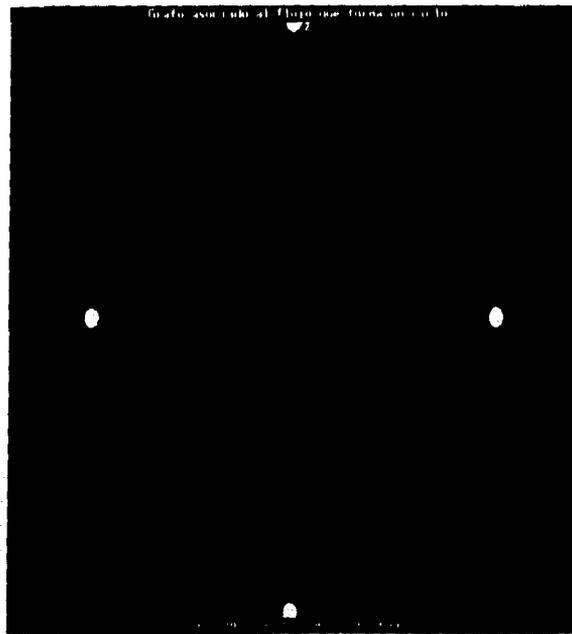
$[1, 2]$

x  $[1, 3]$  1

x  $[3, 4]$  1

x  $[4, 1]$  1

PRESIONE [ENTER] PARA CONTINUAR



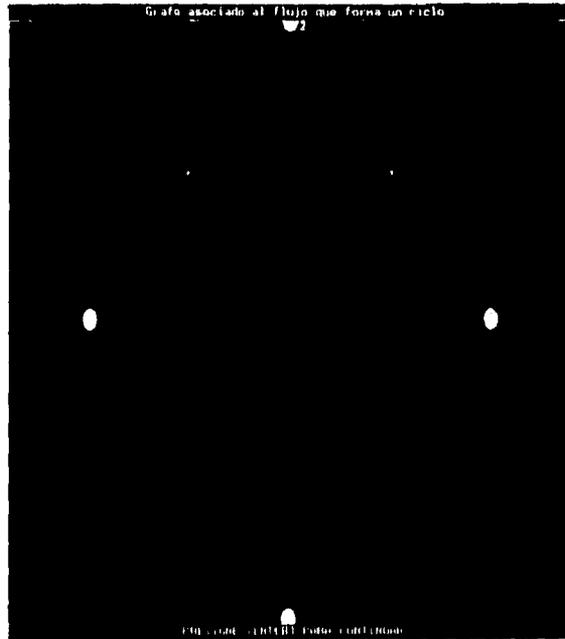
A continuación se presenta la tabla donde se muestra a que conjunto pertenece cada arco.

- El arco 1, 2 No es Conformable y pertenece a K1
- El arco 1, 3 es Conformable y pertenece a L
- El arco 2, 3 Es Conformable y pertenece a B
- El arco 3, 4 Es Conformable y pertenece a B
- El arco 4, 1 No es Conformable y pertenece a B1

PRESIONE [ENTER] PARA CONTINUAR

VI	1	2	3	4
VI UI	UI	2	-2	1 2
1 2		0 -2	1 -1	[-,∞]
		0 -2	0 4	
2 -2			0 -6	[1,2]
			0 0	
3 1			1 -4	[2,2]
			0 0	
4 2		1 -3		[3,2]
		-3 0		

PRESIONE [ENTER] PARA CONTINUAR

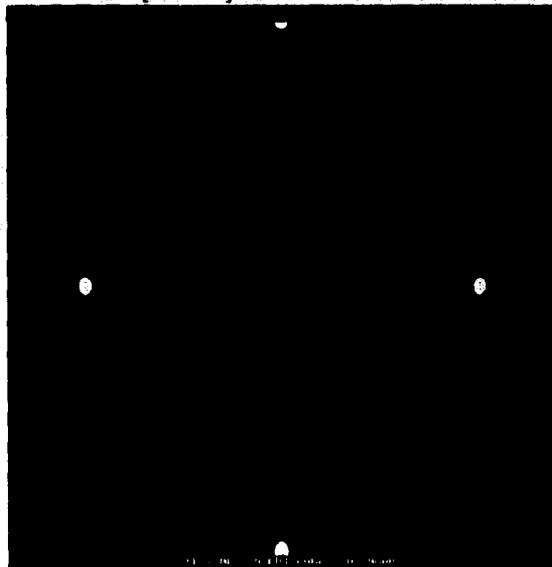


Por favor, si el arco no se encuentra en el circuito  
incerta las coordenadas a continuación.

¿Cuantos arcos no pertenecen al circuito?            0

- x [1,2] 2
- x [2,3] 2
- x [3,4] 3
- x [4,1] 3

**PRESIONE [ENTER] PARA CONTINUAR**



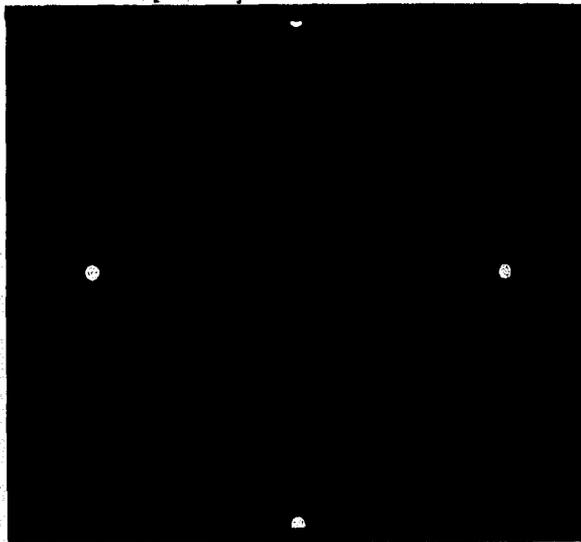
A continuación se presenta la tabla donde se muestra a que conjunto pertenece cada arco.

- El arco 1, 2 Es Conformable y pertenece a K
- El arco 1, 3 es Conformable y pertenece a L
- El arco 2, 3 Es Conformable y pertenece a B
- El arco 3, 4 Es Conformable y pertenece a B
- El arco 4, 1 No es Conformable y pertenece a B1

PRESIONE [ENTER] PARA CONTINUAR

VI	1	2	3	4
VI UI	UI	2	-2	1 2
1 2		2 0	1 -1	$[-, \infty]$
		2 -2	0 4	
2 -2			2 -4	
			2 0	
3 1				3 -2
				3 0
4 2		3 -1		
		-1 0		

PRESIONE [ENTER] PARA CONTINUAR



CAMBIO DE POTENCIALES

Vertices etiquetados { 1 }

Vertices No etiquetados { 2 3 4 }

S1 = { [ 1,3 ] }

S2 = { }

El conjunto S2 es vacío

El mínimo del conjunto S1 es : 4

El valor de Teta es igual a: 4

PRESIONE [ENTER] PARA CONTINUAR

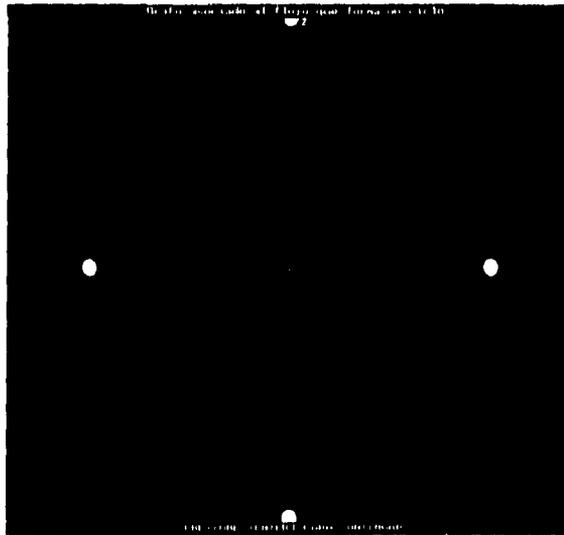
A continuación se presenta la tabla donde se muestra a que conjunto pertenece cada arco.

- El arco 1, 2 Es Conformable y pertenece a K
- El arco 1, 3 Es Conformable y pertenece a B
- El arco 2, 3 Es Conformable y pertenece a B
- El arco 3, 4 Es Conformable y pertenece a B
- El arco 4, 1 No es Conformable y pertenece a L1

PRESIONE [ENTER] PARA CONTINUAR

VI	1	2	3	4
VI UI	UI	6	-2	1 2
1 2	2 0	1 -1		$[-, \infty]$
	2 -6	0 0		
2 -2		2 -4		
		2 0		
3 1			3 -2	$[1, 1]$
		3 0		
4 2	3 -1			$[3, 1]$
	-1 4			

PRESIONE [ENTER] PARA CONTINUAR

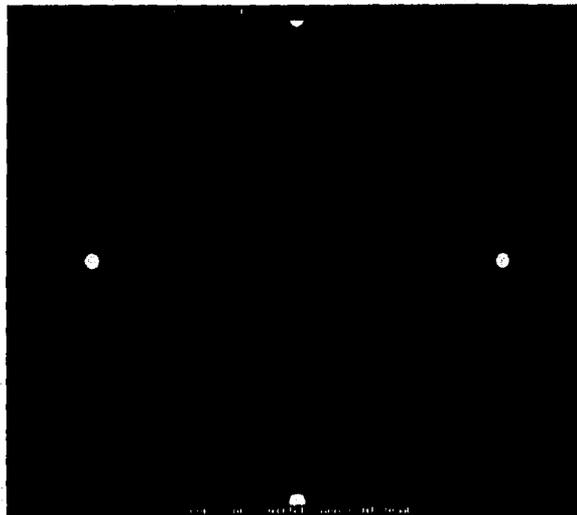


Por favor, si el arco no se encuentra en el circuito  
incerta las coordenadas a continuación.

¿Cuántos arcos no pertenecen al circuito?            0

x [1,3] 2  
x [3,4] 4  
x [4,1] 4

PRESIONE [ENTER] PARA CONTINUAR



A continuación se presenta la tabla donde  
se muestra a que conjunto pertenece cada arco.

El arco 1, 2 Es Conformable y pertenece a K  
El arco 1, 3 Es Conformable y pertenece a B  
El arco 2, 3 Es Conformable y pertenece a B  
El arco 3, 4 Es Conformable y pertenece a B  
El arco 4, 1 es Conformable y pertenece a L

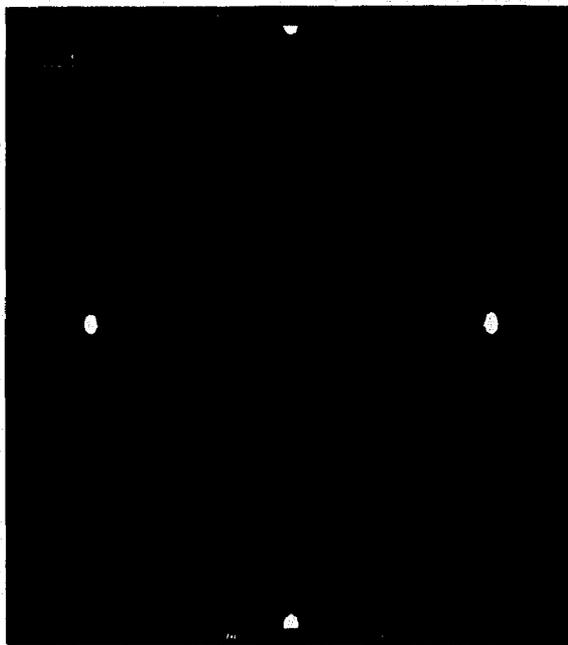
↓  
PRESIONE [ENTER] PARA CONTINUAR

SE HA OBTENIDO LA SOLUCION OPTIMA

I	2	3	4	
1	0	2	2	0
2	0	0	2	0
3	0	0	0	4
4	4	0	0	0

Como desee que se llame el archivo de resultados : RES22  
EL COSTO DEL FLUJO MAXIMO A COSTO MINIMO DE LA RED ES: 4

PRESIONE [ENTER] PARA CONTINUAR



MENU DE METODOS

- 1.- FLUJO MAXIMO DE LA RED
- 2.- DIGRAFICA DE LA RED
- 3.- FLUJO MAXIMO A COSTO MINIMO EN UNA RED
- 4.- MODELO PRIMAL
- 5.- MENU PRINCIPAL

DAME LA OPCION DESEADA: 4

CUAL ES EL NOMBRE DEL ARCHIVO :EJER2

PRESIONE [ENTER] PARA CONTINUAR

MODELO PRIMAL DE LA RED

$$\text{MIN } Z = +(2)(X_{12}) + (5)(X_{13}) + (-3)(X_{23}) + (-1)(X_{34})$$

s.a.

$$X_{12} - X_{23} = 0$$

$$X_{13} + X_{23} - X_{34} = 0$$

$$X_{23} + X_{13} - X_{34} = 0$$

$$X_{34} - X_{41} = 0$$

$$X_{41} - X_{12} - X_{13} = 0$$

$$X_{12} \geq 0 \quad -X_{12} \leq -2$$

$$X_{13} \geq 1 \quad -X_{13} \leq -2$$

$$X_{23} \geq 0 \quad -X_{23} \leq -6$$

$$X_{34} \geq 0 \quad -X_{34} \leq -5$$

$$X_{41} \geq 0 \quad -X_{41} \leq -3$$

$$X_{(I,J)} \in Z$$

PRESIONE [ENTER] PARA CONTINUAR

CAPTURA DE DATOS

- 1.- LECTURA DE DATOS
- 2.- MODIFICACION DE DATOS
- 3.- IMPRESION DE MATRIZ DE COSTOS Y DE CAPACIDADES
- 4.- MATRIZ DE RESULTADOS

DAME LA OPCION DESEADA : 3

CUAL ES EL NOMBRE DEL ARCHIVO :EJER3

PRESIONE [ENTER] PARA CONTINUAR

---

MATRIZ DE COSTOS

---

	1	2	3	4	5
1	0	1	0	0	0
2	0	0	5	0	0
3	0	0	0	-3	0
4	0	0	0	0	-1
5	0	3	0	0	0

PRESIONE [ENTER] PARA CONTINUAR

---

MATRIZ DE CAPACIDADES MINIMAS

---

	1	2	3	4	5
1	0	2	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	2
5	0	0	0	0	0

PRESIONE [ENTER] PARA CONTINUAR

---

MATRIZ DE CAPACIDADES MAXIMAS

---

	1	2	3	4	5
1	0	4	0	0	0
2	0	0	3	5	0
3	0	0	0	4	0
4	0	0	0	0	5
5	6	2	0	0	0

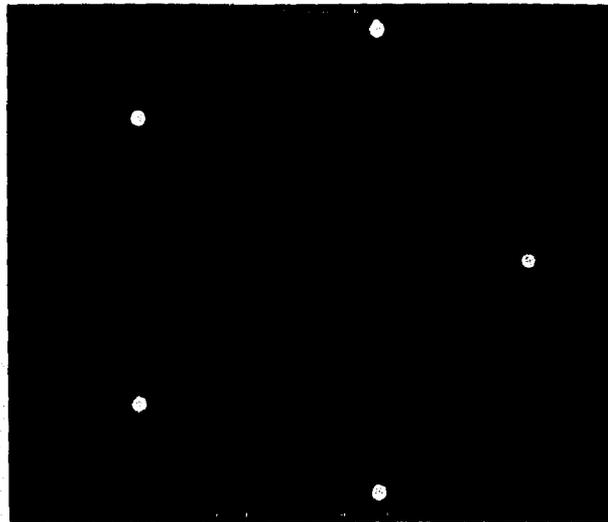
PRESIONE [ENTER] PARA CONTINUAR

MENU DE METODOS

- 1.- FLUJO MAXIMO DE LA RED
- 2.- DIGRAFICA DE LA RED
- 3.- FLUJO MAXIMO A COSTO MINIMO EN UNA RED
- 4.- MODELO PRIMAL
- 5.- MENU PRINCIPAL

DAME LA OPCION DESEADA : 2

CUAL ES EL NOMBRE DEL ARCHIVO :EJER3  
PRESIONE [ENTER] PARA CONTINUAR



MENU DE METODOS

- 1.- FLUJO MAXIMO DE LA RED
- 2.- DIGRAFICA DE LA RED
- 3.- FLUJO MAXIMO A COSTO MINIMO EN UNA RED
- 4.- MODELO PRIMAL
- 5.- MENU PRINCIPAL

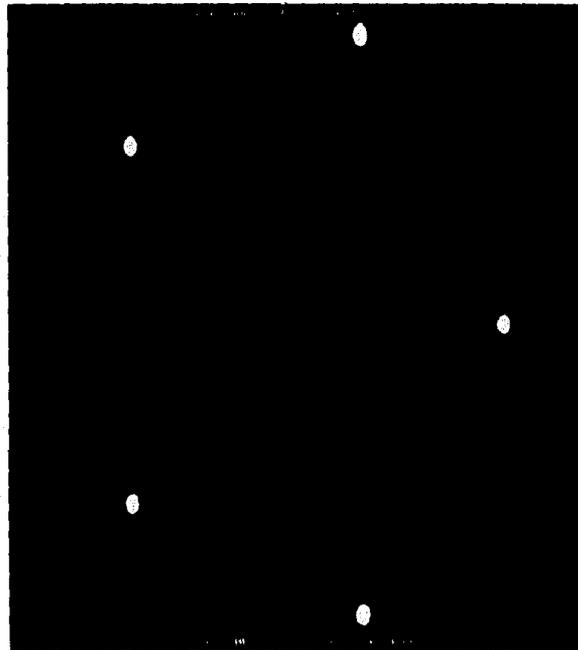
DAME LA OPCION DESEADA : 1

CUAL ES EL NOMBRE DEL ARCHIVO :EJER3  
PRESIONE [ENTER] PARA CONTINUAR

VALORES DE GAMA E INCREMENTO DEL FLUJO  
la corte inicial es 5 y la corte final es 4

GAMA[2] : 4  
GAMA[4] : 4  
GAMA[5] : 4  
x [1,2] 4  
x [2,4] 4  
x [4,5] 4

PRESIONE [ENTER] PARA CONTINUAR

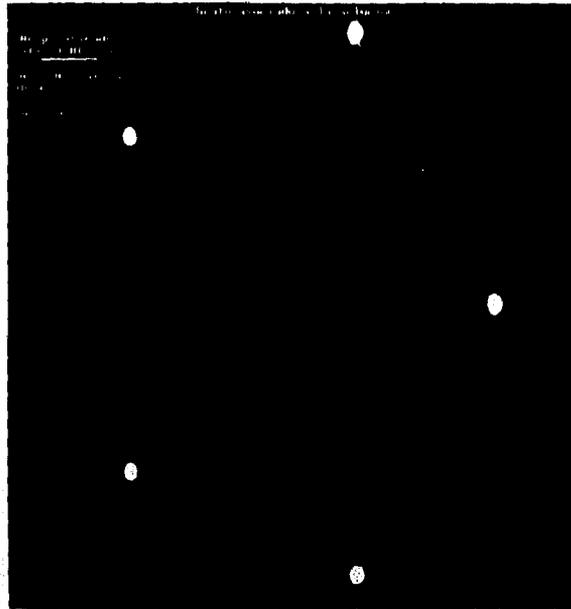


El Flujo Maximo es: 4

	1	2	3	4	5
1	0	4	0	0	0
2	0	0	0	4	0
3	0	0	0	0	0
4	0	0	0	0	4
5	4	0	0	0	0

Como descas que se llame el archivo de resultados : RES31

PRESIONE [ENTER] PARA CONTINUAR



**MENU DE METODOS**

- 1.- FLUJO MAXIMO DE LA RED
- 2.- DIGRAFICA DE LA RED
- 3.- FLUJO MAXIMO A COSTO MINIMO EN UNA RED
- 4.- MODELO PRIMAL
- 5.- MENU PRINCIPAL

DAME LA OPCION DESEADA : 3

CUAL ES EL NOMBRE DEL ARCHIVO :EJER3  
PRESIONE [ENTER] PARA CONTINUAR

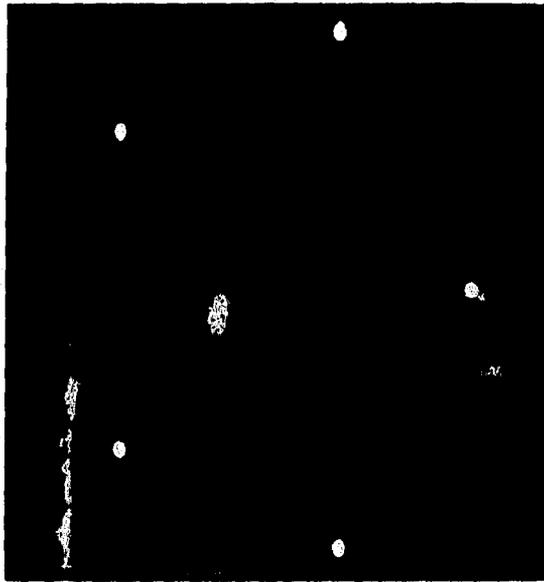
A continuación se presenta la tabla donde se muestra a que conjunto pertenece cada arco.

- El arco 1, 2 No es Conformable y pertenece a B1
- El arco 2, 3 Es Conformable y pertenece a B
- El arco 2, 4 Es Conformable y pertenece a B
- El arco 3, 4 Es Conformable y pertenece a B
- El arco 4, 5 No es Conformable y pertenece a B1
- El arco 5, 1 No es Conformable y pertenece a B1
- El arco 5, 2 Es Conformable y pertenece a B

PRESIONE [ENTER] PARA CONTINUAR

	VI	1	2	3	4	5
VI UI	UI	0	-1	-6	-1	0
1	0		0 4			[-5,4]
			-2 0			
2	-1			0 -3 0 -5		[1,4]
				0 0 0 0		
3	1				0 -4	[2,3]
					0 2	
4	1					0 -5 [3,1]
						-2 0
5	0	0 -4 0 -2				[4,4]
		4 0 0 2				

PRESIONE [ENTER] PARA CONTINUAR



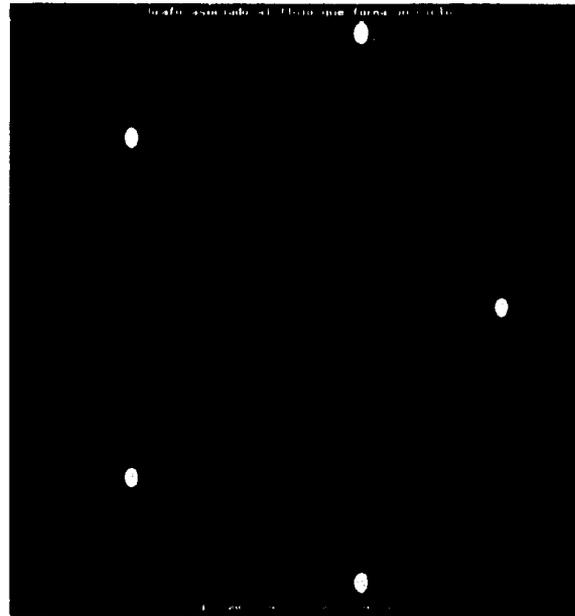
Por favor, si el arco no se encuentra en el circuito  
incerta las coordenadas a continuación.

¿Cuántos arcos no pertenecen al circuito? 1  
Dame las coordenadas

[ 2,3 ]

x [1,2] 4  
x [2,4] 4  
x [4,5] 4  
x [5,1] 4

PRESIONE [ENTER] PARA CONTINUAR



A continuación se presenta la tabla donde se muestra a que conjunto pertenece cada arco.

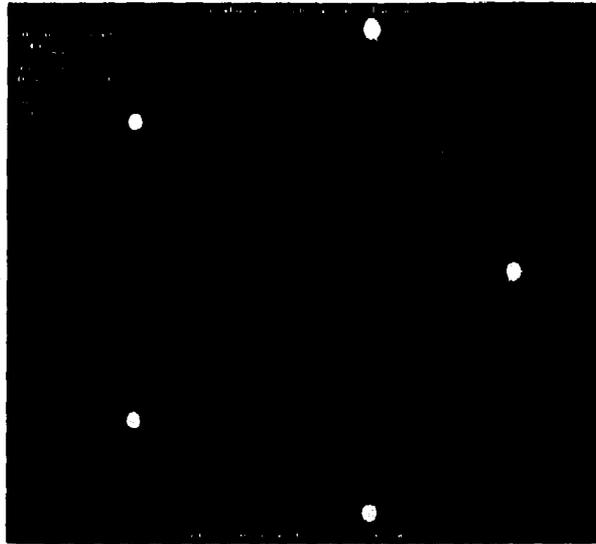
- El arco 1, 2 Es Conformable y pertenece a B
- El arco 2, 3 Es Conformable y pertenece a B
- El arco 2, 4 Es Conformable y pertenece a B
- El arco 3, 4 es Conformable y pertenece a L
- El arco 4, 5 Es Conformable y pertenece a B
- El arco 5, 1 Es Conformable y pertenece a B
- El arco 5, 2 es Conformable y pertenece a L

PRESIONE [ENTER] PARA CONTINUAR

SE HA OBTENIDO LA SOLUCION OPTIMA

	1	2	3	4	5
1	0	4	0	0	0
2	0	0	0	4	0
3	0	0	0	0	0
4	0	0	0	0	4
5	4	0	0	0	0

Como deseas que se llame el archivo de resultados : RES32  
 EL COSTO DEL FLUJO MAXIMO A COSTO MINIMO DE LA RED ES: 0  
 PRESIONE [ENTER] PARA CONTINUAR



MENU DE METODOS

- 1.- FLUJO MAXIMO DE LA RED
- 2.- DIGRAFICA DE LA RED
- 3.- FLUJO MAXIMO A COSTO MINIMO EN UNA RED
- 4.- MODELO PRIMAL
- 5.- MENU PRINCIPAL

DAME LA OPCION DESEADA : 4

CUAL ES EL NOMBRE DEL ARCHIVO : EJER3  
PRESIONE [ENTER] PARA CONTINUAR

MODELO PRIMAL DE LA RED

$$\text{MIN } Z = +(1)(X_{12}) + (5)(X_{23}) + (0)(X_{24}) + (-3)(X_{34}) + (-1)(X_{45})$$

s.a.

$$\begin{aligned} X_{12} + X_{52} - X_{23} - X_{24} &= 0 \\ X_{23} - X_{34} &= 0 \\ X_{24} + X_{34} - X_{45} &= 0 \\ X_{34} + X_{24} - X_{45} &= 0 \\ X_{45} - X_{51} - X_{52} &= 0 \\ X_{51} - X_{12} &= 0 \end{aligned}$$

$$\begin{aligned} X_{52} + X_{12} - X_{23} - X_{24} &= 0 \\ X_{12} &\geq 2 & -X_{12} &\leq -4 \\ X_{23} &\geq 0 & -X_{23} &\leq -3 \\ X_{24} &\geq 0 & -X_{24} &\leq -5 \\ X_{34} &\geq 0 & -X_{34} &\leq -4 \\ X_{45} &\geq 2 & -X_{45} &\leq -5 \\ X_{51} &\geq 0 & -X_{51} &\leq -6 \\ X_{52} &\geq 0 & -X_{52} &\leq -2 \end{aligned}$$

$$X(i,j) \in \mathbb{Z}$$

PRESIONE [ENTER] PARA CONTINUAR

**CAPITULO V.**

**COMENTARIOS Y CONCLUSIONES**

## V. COMENTARIOS Y CONCLUSIONES

Con este trabajo se logró la creación de un paquete el cuál da solución a problemas de redes los cuáles cuentan con capacidades máximas, mínimas y costos involucrados a cada arista; mediante el método de las desviaciones. Así mismo como la solución del flujo máximo.

El paquete tiene las siguientes:

### Ventajas.

- Minimización de tiempo y visualización iteración por iteración.
- Permite el almacenamiento de los datos.
- Da la opción de poder cambiar los costos y capacidades mínimas y máximas de flujo.  
Lo cuál permite manipular los datos y tener diferentes soluciones del mismo problema ya sea con diferentes costos o capacidades asociadas.
- Presenta un análisis de los arcos, diciendo a que conjunto pertenece cada uno de ellos.
- Presenta el grafo asociado a cada iteración.
- Permite redes de 3 a 10 vértices, con mas vértices se debe cambiar de 10 al número de vértices en el programa original.
- Aplicando el método a una red y al obtener la solución óptima, el paquete permite guardar la solución asociada a dicha red en disco flexible, para después no tener que aplicar el método nuevamente, a menos que se desee ver las iteraciones asociadas.
- Dado que el cálculo de potenciales se hace por el método de Ford y Fulkerson estos pueden ser negativos o positivos, en cambio si empezamos con los potenciales iguales a cero y puesto que al calcular teta siempre se incrementarán los potenciales, y por tanto no existe un decremento, puede llevar el algoritmo a encontrar una solución no óptima o a converger muy lentamente, por eso se le dan dos opciones al usuario; la primera es utilizar los potenciales igual a cero para el método o que el programa calcule dichos valores.

- El método de las desviaciones también puede calcular el flujo máximo, solo que la matriz de costos debe ser una matriz con elementos igual a cero.

#### Desventajas

- El programa sólo calcula dos cortes, los cuáles sirven para dar un tope en la capacidad máxima del vértice sumidero. Entonces los dos cortes son:

$$1) W = \{\text{vértice origen}\} \quad \bar{W} = \{i/i=2, \dots, n\}$$

$$2) W = \{i/i=1, \dots, n-1\} \quad \bar{W} = \{n\}$$

entonces, puede existir un corte menor a los anteriores y puede que el método no converja o se obtengan soluciones erróneas. Para evitar esto debe ver en las tablas posteriores cuál fue el flujo máximo que va del nodo sumidero al origen y de acuerdo a esto decrementar las capacidades máximas de los nodos que salen del origen o bien de los nodos que llegan al sumidero tales que sea igual al último flujo obtenido y volver a aplicar matriz de costos o las matrices de capacidades y se aplicará el método.

- Si el número de nodos y aristas crece, ocasiona que la matriz de soluciones sea más densa y pueda ocasionarse un error.

Se propone también que dicho paquete sea implantado o instalado en el centro de cómputo para que así el alumno pueda hacer uso de él y el maestro de Optimización Combinatoria pueda dar este método sin pérdida innecesaria de tiempo y esfuerzo.

En este programa se logró que graficar la solución, lo cual otorga una mejor visualización de los resultados que se van obteniendo iteración a iteración, proporcionando una herramienta didáctica la cual explica el método de las desviaciones detalladamente, y esto ocasiona que el usuario comprenda mejor el tema.

## GLOSARIO

**CARÁCTER:**

Cada uno de los elementos de un conjunto, que pueden disponerse en grupos ordenados para expresar información. Cada carácter posee dos formas:(1) la forma inteligible para las personas, la gráfica, que comprende los dígitos decimales 0-9, las letras A-Z, los signos de puntuación y otros símbolos de estructuración y control, y (2) la forma inteligible para las computadoras, el código, que consiste en un de dígitos binarios (bits).

**PROGRAMA :**

Plan para solucionar un problema. Un programa completo comprende planes para la transcripción de los datos, codificación por la computadora y para la asimilación de los resultados de la organización (La lista de instrucciones codificadas se llama rutina). Planear una operación de computadora o un proceso, a partir de la formulación de una pregunta, hasta la entrega del resultado, incluyendo la integración de la operación en un sistema existente. Así, la programación consiste en planear y codificar, incluyendo el análisis numérico, el de sistemas, la especificación de formatos impresos y cualquier otra función necesaria para la integración en una organización computarizada.

## BIBLIOGRAFIA.

---

Efectivamente, una rutina ejecutiva es parte de la misma máquina (sinónimo de sistema del supervisor y programa supervisor).

**SISTEMA OPERATIVO :** Una colección integrada de rutina de servicio para supervisar el secuenciamiento y procesamiento de programas por parte de una computadora.

**VARIABLE :** Cualquier factor o condición que se puede medir, modificar y controlar (p.e., temperatura, presión, caudal, nivel de líquido, humedad, peso, composición química, calor, etc.). Una magnitud que puede tomar cualquiera de un conjunto dado de valores.

## ANEXO A

## A.1. GRÁFICA CONEXA Y FUERTEMENTE CONEXA

Gráfica conectada o conexa: Una gráfica esta conectada si podemos alcanzar un vértice  $v_i$ , a partir de cualquier otro vértice  $v_j$  desplazándonos a través de las líneas.

Formalmente una gráfica  $G$  se dice que esta conectada si hay por lo menos una trayectoria entre cada par de vértices de otra manera  $G$  es desconectada.

Teorema.- Si una gráfica tiene exactamente 2 vértices de grado impar debería existir una trayectoria.

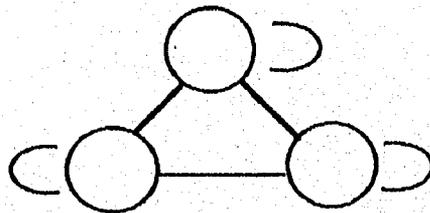
Una gráfica simple con  $n$  vértices y  $k$  componentes o conjuntos, puede tener a lo más

$$\frac{(n-k)(n-k+1)}{2} \text{ aristas o líneas.}$$

Se le llama grafo completo a un grafo simple en el que cualquier par de vértices son adyacentes.

Un grafo completo de  $n$  vértices se designa  $K_n$ , tiene exactamente  $\frac{(n)(n-1)}{2}$  aristas.

Una gráfica es fuertemente conexa si un vértice  $v_i$  puede pasarse a cualquier vértice  $v_j$ , utilizando sólo una arista, es decir, entre cada par de vértices hay una trayectoria o circuito de longitud uno. Ver la fig. 1.



El número de aristas exactas en un grafo fuertemente conexo esta dado por  $\frac{(n)(n+1)}{2}$

## A.2. GRÁFICA BIPARTIDA O BIPARTITA.

Si un conjunto de vértices de un grafo  $G$  puede ser dividido en 2 subconjuntos  $V_1$  y  $V_2$ , de tal forma que cada arista de  $G$  una un vértice del conjunto  $V_1$  con un vértice del conjunto  $V_2$ , entonces  $G$  es un grafo bipartido, el cual se denota como  $G(V_1, V_2)$

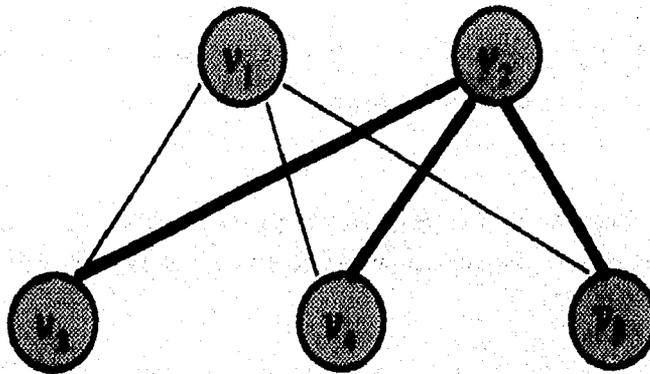
Entonces una gráfica bipartida completa es una gráfica donde todos los vértices de  $V_1$  son adyacentes a los elementos de  $V_2$ . Este grafo se denota por  $K_{r,s}$  donde  $r$  es el número de vértices en  $V_1$  y  $s$  es el número de vértices en  $V_2$ . Este tipo de gráficas son utilizadas para modelar problemas de asignación de recursos, servicios, etc.

Ejemplo:

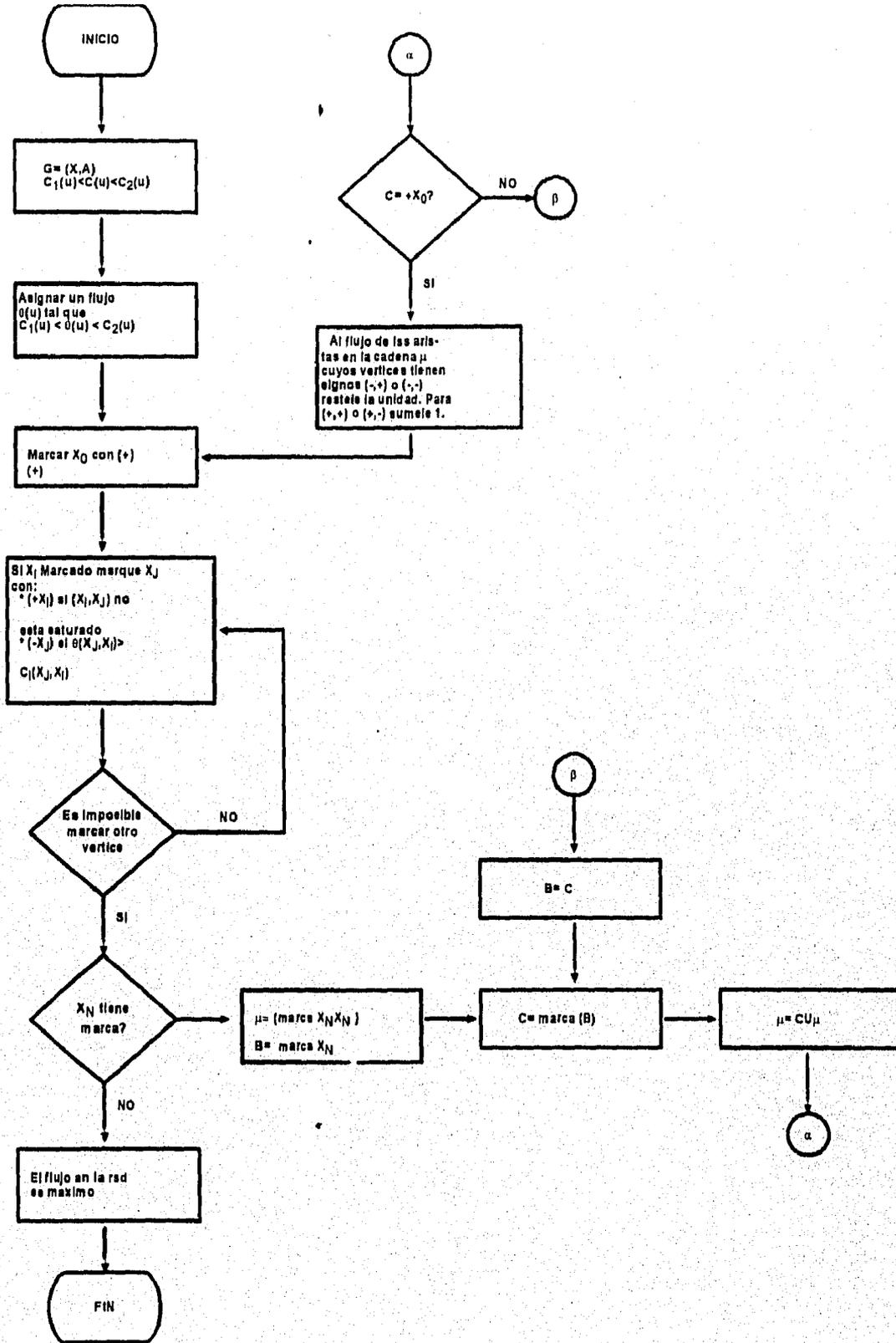
$$V = \{v_1, v_2, v_3, v_4, v_5\}$$

$$V_1 = \{v_1, v_2\}$$

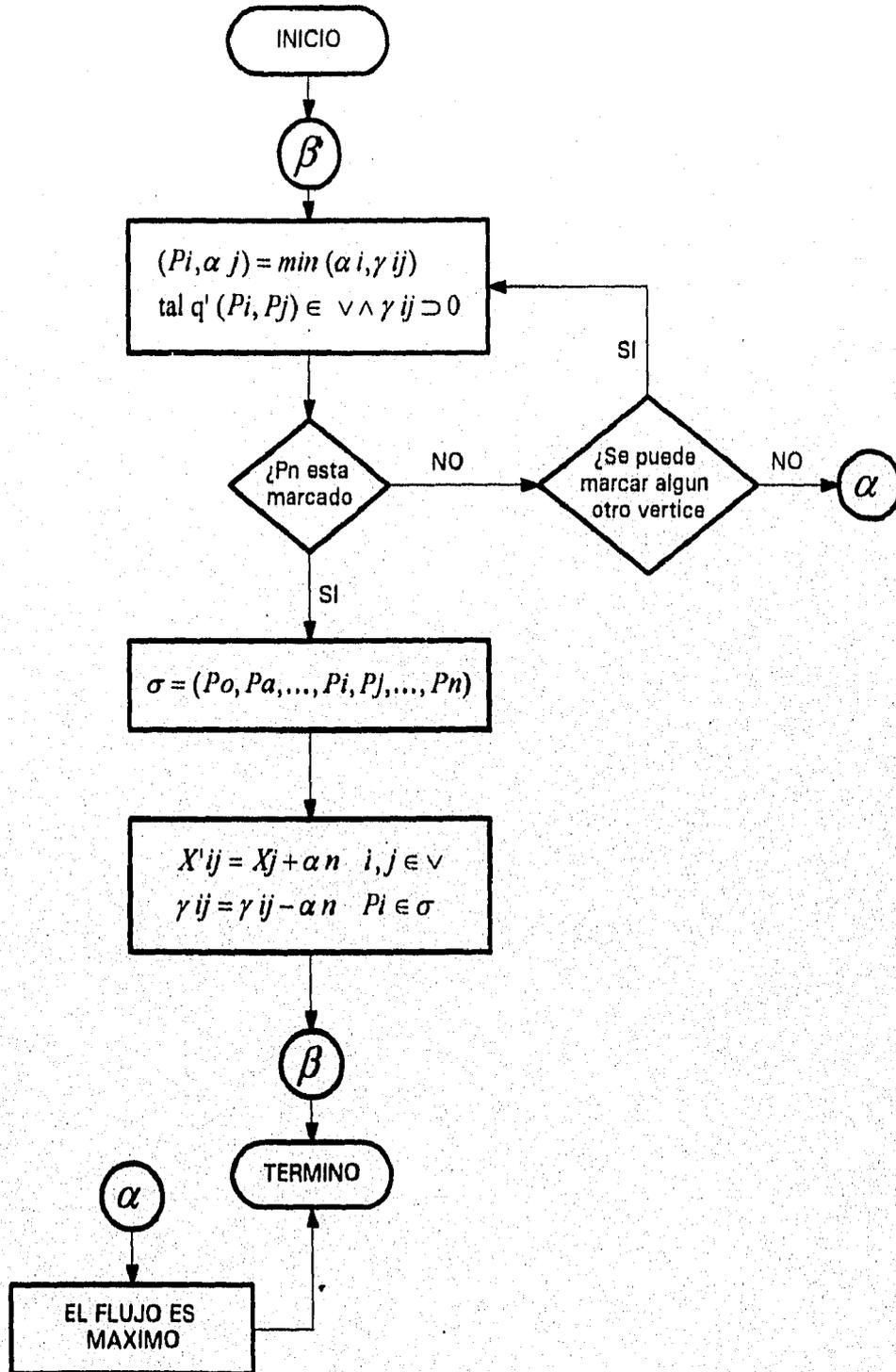
$$V_2 = \{v_3, v_4, v_5\}$$



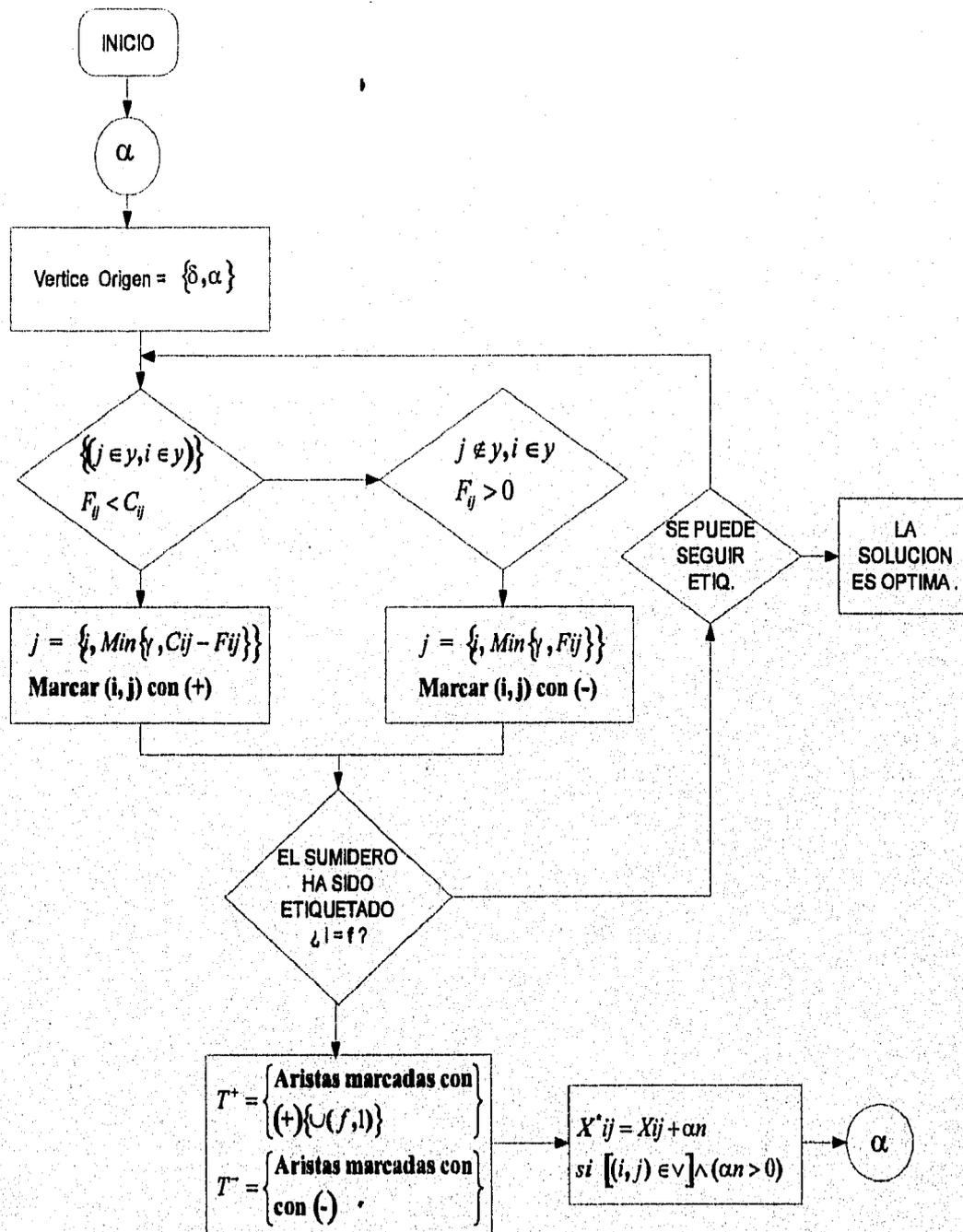
ANEXO B. (B.1)

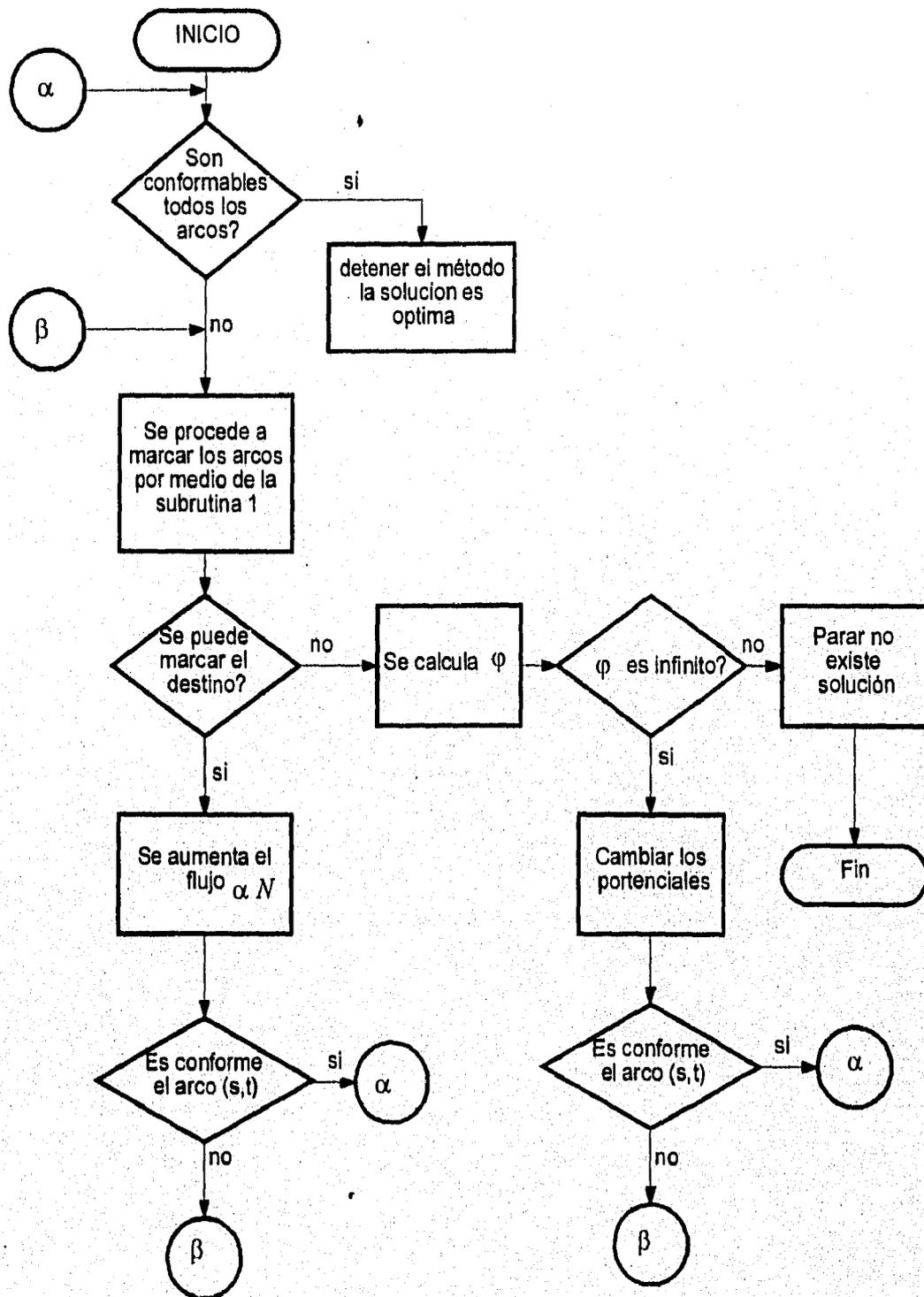


ANEXO B.2



## ANEXO B.3





## ANEXO D.

```

UNIT PORTADA;
INTERFACE
USES CRT,unidad;
PROCEDURE PORRR;
IMPLEMENTATION

```

```

PROCEDURE PORRR;
PROCEDURE PORTADA_PRINCIPAL;

```

```

VAR
  I,K:INTEGER;
BEGIN
  venta (CLRSCR);
  GOTOXY(10,5);
  WRITELN('000 0000 00 000000000000 00000 00000 ');
  GOTOXY(10,6);
  DELAY(500);
  WRITELN('000 0000000000000000 000000 000000 000000000000000 ');
  GOTOXY(10,7);
  DELAY(500);
  WRITELN(' 00000000000000 00 0000 000 000 000 ');
  DELAY(500);
  FOR I:=1 TO 30 DO
  BEGIN
    GOTOXY(I,10);
    WRITE('p');
    DELAY(20);
  END;
  FOR I:=1 TO 28 DO
  BEGIN
    GOTOXY(I,12);
    WRITE('p');
    DELAY(20);
  END;
  FOR I:=1 TO 24 DO
  BEGIN
    GOTOXY(I,14);
    WRITE('p');
    DELAY(20);
  END;
  FOR I:=1 TO 21 DO
  BEGIN
    GOTOXY(I,16);
    WRITE('p');
    DELAY(20);
  END;
  FOR I:=1 TO 19 DO
  BEGIN
    GOTOXY(I,18);
    WRITE('p');
    DELAY(20);
  END;
  FOR I:=1 TO 17 DO
  BEGIN
    GOTOXY(I,20);
    WRITE('p');
    DELAY(20);
  END;
  FOR I:=10 TO 20 DO
  BEGIN
    GOTOXY(38-I,I);

```

```
WRITE('PÚÚ');
GOTOXY(30,1);
WRITE('ÚÚ');
GOTOXY(50-1,1);
WRITE('PÚÚ');
GOTOXY(42,1);
WRITE('ÚÚ ÚÚ');
GOTOXY(37+1,1);
WRITE('PÚÚ ÝÝ');
DELAY(100);
END;
FOR I:=45 TO 55 DO
BEGIN
GOTOXY(1,18);
WRITE('Ú');
DELAY(100);
END;
FOR I:=1 TO 15 DO
BEGIN
GOTOXY(52+I,10);
WRITE('Ý');
GOTOXY(52+I,11);
WRITE('Ý');
GOTOXY(60+I,19);
WRITE('Ý');
GOTOXY(60+I,20);
WRITE('Ý');
DELAY(100);
END;
DELAY(2000);
END;

PROCEDURE PORTADA;
VAR
Q:INTEGER;
BEGIN
VENTA;
WRITE('É');
FOR Q:=2 TO 78 DO
WRITE('Í');
WRITE('Í');
WRITELN;
FOR Q:=1 TO 23 DO
WRITELN(' :77,');
WRITE('É');
FOR Q:=2 TO 78 DO
WRITE('Í');
WRITE('¼');
FOR Q:=2 TO 78 DO
BEGIN
GOTOXY(Q,5);
WRITE('Í');
END;
GOTOXY(22,2);
WRITE('UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO');
GOTOXY(21,3);
WRITE(' E N E P');
GOTOXY(21,4);
WRITE(' ACATLAN');
GOTOXY(5,4);
WRITE('MATEMATICAS APLICADAS Y COMPUTACION');
GOTOXY(15,10);WRITE('EL PRESENTE PAQUETE HA SIDO ELABORADO CON EL FIN');
GOTOXY(15,11);WRITE('DE ILUSTRAR Y RESOLVER EL PROBLEMA DE FLUJO MAXIMO');
GOTOXY(15,12);WRITE('Y EL PROBLEMA DE FLUJO MAXIMO A COSTO MINIMO, YA ');
GOTOXY(15,13);WRITE('QUE ELIMINA EL TEDIOSO TRABAJO DE RESOLVER ');
gotoxy(15,14);write('MANUALMENTE LOS PROBLEMAS ANTES CITADOS.');
```

```

DELAY(10000);
END;
BEGIN
PORTADA_PRINCIPAL;
PORTADA;
END;
END.

```

## UNIDAD UNIDAD

```

UNIT UNIDAD;
INTERFACE
USES
  CRT;
TYPE
MATRIZ = ARRAY[1..10,1..10] OF INTEGER;
VECTOR1 = ARRAY[1..10] OF INTEGER;
procedure VENTA;
PROCEDURE sonier;
PROCEDURE COMPONER;
PROCEDURE DIGITAR (VAR N1,dig,w:INTEGER);
PROCEDURE AFIRMAR(VAR V:INTEGER);
procedure afirmari(var v,j:integer);
IMPLEMENTATION
PROCEDURE VENTA;
VAR
w:integer;
OrigMode,LastCol,LastRow: Word;
Ch: Char;
procedure Initialize;
{ Initialize the video mode, LastCol, LastRow, and the random number }
{ generator. Paint the help line. }
begin
  CheckBreak:=False;      { turn off Contrl-C checking }
  OrigMode:=LastMode;     { Remember original video mode }
  TextMode(Lo(LastMode)+Font8x8); { use 43 or 50 lines on EGA/VGA }
  LastCol:=Lo(WindMax)+1; { get last column, row }
  LastRow:=Hi(WindMax)+1;
  GoToXY(1,LastRow);      { put message line on screen }
  TextBackground(Black);
  TextColor(White);
  Dec(LastRow,80 div LastCol); { don't write on message line }
end; { Init }
procedure VENTANA;
var
  Xin,Yin,Width,Height: Word;
begin
  Width:=79; Height:=24;
  Xin:=1; Yin:=1;
  Window(Xin,Yin,Xin+Width,Yin+Height);
  if OrigMode = Mono then
  if OrigMode = Mono then
  begin
    TextBackground(White);
    TextColor(Black);
    ClrScr;
    Window(Xin+1,Yin+1,Xin+Width-1,Yin+Height-1);
    TextBackground(Black);
    TextColor(White);
    CLRSCR;
  end
  else
  begin
    TextBackground(1);
    TextColor(9);
  end
end

```

ANEXO

```

clrscr;
window(xin+1,yin+1,xin+width-1,yin+height-1);
textbackground(7);
textcolor(9);
clrscr;
end;

end; { MakeWindow }
begin
  Initialize;
  VENTANA;
END;
PROCEDURE SONIER;
BEGIN
  SOUND(100); DELAY (100);NOSOUND;
END;
PROCEDURE COMPONER;
BEGIN
  WRITELN; SONIER;
  GOTOXY(13,22);WRITE(' PRESIONA LA TECLA CORRECTA');
END;
PROCEDURE DIGITAR (VAR NI,DIG,W:INTEGER);
VAR
  Z : ARRAY[1..15] OF CHAR;
  SON,B,I,J,D,vv,nn :INTEGER;
  CAR : CHAR;
BEGIN
  B:=0; Z[1]:= #48;
  Z[2]:= #49; Z[3]:= #50;
  Z[4]:= #51; Z[5]:= #52;
  Z[6]:= #53; Z[7]:= #54;
  Z[8]:= #55; Z[9]:= #56;
  Z[10]:= #57; Z[11]:= #45;
  SON:=1; if(w>10) and (w<=un+10)then
  vv:=20; vv:=72;
  REPEAT
  CAR:=READKEY;
  CASE DIG OF
  1: BEGIN
  FOR I:=2 TO 5 DO
  IF CAR=Z[I] THEN
  BEGIN
  NI:=I-1; GOTOXY(50,20);WRITE (NI:2);
  GOTOXY(10,22);WRITE(' ');
  B:=1;
  END;
  END;
  2: BEGIN
  FOR I:=2 TO 6 DO
  IF CAR=Z[I] THEN
  BEGIN
  NI:=I-1; GOTOXY(50,20);WRITE (NI:2);
  GOTOXY(10,22);WRITE(' ');
  B:=1;
  END;
  END;
  3: BEGIN
  FOR I:=1 TO 10 DO
  IF CAR=Z[I] THEN
  BEGIN
  NI:=I-1; write(ni);
  GOTOXY(10,22);WRITE(' ');
  B:=1;
  END;
  END;
  4:BEGIN

```

```

FOR I:=4 TO 10 DO
IF CAR=Z[I] THEN
BEGIN
NI:=I-1; GOTOXY(60,6);WRITE (NI:2);
GOTOXY(10,22);WRITE(' ');
B:=1;
END;
END;
5: BEGIN

if car=z[11] then
BEGIN
car:=readkey; SON:=(-1);
end;
FOR I:=1 TO 10 DO
IF CAR=Z[I] THEN
BEGIN
NI:=(I-1)*(SON); GOTOXY(vv,W);WRITE (NI:2);
GOTOXY(10,22);WRITE(' ');
B:=1;
END;
END;
END;
IF B=0 THEN
COMPONER;
UNTIL B=1;
END;
PROCEDURE AFIRMAR (VAR V: INTEGER);
VAR
YIN:ARRAY [1..2] OF CHAR;
CAR : CHAR;
BEGIN
YIN[1]:=#48; YIN[2]:=#49;
REPEAT
CAR:=READKEY;
IF (CAR<>YIN[1]) AND (CAR<>YIN[2]) THEN
COMPONER;
UNTIL (CAR=YIN[1]) OR (CAR=YIN[2]);
IF CAR=YIN[1] THEN
BEGIN
GOTOXY(30,4);WRITE (' 0'); GOTOXY(10,22);WRITE(' ');
V:=0;
END
ELSE
BEGIN
GOTOXY(30,4);WRITE (' 1'); GOTOXY(10,22);WRITE(' ');
V:=1;
END;
END;
PROCEDURE AFIRMAR1 (VAR V,J: INTEGER);
VAR
YIN:ARRAY [1..2] OF CHAR;
CAR : CHAR;
I,w: INTEGER;
BEGIN
YIN[1]:=#48; YIN[2]:=#49;
REPEAT
CAR:=READKEY;
IF (CAR<>YIN[1]) AND (CAR<>YIN[2]) THEN
COMPONER;
UNTIL (CAR=YIN[1]) OR (CAR=YIN[2]);
IF CAR=YIN[1] THEN
BEGIN
GOTOXY(50,W);WRITE (' 0'); GOTOXY(10,22);WRITE(' ');
V:=0;
END
END

```

ANEXO

---

```
ELSE  
BEGIN  
  GOTOXY(50,W);WRITE(' 1');  
  GOTOXY(10,22);WRITE('  ');  
  V:=1;  
END;  
END;  
END.
```

## BIBLIOGRAFÍA

BAZARAA, Mokhtar S., JARVIS John J.,  
**Programacion Lineal y Flujo en Redes,**  
Ed. Limusa, México, 1981.

BERGE, Claude,  
**Graphs and Hpergraphs,**  
University de Paris,  
Paris, 1970, Vol. 6.

BONDY, J.A. Andmorty,  
**Graphe Theory whit Applications,**  
University de Waterloo,  
Ontario Canadá, 1974.

FORD, L.R. y FULKERSON, D.R.  
**Flows in Networks,**  
Ed . University Press Princenton,  
New Yersey, 1962

GOLDSTEIN, E. Y YUDIN D.B.,  
**Programación Lineal (Problemas y Aplicaciones)**  
Ed. Paraninfo, Madrid, 1977

HESSE, Rick & WOOLSEY Gene,  
**Applied Management Science,**  
Ed. Science Research Asociates, USA, 1980

HILLER, Frederick S. y LIBERMAN, Gerald J.,  
**Introduccion a la Investigación de Operaciones,**  
Ed. Mc Graw Hill, México, 1990, 4ta. edición.

JAUFFRED, Francisco J.M., MORENO Alberto,  
**Métodos de Optimización Lineal -Graficas,**  
Serie Métodos para Análisis de Sistemas de Ingeniería,  
México, 1976, 3 era. ed., Vol. 2, .

JENSEN, Paul A.,  
**Network Flow Programming,**  
Ed. Wiley, Canadá, 1980.

BIBLIOGRAFIA.

---

JOYANES Aguilar Luis,  
**Programacion Turbo Pascal,**  
Ed. Mc. GRAW-HILL,  
México, 1991.

KENNINGTON, Helgason,  
**Algorithms Network Programming,**  
Ed. Wiley, Canadá, 1980.

MINIEKA Edward,  
**Optimitation Algorithms for Networks and Graph,**  
Ed. Dekker, New York,

NEMHAUSER, George L. y WOLSEY, Laurence A.  
**Intenger and Combinatorial Optimization**  
Ed. Wiley Interscience Series in Discrete Mathematics and Optimization  
New York, 1988.

ROCKAFELLAR,  
**Network Flows and Monotropic Optimization,**  
Ed. Welley , Canadá, 1984

SIMONNARD M.,  
**Programación Lineal**  
Ed. Paraninfo, Madrid, 1972.

SWAMY M.N.S., THULASIRAM K.,  
**Graphs, Networks and Algorithms,**  
Ed. Wiley, New York, 1981.

**NETWORKS AND INTERNATIONAL JORNAL**  
Ed. Wiley Interscience Vol. 23 y 24 No. 8 , 1994.,

**BGIDEMO.PAS**  
VERSION 5.5