

35  
257



UNIVERSIDAD NACIONAL AUTONOMA  
DE MEXICO

FACULTAD DE CIENCIAS

ANALISIS DE FLUJOS

**T E S I S**

QUE PARA OBTENER EL TITULO DE

**A C T U A R I A**

**P R E S E N T A :**

**GEORGINA LAURA NEYDE GONZALEZ ISUNZA**



DIRECTOR DE TESIS

M. en C. MARIA DEL CARMEN HERNANDEZ AYUSO

1996

DIVISION DE ESTUDIOS PROFESIONALES



**TESIS CON  
FALLA DE ORIGEN**

FACULTAD DE CIENCIAS  
SECCION ESCOLAR

**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.



UNIVERSIDAD NACIONAL  
AVENIDA DE  
MEXICO

M. en C. Virginia Abrín Barule  
Jefe de la División de Estudios Profesionales de la  
Facultad de Ciencias  
Presente

Comunicamos a usted que hemos revisado el trabajo de Tesis: **Análisis de Flujos**

realizado por **Georgina Laura Neyde González Isunza**

con número de cuenta **9052053-0**, pasante de la carrera de **Actuaría**

Dicho trabajo cuenta con nuestro voto aprobatorio.

Atentamente

Director de Tesis  
Propietario M. en C. María del Carmen Hernández Ayuso  
Propietario M. en I.O. Víctor Rafael Pérez Pérez  
Propietario Mat. José Galaviz Casas  
Suplente Mat. Mónica Leñero Padierna  
Suplente M. en C. Beatriz Rodríguez Fernández

*María del Carmen Hernández Ayuso*  
*[Firma]*  
*[Firma]*  
*Beatriz Rodríguez Fernández*

*Claudia Corilla O.*  
Consejo Departamental de Matemáticas

## AGRADECIMIENTOS

Quiero aprovechar esta página para compartir con los lectores un fragmento de un pensamiento que ha sido el pilar de mi vida desde que lo descubri y que espero aún en momentos difíciles les reafirme lo bello de la existencia.

...

*Pedí salud para hacer cosas grandes  
me dio enfermedad para hacer cosas buenas.*

...

*Pedí poder para obtener alabanzas  
me dio debilidad para sentir la necesidad de Dios.*

*Pedí todo para poder disfrutar de la vida  
me concedió vida para poder disfrutar de todo.*

*No recibí nada de lo que pedí  
pero sí todo lo que deseaba.*

*A pesar de mí mismo,  
las peticiones que no hice me fueron concedidas.*

*Yo, entre los hombres  
SOY EL MAS AFORTUNADO*

*Oración de un soldado anónimo.*

Por esto agradezco a El, que me ha hecho la mujer más afortunada y que me ha dado vida hasta hoy para compartir con tanta gente que me ha brindado un gran apoyo para culminar este trabajo y llegar a este día, la alegría que en este momento siento.

*A El,*

*A tanta gente,*

## GRACIAS

## INDICE

<b>INTRODUCCION GENERAL</b>	<b>I</b>
<b>1. CONCEPTOS BASICOS</b>	<b>1-1</b>
1.1 Redes	1-1
1.2 Trayectorias y cortes	1-3
1.3 Flujos, potenciales	
Divergencias, tensiones	1-5
1.4 Circulaciones	
Red aumentada	1-11
1.5 Conexidad	1-12
1.6 Problema de la trayectoria coloreada o pintada	
Algoritmo de enrutamiento	1-13
1.7 Lema y algoritmo de Minty	1-17
1.8 Problema de distribución factible (Factibilidad de flujo)	
Algoritmo de distribución factible	1-20
1.9 Problema de diferencial factible	
Algoritmo de rectificación de tensión	1-26
1.10 Arboles, bosques	
Algoritmo para el reconocimiento de árboles y bosques	1-32
<b>2. FLUJOS</b>	<b>2-1</b>
2.1 Trayectorias elementales, anticirculaciones	
Flujos conformables	2-1
2.2 Teorema de realización conformable	2-6
2.3 Algoritmo de realización: descripción	2-7

<b>2.4</b>	Justificación del algoritmo	
	Diagrama de flujo del algoritmo	
	Ejemplos	2-9
<b>2.5</b>	Caso especial: circulaciones	2-16
<b>2.6</b>	Resultados de la teoría de árboles y bosques	
	Teorema de expansión	2-23
<b>2.7</b>	Representaciones de Tucker del espacio de circulaciones	2-27
<b>2.8</b>	Representación inicial de Tucker	2-29
<b>2.9</b>	Teorema de bases	
	Cómo obtener más fácilmente una representación inicial de Tucker	2-32
<b>2.10</b>	Pivoteo: cambio de una base a otra	2-36
<b>2.11</b>	Flujos extremos	2-43
<b>2.12</b>	Algoritmo para la representación extrema de flujos	2-46
<b>2.13</b>	Justificación del algoritmo	
	Ejemplo	2-48
<b>3. DIFERENCIALES</b>		<b>3-1</b>
<b>3.1</b>	Diferenciales conformables	3-1
<b>3.2</b>	Teorema de realización conformable para diferenciales	3-3
<b>3.3</b>	Algoritmo de realización para diferenciales: descripción	3-4
<b>3.4</b>	Justificación del algoritmo	
	Diagrama de flujo del algoritmo	
	Ejemplos	3-5
<b>3.5</b>	Representaciones de Tucker para diferenciales (Problema dual a las representaciones de Tucker para flujos)	3-12
<b>3.6</b>	Teorema de bases	3-15
<b>3.7</b>	Pivoteo: cambio de una base a otra	3-21
<b>3.8</b>	Diferenciales extremos	3-27
<b>3.9</b>	Algoritmo para la representación extrema de diferenciales	3-30

<b>3.10</b>	<b>Justificación del algoritmo</b>	
	Ejemplo	3-32
<b>3.11</b>	<b>Analogías</b>	3-36
<b>4.</b>	<b>EL PAQUETE COMPUTACIONAL</b>	4-1
<b>4.1</b>	<b>Representación de redes en el paquete computacional</b>	4-1
<b>4.2</b>	<b>Estructura general del programa</b>	
	Procedimientos de relevancia	4-3
<b>4.3</b>	<b>Principales constantes a lo largo del paquete de cómputo</b>	4-14
<b>4.4</b>	<b>Manual del usuario</b>	4-15
<b>4.4.1</b>	<b>Cómo comenzar</b>	
	Un viaje a través de los menús	4-15
<b>4.4.2</b>	<b>Sobre los archivos de datos que representan a las redes</b>	4-17
<b>4.4.3</b>	<b>Resultados</b>	
	Ejemplos	4-21

## **CONCLUSIONES**

### **Apéndice**

Guía de referencia sobre los principales procedimientos y funciones con sus parámetros	A-1
--	-----

### **Bibliografía**

## INTRODUCCION

Los modelos de redes son utilizados desde sus comienzos en numerosas aplicaciones de una manera muy natural; en la actualidad resuelven problemas en diversos campos como: física, administración, química, planeación, Ingeniería, etc. Algunas de estas aplicaciones son usadas para controlar los sistemas de comunicaciones donde podemos representar satélites, computadoras, centrales telefónicas, etc. en los nodos, así mismo los cables y rutas de transmisión podrían asociarse a los arcos de la red de manera que podemos relacionar a los paquetes de datos o a las transmisiones mismas con los flujos que se asocian a los arcos. Otros problemas muy conocidos son los de sistemas de transporte (tanto terrestre como aéreo), tráfico urbano, sistemas de producción-distribución, sistemas hidráulicos, etc. De aquí la importancia del estudio de esta rama de las matemáticas que proporciona alternativas gráficas para el entendimiento de esta gran variedad de sistemas.

Los primeros destellos de Teoría de Gráficas se dieron en 1736 con el ya famoso problema de Euler "Los siete puentes de Königsberg". Entre finales de los años 40's y los años 50's se desarrollaron la mayoría de los algoritmos principales para flujo en redes, entre ellos el de rutas más cortas y el de flujo máximo de Ford y Fulkerson. Paralelamente a la teoría de redes se fue desarrollando la programación lineal, siendo Von Neumann el que introduce el concepto de dualidad en esta rama de las matemáticas en 1947 y es hasta 1951 que Dantzig crea el método simplex que resuelve eficientemente la mayoría de los problemas de programación lineal y que, hasta la fecha, también es muy socorrido para resolver problemas de redes.

En los años 60's y 70's hubo también muchas contribuciones, particularmente en la creación de algoritmos computacionales cada vez más eficientes para los problemas de costo mínimo.

En los 80's fueron importantes también las mejoras en la eficiencia de los algoritmos; sin embargo fue en esta década que Rockafellar (1984) encontró relaciones fuertes entre los problemas primales y duales, utilizando resultados de teoría de gráficas, mismas que empleó para mejorar los algoritmos existentes sobre todo en los problemas de factibilidad y optimalidad estableciéndose un puente que acerca a la programación lineal y a la teoría de redes fuertemente permitiendo que los algoritmos de redes fueran más independientes solucionándose los problemas por completo con sus herramientas e ideologías, cosa que permitirá una mayor retroalimentación entre la programación lineal y la teoría de redes.

El presente trabajo de tesis retoma el nuevo curso que Rockafellar le imprimió a la teoría de redes. Se pretende realizar un trabajo constructivo que lleve desde los algoritmos esenciales que resultan de la aplicación del lema de Minty y del teorema de la red coloreada a la resolución de problemas atacados comúnmente mediante la programación lineal, como lo son la búsqueda de bases para el espacio de circulaciones y diferenciales, así como los problemas de distribución y diferencial factibles. Además se tiene el objetivo de desarrollar un paquete de cómputo que realice los algoritmos referentes a flujos que el trabajo expone, con dos fines: el primero, proporcionar material didáctico a la gente que comienza a familiarizarse con estos problemas y el segundo que tenga una estructura que pueda perfeccionarse posteriormente para poder ser usado en aplicaciones más extensas, en lugar de que éstos tengan que ser programados desde cero por haber sido pensados para un fin demasiado reducido.

El primer capítulo da un acercamiento a la teoría de redes, aplicando desde sus comienzos el enfoque de Rockafellar con el lema de Minty y el teorema de la red coloreada. El algoritmo de enrutamiento y el algoritmo de Minty también son vistos en este capítulo pues serán pilar para muchas demostraciones y algoritmos posteriores. Los problemas de distribución factible y de diferencial factible se plantean aquí para ser desarrollados en capítulos posteriores.

El segundo capítulo ataca la parte central de la tesis correspondiente a flujos. Destacan el teorema de realización conformable, las representaciones de Tucker basadas en bosques maximales que son bases para el espacio de circulaciones y el teorema de representación extrema de flujos.

En el tercer capítulo se desarrolla la teoría de diferenciales en redes de manera completamente similar al segundo capítulo. Así se tienen conceptos, resultados teóricos y problemas definidos sobre redes cuyos algoritmos se resuelven de manera paralela a la del segundo capítulo. Esta similitud entre los problemas de flujos y diferenciales es la que permite la generalización del tipo de algoritmos tratados en este trabajo, a problemas de optimización definidos no necesariamente sobre una red (por ejemplo de programación lineal).

Finalmente el cuarto capítulo está dedicado al diseño del paquete de cómputo y al manual del usuario. En seguida de este capítulo se encuentra un apéndice que contiene la sintaxis y una descripción somera de los principales procedimientos y funciones que se usan a lo largo del código del paquete.

[El ejemplar incluye un disco de 3 1/2 pulgadas con el programa ejecutable y su respectivo código].

# 1

## PRIMER CAPITULO

### CONCEPTOS BASICOS

Este primer capítulo tiene como objetivo proporcionar los conceptos básicos de gráficas y los resultados de vital importancia a partir de los cuales se desarrolla en los siguientes dos capítulos el tema central del trabajo de tesis. La teoría fundamental se basa en resultados de teoría de gráficas retomados por R. T. Rockafellar en su libro "Network flows and monotropic optimization" (New York, 1984, J.Wiley) con un enfoque novedoso y de gran utilidad. Así mismo se definen algunos problemas de flujos y diferenciales que por no ser parte esencial de este trabajo solamente se esbozan. Especialmente el de diferenciales fue incluido porque surge como subproblema del problema de representación de diferenciales. La parte medular del capítulo la forman el Teorema de la Red Coloreada y el Lema de Minty, que se aplican en todos los algoritmos presentados, puesto que permiten detectar trayectorias o cortes con características especiales.

Primeramente se darán una serie de conceptos básicos, qué es una gráfica, una trayectoria, un circuito, etc.

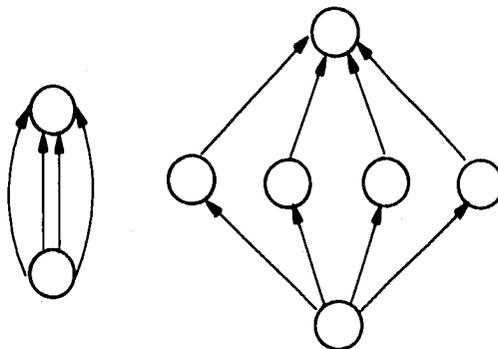
#### SECCION 1.1

#### REDES

Una red es una pareja de conjuntos  $A$  y  $N$  y una función  $f$  que asocia a cada elemento  $j \in A$  un par  $(i, i') \in N \times N, i \neq i'$ . Los elementos de  $N$  reciben el nombre de nodos o vértices y los de  $A$  se llaman arcos y se representan como  $j \sim (i, i')$ , en este caso se dice que  $i$  es el extremo

inicial e  $i'$  es el extremo final del arco  $j$ . Gráficamente los arcos se representan mediante flechas. Si existe más de un elemento con la misma representación, es decir, diferentes arcos que tienen el mismo nodo inicial y el mismo nodo final, se dice que la red tiene **multiplicidades**.

A una red en la cual cada arco corresponde de manera única a una pareja de nodos se le llama **red dirigida** o **digráfica** y el conjunto de arcos puede ser representado mediante un subconjunto de  $N \times N$ . Este tipo de gráficas no acepta multiplicidades; de hecho este tipo de arcos no se necesitan estrictamente pues se puede modificar la gráfica de manera que sigan teniendo la misma función pero sin aparecer como arcos paralelos, un ejemplo de esto se da en la figura :



Mediante la asignación de etiquetas a los arcos podremos trabajar con multiplicidades por lo que en este trabajo no se excluirá a las mismas y en caso de existir un único arco de la forma  $(i, i')$  se permitirá la notación  $j=(i, i')$

La función de incidencia será elemental pues constituye una caracterización algebraica de la red, a partir de la cual muchos resultados son más fáciles de visualizar.

Dada una red  $G=(N, A)$ , donde  $N$  es el conjunto de nodos y  $A$  el conjunto de arcos, la **función de incidencia** se define como

$$e(i, j) = e_{ij} = \begin{cases} 1 & \text{si } i \text{ es nodo inicial del arco } j \\ -1 & \text{si } i \text{ es nodo final del arco } j \\ 0 & \text{en otro caso} \end{cases}$$

y obtenemos así la matriz  $E$  de incidencia, de dimensión  $m \times n$ , (con  $m=|M|$  y  $n=|A|$ ) cuyo  $ij$ -ésimo elemento es  $e_{ij}$ . Tener una representación algebraica de la red ofrece muchas ventajas, como realizar los algoritmos que se asocian a ellas en computadora, proporcionar alternativas para la búsqueda de soluciones, hacer comparaciones con problemas planteados en otras ramas como lo es la programación lineal, etc. Tiene también algunas desventajas como el que en cada columna siempre existirán un 1, un -1 y los demás elementos de la columna serán ceros. En general se puede decir que es una herramienta bastante útil y que permite más versatilidad.

## SECCION 1.2

### TRAYECTORIAS Y CORTES

Un concepto muy intuitivo en redes es el de trayectoria, una trayectoria  $P$  en una red  $G$  es una secuencia finita alternada de nodos y arcos  $i_0, j_1, i_1, j_2, i_2, \dots, j_r, i_r$ , ( $r > 0$ ), donde  $i_k$  es un nodo,  $j_k$  es un arco y el arco  $j_k$  puede tener cualquiera de las siguientes representaciones:  $j_k \sim (i_{k-1}, i_k)$  o  $j_k \sim (i_k, i_{k-1})$ . En digráficas las trayectorias pueden representarse mediante una secuencia de nodos y flechas que indiquen el sentido del arco entre dos nodos adyacentes; no es posible usar esta secuencia en otro caso pues podemos tener mismas representaciones para distintas trayectorias al tener más de un arco entre dos nodos en mismo sentido. La trayectoria en digráficas sería de la siguiente forma:

$$i_0 \rightarrow i_1 \leftarrow i_2 \rightarrow i_3 \dots i_{r-1} \rightarrow i_r \quad r > 0, i_k \in N \quad \text{para todo } k = 1, \dots, r$$

Donde  $i_0$  es el nodo inicial e  $i_r$  es el nodo final de la trayectoria  $P$ . Se dice que esta trayectoria va de  $i_0$  a  $i_r$  y se denota como  $P: i_0 \rightarrow i_r$ . Ahora, si no se tiene una digráfica, puede tenerse una secuencia de arcos representados por etiquetas, las cuales fueron definidas previamente.

Si el arco es de la forma  $j_k \sim (i_{k-1}, i_k)$  se dice que  $j_k$  se recorre positivamente y si  $j_k \sim (i_k, i_{k-1})$  que se recorre negativamente, así podemos formar el conjunto de los arcos recorridos positivamente denotado como  $P^+$  y el conjunto  $P^-$  de los arcos recorridos negativamente.

Una **trayectoria elemental o simple** es aquella en la que cada arco y cada nodo de la secuencia son recorridos sólo una vez. Estas son las trayectorias más frecuentemente usadas, pues tenemos la particularidad de que toda trayectoria puede ser descompuesta en trayectorias elementales. Esto se aplicará en el capítulo 2 cuando veamos el Teorema de Realización Conformable.

Un **circuito elemental** es una trayectoria elemental tal que el nodo inicial y el nodo final de la trayectoria son el mismo.

Para cualquier trayectoria o circuito  $P$  podemos definir su **función de incidencia**, muy similar a la que dimos anteriormente. Esta función describe el sentido en el que el arco  $j$  es usado en la trayectoria.

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

Un **corte o cortadura** es un conjunto  $Q = [S, N \setminus S]$  formado por dos conjuntos de arcos  $Q^- = [S, N \setminus S]^-$ ,  $Q^+ = [S, N \setminus S]^+$ , en donde  $Q^-$  contiene a los arcos de la forma  $j = (i, i')$  tal que  $i \in N \setminus S$  (el complemento de  $S$  en  $N$ ) e  $i' \in S$  y  $Q^+$  contiene a los arcos de la forma  $j = (i, i')$  donde  $i \in S$  e  $i' \in N \setminus S$ .

Este es un concepto dual al de trayectoria; una propiedad útil que relaciona estos dos conceptos es el siguiente: dados una trayectoria  $P: i_0 \rightarrow i_r$  y un corte  $Q = [S, N \setminus S]$ , en el cual  $i_0 \in S$  e  $i_r \in N \setminus S$ ,  $P$  y  $Q$  se intersectan por lo menos en un arco.

La función de incidencia de un corte es la siguiente :

$$e_Q(j) = \begin{cases} 1 & \text{si } j \in Q^+ \\ -1 & \text{si } j \in Q^- \\ 0 & \text{si } j \notin Q \end{cases}$$

De nuevo esta función describe el sentido del arco  $j$  en el corte.

## SECCION 1.3

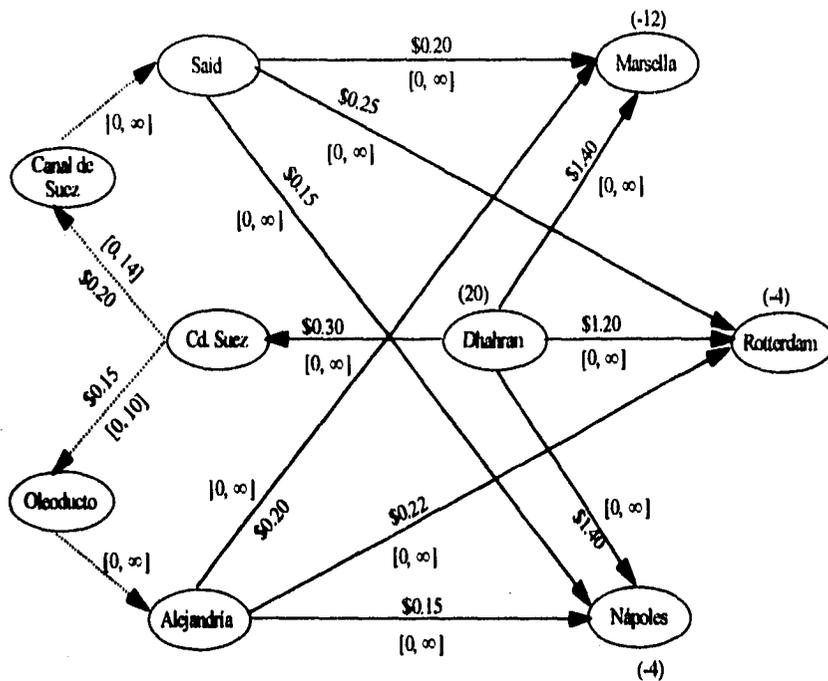
### FLUJOS, POTENCIALES DIVERGENCIAS, TENSIONES

El concepto de red proporciona una representación gráfica para una cantidad enorme de problemas, además es posible asociar información a los nodos o a los arcos y entonces definir problemas de optimización sobre ellas convirtiéndose así en una de las herramientas más útiles. Consideremos el siguiente ejemplo.

Un exportador de aceite quiere encontrar la forma más económica de abastecer a sus clientes. El exportador se encuentra en la ciudad de Dhahran en Arabia Saudita y sus clientes en los puertos de Rotterdam, Marsella y Nápoles en Europa, con una demanda de 4, 12 y 4 millones de barriles respectivamente. El exportador tiene que transportar el 30% de su producto en buques tanque de gran calado y el resto (almacenado en barriles) se transportará en otras embarcaciones. Existen tres alternativas para llegar a cada uno de los puertos, éstas se enlistan a continuación con sus respectivos costos de transporte por barril de aceite:

1. Rodear Africa y navegar hasta Rotterdam, Marsella y Nápoles con un costo de \$1.2, \$1.40 y \$1.40 respectivamente.
2. Navegar de Dhahran a Ciudad Suez y posteriormente a puerto Said pasando por el canal de Suez y en seguida tomar rumbo hacia Rotterdam, Marsella y Nápoles. El costo de transporte de Dhahran a Ciudad Suez es de \$0.30, además existe un costo adicional por barril de \$0.20 por recorrer el canal de Suez a través del cual no pueden atravesar buques tanque de gran calado. Finalmente los costos de transporte de puerto Said a Rotterdam, Marsella y Nápoles son de \$0.25, \$0.20 y \$0.15 respectivamente.
3. Navegar hasta ciudad Suez y mandar el producto mediante el oleoducto que llega hasta Alejandría y de nuevo embarcarlo para llegar a Rotterdam, Marsella y Nápoles. El costo por mandar el aceite a través del oleoducto es de \$0.15 por barril; el sistema tiene una capacidad de 10 millones de barriles. El costo de transporte a partir de Alejandría es de \$0.22, \$0.20 y \$0.15 para llegar a Rotterdam, Marsella y Nápoles respectivamente.

La siguiente red representa todo el problema:



Otro ejemplo es el siguiente:

Un ingeniero desea un calendario de actividades para saber cuanto tiempo le llevará construir la casa y las fechas de las actividades cuyo retraso afectaría al tiempo de duración del proyecto. La preparación de la construcción de una casa puede descomponerse en los trabajos siguientes

No.	Actividad	Duración (en días)
1.	Ejecución de las excavaciones	10
2.	Colocación de la grúa	2
3.	Cimientos	5
4.	Ramificación eléctrica	3
5.	Instalación de la fosa séptica	6

Sin embargo existen restricciones de precedencia para realizar estas actividades y son las siguientes:

- La grúa no puede funcionar si la ramificación eléctrica no está lista.
- Para la colocación de cimientos se necesita la grúa.
- La fosa séptica y los cimientos no pueden efectuarse si las excavaciones no están listas.

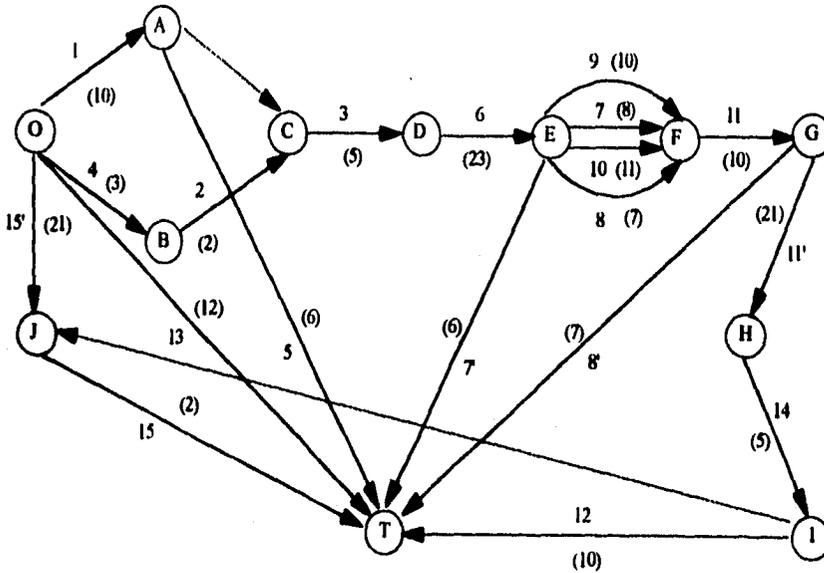
La construcción de la casa comprende además:

No.	Actividad	Duración (en días)
6.	Albañilería	23
7.	Cubierta de techo que se divide en armadura y cubierta de techo	8 y 6 respec.
8.	Plomería; calefacción	2
9.	Electricidad	10
10.	Trabajos pequeños	11
11.	Yeso	10
12.	Pintura	10
13.	Tuberías para el agua	12
14.	Mosaicos	5
15.	Colocación de sanitarios	2

Para estas actividades las restricciones de precedencia son:

- La albañilería no puede comenzarse sin que la cimentación haya sido terminada.
- La albañilería debe estar lista para poder empezar las tareas 7, 8, 9 y 10.
- Es necesario que la plomería, la armadura, los trabajos pequeños y la electricidad sean terminados para empezar el enyesado (que tarda en secarse 21 días).
- La calefacción puede empezarse una vez que se ha terminado el enyesado aún cuando éste no haya terminado de secar.
- El mosaico se coloca cuando el yeso está seco.
- Los sanitarios pueden colocarse cuando los mosaicos estén colocados (tardan 21 días en mandar los sanitarios).
- Los trabajos de pintura empiezan cuando el yeso está seco y los trabajos pequeños, electricidad y colocación de mosaicos terminados.

La red que representa el problema es la siguiente:



En donde:

Etiqueta	Actividad	Duración (en días)
7	Armadura	8
7'	Cubierta de techo	6
8	Plomería	7
8'	Calefacción	7
11'	Secado del yeso	21
15'	Tiempo de entrega de los sanitarios	21

Las etiquetas para los nodos (A a la J, O, T) representan las fechas de comienzo de las actividades, en donde O significa la fecha de inicio del proyecto y T la fecha de término.

Nótese que en ambos ejemplos se ha incluido toda la información del problema en la red al asociar números o rangos a los arcos y nodos.

Así pues, un flujo en una red  $G$  es una función que le asocia a cada arco de  $A$  un número real: un flujo puede representarse como un vector que pertenece a  $R^{|A|}$  y a cada entrada se le llama el flujo a través del arco  $j$

Podemos decir que el flujo que "sale" del nodo  $i$  es la suma de todos los flujos correspondientes a los arcos  $j$  cuyo nodo inicial es  $i$  (como la cantidad máxima de agua que puede salir de una toma a través de sus diferentes salidas.), es decir aquellos arcos que tienen asociado un 1 en la matriz de incidencia ( $e(i, j) = 1$ ); por otro lado el flujo que "entra" es la suma del flujo a través de todos los arcos cuyo nodo terminal es  $i$  y por tanto  $e(i, j) = -1$ . Finalmente la diferencia entre el flujo total que sale del nodo  $i$  y el flujo total que entra al nodo  $i$  se le llama divergencia del nodo  $i$ ; se denota  $Y(i)$  y es igual a :

$$Y(i) = \sum_{j \in A} e(i, j)X(j) = \text{Divergencia del flujo } X \text{ en el nodo } i$$

Se dice que un nodo  $i$  es un origen o fuente si su divergencia es mayor que cero y que es un destino o sumidero si la divergencia asociada a él es menor que cero.

Cuando la divergencia es igual a cero en un cierto nodo se dice que en ese nodo el flujo se conserva.

Una resultado importante es el principio de divergencia total que establece que en una red  $G$  se cumple:

$$\sum_{i \in N} Y(i) = 0$$

Matricialmente puede expresarse  $Y = EX$ . El vector divergencia con respecto a un flujo se denota  $divX$ ; notemos que el vector  $Y \in R^{|N|}$ , cuya  $i$ -ésima componente es  $Y(i)$ , es precisamente  $divX$

La divergencia de  $X$  desde  $S$  se define como la suma de las divergencias de los nodos pertenecientes al conjunto  $S$ , es decir :

$$Y(S) = \sum_{i \in S} Y(i)$$

La función de incidencia de una trayectoria puede ser vista como un flujo a través de ella; observemos que este flujo define divergencia igual a cero en cualquier nodo diferente del nodo inicial y final de la trayectoria, uno para el nodo inicial y menos uno para el final. En este caso el nodo inicial es fuente y el nodo final destino.

Un resultado que también será de utilidad es el que relaciona los vectores :

### Principio fundamental de divergencia

Sea  $X$  un flujo en la red  $G$  y sea el corte  $Q=[S, N/S]$ . Entonces :

$$Y(S) = e_Q X, \quad \text{donde } Y = \text{div} X,$$

Notemos que si  $S=N$  entonces  $Q=\phi$  y tenemos el caso particular del principio de divergencia total. A  $e_Q X$  se le llama flujo a través de un corte  $Q$ .

Así como tenemos que el flujo es el resultado de una función que asocia reales a arcos, tenemos una función análoga que asocia reales a nodos; a este número se le llama potencial  $U(i)$ ; éste es un concepto dual al de flujo y así como la divergencia es una función definida a partir del flujo, tenemos la tensión (también llamada diferencial) definida a partir del potencial.

La tensión  $V(j)$  en el arco  $j$  es la diferencia del potencial del nodo final menos el potencial del nodo inicial.

$$V(j) = U(i') - U(i) = \text{Tensión a través de } j \sim (i, i')$$

Que puede también calcularse como

$$V(j) = - \sum_{i \in N} U(i) e(i, j)$$

o matricialmente  $\Delta U = V = -UE$  donde  $\Delta U$  se lee diferencial del potencial  $U$ . Y tenemos que  $Y=EX$  y  $V=-UE$  es una pareja de sistemas duales.

El conjunto de todos los diferenciales se preserva bajo las operaciones de suma y multiplicación por un escalar, así este conjunto es un subespacio de  $R^{M1}$  llamado espacio de diferenciales.

Ahora podemos notar que la función de incidencia del corte  $Q$  puede interpretarse como un diferencial que corresponde al siguiente potencial

$$e_S(j) = \begin{cases} 1, & i \in S \\ 0, & i \notin S \end{cases}$$

es decir  $e_Q = -\Delta e_S = \Delta e_{N \setminus S}$ . De aquí que el corte corresponda al concepto dual de trayectoria.

#### SECCION 1.4

#### CIRCULACIONES RED AUMENTADA

Una circulación es un flujo  $X$  para el cual la divergencia en cada uno de los nodos es igual a cero.

Las circulaciones se preservan bajo las operaciones de suma y multiplicación por un escalar por lo que forman un subespacio lineal de  $R^{M1}$  llamado espacio de circulaciones. Este espacio es complementario ortogonal al formado por los diferenciales.

Una de las razones por la que las circulaciones son tan importantes es que la teoría relacionada con ellas se simplifica en mucho y que podemos definir a partir de cualquier flujo sobre una red  $G$  una circulación sobre una red  $\bar{G}$  llamada aumentada que a continuación explicaremos. Se trabaja sobre esta red y una vez que se obtuvo el resultado se transforma de nuevo a la red original, haciendo mucho más sencillos los cálculos.

## Red aumentada

Sea  $G = (N, A)$  una red cualquiera y  $X$  el flujo asociada a ella.

### Paso 1

Agregar un nodo  $\bar{i}$  (al que llamaremos nodo de distribución) a  $G$ . Sea  $\bar{N} = N \cup \bar{i}$ .

### Paso 2

Crear un arco  $j_i \sim (\bar{i}, i)$  para cada  $i \in N$  que se llamará arco de distribución. Sea  $\bar{A} = A \cup \{j_i \mid i \in N\}$

$\bar{G} = (\bar{N}, \bar{A})$  es la red aumentada de  $G$

### Paso 3

Se denotará al flujo de  $\bar{G}$  como  $\bar{X}$  definido a partir de  $X$  de la siguiente manera:

$$\bar{X}(j) = X(j) \quad \text{para todo } j \in A$$

$$\bar{X}(j_i) = Y(i) \quad \text{para cada } j_i \text{ arco de distribución}$$

$\bar{X}$  es una circulación.

## SECCION 1.5

### CONEXIDAD

Un concepto realmente útil es el de conexidad; se dice que una red  $G$  es **conexa** si para todo par de nodos diferentes  $s$  y  $s'$ , existe una trayectoria  $P : s \rightarrow s'$ . Si la red  $G$  no es conexa puede particionarse en  $k$  componentes conexas, esto es en  $k$  redes  $(N_i, A_i)$ ,  $i=1, \dots, k$  en donde los subconjuntos de nodos  $N_i$ ,  $i=1, \dots, k$  forman una partición de  $N$  y son tales que dos nodos  $s, s'$  pertenecen al mismo  $N_i$  si y sólo si, existe una trayectoria  $P : s \rightarrow s'$  o  $s=s'$ . Los subconjuntos  $A_i$  están definidos como los arcos cuyos extremos están en  $N_i$  (de hecho  $A_i$ ,  $i=1, \dots, k$  forman una partición de  $A$ )

## SECCION 1.6

### PROBLEMA DE LA TRAYECTORIA COLOREADA O PINTADA ALGORITMO DE ENRUTAMIENTO

En algunos problemas surge la necesidad de determinar trayectorias o cortes con ciertas restricciones de recorrido. Dependiendo de cuáles sean éstas se definirá una partición, llamada coloración, del conjunto de arcos de una red de la siguiente manera :

- Verde - Para aquéllos arcos que pueden recorrerse tanto en su sentido (positivamente) como en sentido contrario (negativamente).
- Blanco - Para aquéllos que pueden recorrerse únicamente positivamente.
- Negro - Para aquéllos que pueden recorrerse únicamente negativamente.
- Rojo - Para los que no pueden recorrerse en ningún sentido.

Estas coloraciones son de gran utilidad pues con ellas se pueden simular una gran cantidad de situaciones, por ejemplo una red de calles, avenidas y autopistas donde el rojo puede representar las avenidas en reparación por las que no se puede transitar, verde las de doble sentido, blanca las simples y si analizamos el caso del transporte público el negro podría ser para aquel tramo de calles en las que los autobuses transitan contra la circulación.

También son muy útiles las coloraciones para encontrar cierto tipo de trayectorias como son las positivas, las que cumplen ciertas condiciones de flujo, ciertas condiciones de tensión, etc.; en fin, dándonos cuenta de la infinidad de aplicaciones demos paso a la definición formal del problema.

#### Problema de la trayectoria coloreada (o pintada)

Sean dos conjuntos no vacíos de nodos  $N^+$  y  $N^-$  en la red  $G$  y sea una coloración de  $A$  consistente en los colores verde, blanco, negro y rojo. El problema es determinar una trayectoria  $P : N^+ \rightarrow N^-$  (es decir,  $P : I_0 \rightarrow I_r$  con  $I_0 \in N^+$  e  $I_r \in N^-$ ) tal que todo arco de  $P^+$  es

verde o blanco y todo arco de  $P^-$  es verde o negro. Es decir todo arco de  $P$  cumple las restricciones impuestas en su recorrido.

A las trayectorias tales que todo arco recorrido positivamente es verde o blanco y todo arco recorrido negativamente es negro o verde se les llama **trayectorias compatibles con la coloración**.

Podemos obtener un problema análogo al de la trayectoria pintada para cortes, éste es el problema del corte coloreado; dicho problema dice : Sean  $N^+$  y  $N^-$  contenidos en  $N$  tales que  $N^+ \cap N^- = \emptyset$ . Sea una coloración en la red  $G$  con los colores verde, blanco, negro y rojo. El problema es determinar un corte  $Q : N^+ \downarrow N^-$  (un corte  $Q$  que separa a  $N^+$  de  $N^-$ , es decir un corte de la forma  $Q[S, N \setminus S]$ , donde  $N^+$  está contenido en  $S$  y  $N^-$  está contenido en  $N \setminus S$ ) tal que todo arco de  $Q^+$  sea rojo o negro mientras que todo arco de  $Q^-$  sea rojo o blanco. A este corte se le llama **corte compatible con la coloración** y si además separa  $N^+$  de  $N^-$  constituye la solución al problema del corte coloreado.

Los problemas de la trayectoria coloreada y el del corte compatible con la coloración, son problemas duales.

En realidad, con el mismo algoritmo podemos resolver ambos problemas, gracias al teorema de la red coloreada. La demostración es constructiva por lo que proporciona el algoritmo para su solución. A continuación se enuncia tal algoritmo:

#### **Teorema de la red coloreada**

Sean  $N^+$  y  $N^-$  contenidos en  $N$ , tales que  $N^+ \cap N^- = \emptyset$ . Entonces, para toda coloración de la red  $G$  con los colores verde, blanco, negro y rojo, una y sólo una de las siguientes afirmaciones es válida :

1. El problema de la trayectoria coloreada tiene solución  $P$
2. El problema del corte coloreado tiene solución  $Q$

**Algoritmo de enrutamiento (o de coloración)**

Su propósito es resolver el problema de la trayectoria coloreada, de no existir dicha trayectoria proporcionará un corte compatible con la coloración que separa a  $N^+$  de  $N^-$

**Paso 1**

Sea  $S=N^+$  y sea  $\theta$  vacío.

**Paso 2**

Determinese el corte  $Q=\{S, N \setminus S\}$ .

- Si existe  $j \in Q^+$  tal que  $j$  es verde o blanco, o si existe  $j \in Q^-$  tal que  $j$  es verde o negro ir a 3.
- Si no existe tal  $j$  terminar. En este caso no existe solución al problema de la trayectoria compatible y  $Q$  es un corte compatible.

**Paso 3**

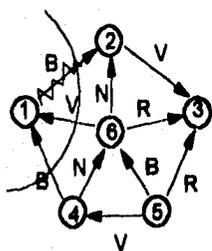
Sea  $\theta(i)=j$ , en donde  $i$  es el extremo de  $j$  que no está en  $S$ .

Sea  $S=S \cup \{i\}$  (El enrutamiento  $\theta$  es compatible con la coloración).

- Si  $i \in N^-$  terminar. Los arcos del enrutamiento  $\theta$  contienen una trayectoria  $P: N^+ \rightarrow i, i \in N^-$  compatible con la coloración.
- Si  $i \in N^+$  se tiene  $S \cap N^- = \phi$ . Ir a 2.

A continuación se da un ejemplo que dejará más claro el procedimiento.

**Iteración 1:**

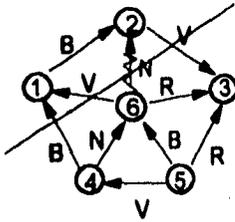


$N^+ = 1$   
 $N^- = 3$

S inicial	$Q^+ (j^-)$	$Q^- (j^-)$
{1}	(1, 2)	(6, 1), (4, 1)

i	$\theta(i)$	S final
2	(1, 2)	{1, 2}

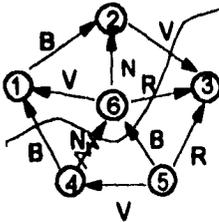
Iteración 2:



$S$ inicial	$Q^+ (j^-)$	$Q^- (j^-)$
{1, 2}	(2, 3)	(4, 1), (6, 1), (6, 2)

$i$	$\alpha(i)$	$S$ final
6	(6, 2)	{1, 2, 6}

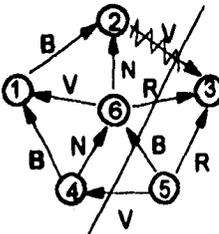
Iteración 3:



$S$ inicial	$Q^+ (j^-)$	$Q^- (j^-)$
{1, 2, 6}	(6, 3), (2, 3)	(4, 1), (4, 6), (5, 6)

$i$	$\alpha(i)$	$S$ final
4	(4, 6)	{1, 2, 6, 4}

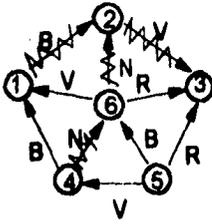
Iteración 4:



$S$ inicial	$Q^+ (j^-)$	$Q^- (j^-)$
{1, 2, 6, 4}	(6, 3), (2, 3)	(5, 4), (5, 6)

$i$	$\alpha(i)$	$S$ final
3	(2, 3)	{1, 2, 6, 4, 3}

{3}  $\in N$ . FIN



$$G(3) = (2, 3), G(2) = (1, 2)$$

$$P \ 1 \rightarrow 2 \rightarrow 3$$

## SECCION 1.7

### LEMA Y ALGORITMO DE MINTY

El Lema de Minty es un resultado de teoría de gráficas que de hecho es equivalente al teorema de la red coloreada, además es también un teorema de alternativas. En algunos de los problemas tratados en este trabajo se utilizará este lema que a continuación se enuncia:

#### Lema de Minty

Dada una coloración en la red  $G$  con los colores verde, blanco, negro o rojo y cualquier arco  $\bar{j}$  blanco o negro, una y sólo una de las siguientes afirmaciones es válida

1. Existe un circuito (elemental)  $P$  compatible con la coloración que usa  $\bar{j}$ .
2. Existe un corte (elemental)  $Q$  compatible con la coloración que usa  $\bar{j}$ .

Así el Algoritmo de Minty tiene como fin identificar si un arco determinado  $\bar{j}$  blanco o negro, pertenece a un circuito o a un corte y queda establecido de la siguiente forma :

## Algoritmo de Minty

### Paso 1

Seleccionar un arco  $\bar{j}=(i, i')$  blanco o negro.

### Paso 2

- Si el arco es blanco  $N^+ = \{i'\}$  y  $N^- = \{i\}$ .
- Si el arco es negro  $N^+ = \{i\}$  y  $N^- = \{i'\}$ .

### Paso 3

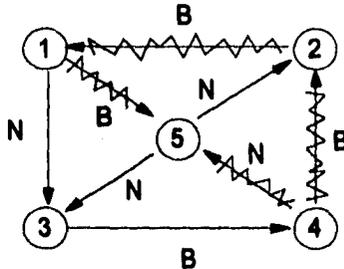
Aplicar el algoritmo de enrutamiento utilizando  $N^+$  y  $N^-$  definidos en paso 2.

### Paso 4

- Si se encontró una trayectoria  $P$  elemental entonces  $C=P \cup \bar{j}$  es un circuito compatible con la coloración que contiene a  $\bar{j}$ .
- Si se encontró un corte  $Q$  elemental que separa a  $N^+$  de  $N^-$  entonces obtuvimos un corte compatible con la coloración que contiene a  $\bar{j}$ .

Tanto el algoritmo de enrutamiento como el de Minty constituyen subrutinas de todos los algoritmos vistos a lo largo del trabajo.

Un ejemplo del algoritmo de Minty se aplica a la siguiente red :



B o N	N <sup>+</sup>	N <sup>-</sup>	TRAYECTORIA ENCONTRADA (Después del algoritmo de enrutamiento)
(2, 1)	1	2	P: 1 → 5 ← 4 → 2

De aquí que el circuito compatible encontrado es : P : 1 → 5 ← 4 → 2 → 1

El algoritmo de coloración y el de Minty son utilizados (como subrutinas) para detectar cierto tipo de trayectorias en la resolución diversos problemas como por ejemplo: el problema de flujo máximo, problema de trayectoria mínima y problemas de factibilidad, entre otros. En estos problemas se definen restricciones sobre arcos y nodos. En particular los problemas de flujo factible y tensión factible son:

**Problema de factibilidad de flujo:**

Sean  $C(j)=[c^-(j), c^+(j)]$  para todo arco  $j$ , con  $c^-(j), c^+(j) \in R$ , intervalos a los que llamaremos de capacidad y sean  $C(i)$ , para todo  $i \in N$ , intervalos a los que llamaremos de oferta. Se desea determinar un flujo  $X$  tal que  $X(j) \in C(j)$  para todo  $j \in A$  y  $Y(i) \in C(i)$  para todo  $i \in N$  donde  $Y = \text{div} X$ . Este problema es equivalente al de distribución factible que difiere del anterior al considerar un valor de terminado  $b(i)$  de oferta en el nodo  $i$  en lugar del intervalo  $C(i)$ .

**Problema de factibilidad de tensión:**

Determinar un potencial  $U$  tal que  $V(j) = (\Delta U)(j) \in D(j)$  para todo  $j \in A$  en una red con intervalos  $D(j)=[d^-(j), d^+(j)]$  a los que llamaremos de generación.

Estos problemas se resuelven de modo paralelo y se dedican las dos siguientes secciones a la presentación de ambos algoritmos.

En este trabajo surge como subrutina el problema de diferencial factible; sin embargo se han incluido los dos problemas de factibilidad definidos para las dos parejas de variables que se tratan en este capítulo  $((X, Y), (U, V))$  con objeto de que se observe el gran paralelismo entre ambos.

## SECCION 1.6

### PROBLEMA DE DISTRIBUCIÓN FACTIBLE (Factibilidad de flujo)

#### ALGORITMO DE DISTRIBUCION FACTIBLE

Algunas veces se asocia un intervalo real no vacío  $C(j)$  a cada arco  $j$ . Este intervalo recibe el nombre de **intervalo de capacidad** del arco  $j$ . Se denotarán los extremos de  $C(j)$  como  $c^-(j)$ , capacidad inferior del arco  $j$  y  $c^+(j)$  capacidad superior del arco  $j$ . Estos intervalos nos señalarán una banda de números en los cuales el flujo es válido para un problema en particular.

A partir de esto se define que un flujo  $X$  es **factible con respecto a las capacidades** de los arcos si  $X(j) \in C(j)$  para todo  $j \in A$ .

Se define la **capacidad superior**  $c^+(Q)$  para el flujo a través del corte  $Q$  como la suma de los términos  $c^-(j)$ , para  $j$  perteneciente a  $Q^+$ , menos la suma de los términos  $c^+(j)$ , para  $j$  perteneciente a  $Q^-$ . Análogamente se define la **capacidad inferior**  $c^-(Q)$  de  $Q$ . Para un corte  $Q$  sin multiplicidades se tiene :

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

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

Las condiciones que garantizan la existencia de solución para el problema de distribución factible se establecen en el siguiente teorema. Sea  $b(S)$  la cantidad total de oferta del subconjunto de vértices  $S$ ; es decir

$$b(S) = \sum_{i \in S} b(i)$$

### Teorema de distribución factible

El problema de distribución factible tiene solución si y sólo si  $b(N) = 0$  y  $b(S) \leq C^+(Q)$  para todo corte  $Q = [S, N/S]$  con  $S \subset N$ .

Naturalmente en este problema, una vez que se tiene un flujo cualquiera factible con respecto a capacidades, comenzaremos por formar grupos de nodos: uno constituido por aquellos nodos que cumplen con tener una divergencia igual a la oferta planteada, otro constituido por aquellos nodos cuya divergencia es menor a su oferta correspondiente y uno más formado por aquellos nodos cuya divergencia es mayor a su oferta correspondiente. Además creamos otros grupos de arcos que corresponderán a las posibilidades de modificación del flujo asociado a los mismos, de aquí la utilidad de los colores definidos en el algoritmo de enrutamiento; estos grupos serán distinguidos cada uno con un color, verde para aquellos arcos tales que su flujo puede ser tanto aumentado como disminuido, blanco para aquellos arcos cuyo flujo únicamente puede ser aumentado, negro para los arcos cuyo flujo sólo puede ser disminuido y rojo será para los arcos en los cuales el flujo no puede ser modificado. Así al modificar el flujo de una trayectoria compatible con la coloración que va de  $N^+$  a  $N^-$  en la cantidad correcta, se modificarán únicamente el nodo inicial y final construyendo un flujo en cada iteración que "rompa menos" las restricciones de oferta.

Para determinar un potencial tal que  $V(j) = (\Delta U)(j) \in D(j)$  para todo  $j \in A$  utilizamos el siguiente algoritmo :

## Algoritmo de distribución factible

Su propósito es encontrar solución al problema de distribución factible (si existe).

### Algoritmo

#### Paso 1

Determinar un flujo factible  $X$  con respecto a las capacidades de los arcos.

#### Paso 2

Sean

$$N^+ = \{i \in N \mid b(i) > Y(i)\}$$

$$N^- = \{i \in N \mid b(i) < Y(i)\}$$

- Si  $N^+ = \emptyset = N^-$  terminar. El flujo  $X$  resuelve el problema.
- En otro caso, colorear los arcos de la red de la siguiente manera.

<b>Verde</b>	si $c^-(j) < X(j) < c^+(j)$
<b>Blanco</b>	si $c^-(j) = X(j) < c^+(j)$
<b>Negro</b>	si $c^-(j) < X(j) = c^+(j)$
<b>Rojo</b>	si $c^-(j) = X(j) = c^+(j)$

### Paso 3

Aplicar el algoritmo de enrutamiento a la red.

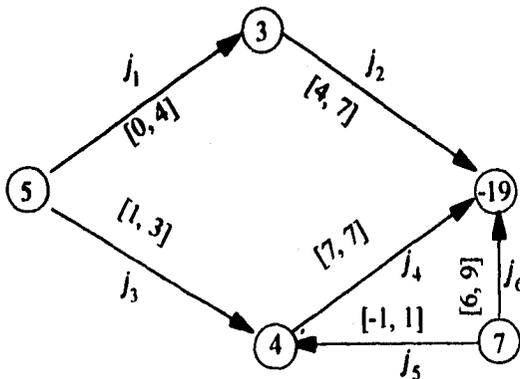
- Si se determina un corte  $Q=[S, N \setminus S]$  compatible con la coloración, terminar. En este caso no existe solución al problema.
- Si se determina una trayectoria  $P: N^+ \rightarrow N^-$  compatible con la coloración, calcular:

$$\alpha = \min \begin{cases} c^+(j) - X(j) & \text{para } j \in P^+ \\ X(j) - c^-(j) & \text{para } j \in P^- \\ b(i) - Y(i), & i \text{ nodo inicial de } P \\ Y(i) - b(i), & i \text{ nodo final de } P. \end{cases}$$

Actualizar  $X = X + \alpha e_p$ , e ir a 2

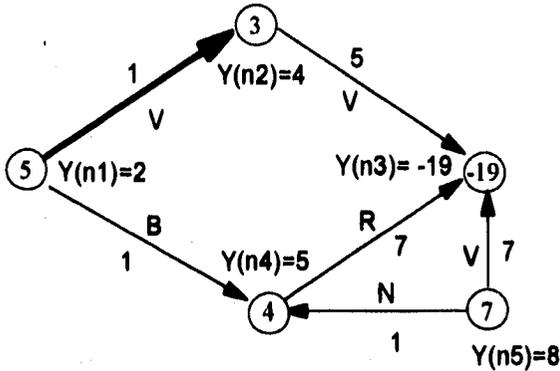
En seguida un ejemplo que ilustra el algoritmo:

Encontrar un flujo tal que la divergencia en cada nodo corresponda a las ofertas asociadas a los mismos.



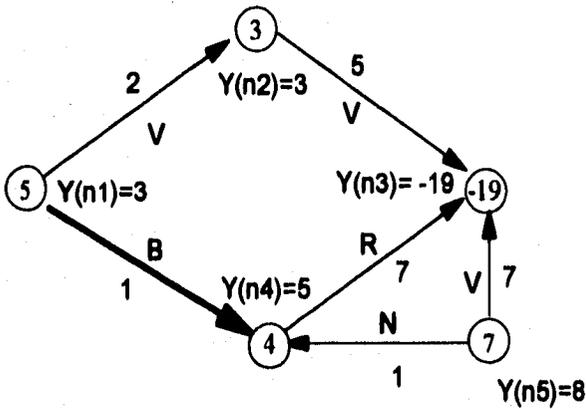
**Iteración 1**

Se inicia con el flujo mostrado en la figura



$N^+ = \{n1\}$   
 $N^- = \{n2, n4, n5\}$   
 $P^+ = \{(n1, n2)\}$   
 $P^- = \emptyset$   
 $\alpha = \min\{3, 3, 1\} = 1$

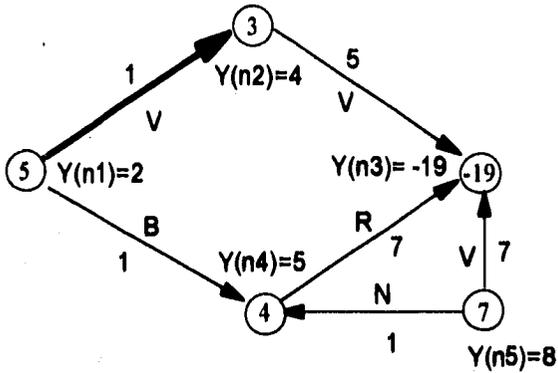
**Iteración 2**



$N^+ = \{n1\}$   
 $N^- = \{n4, n5\}$   
 $P^+ = \{(n1, n4)\}$   
 $P^- = \emptyset$   
 $\alpha = \min\{2, 2, 1\} = 1$

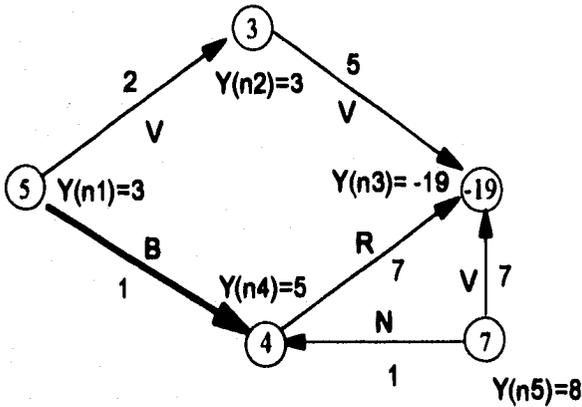
**Iteración 1**

Se inicia con el flujo mostrado en la figura



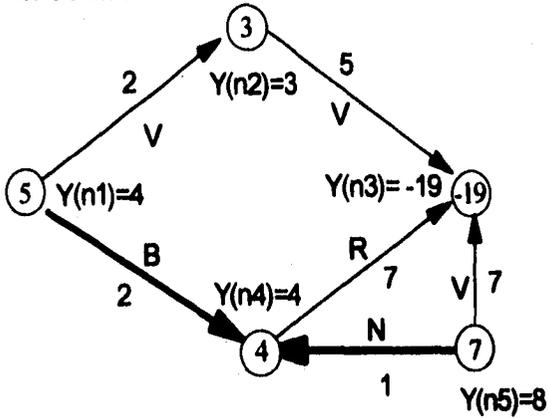
$N^+ = \{n1\}$   
 $N^- = \{n2, n4, n5\}$   
 $P^+ = \{(n1, n2)\}$   
 $P^- = \emptyset$   
 $\alpha = \min\{3, 3, 1\} = 1$

**Iteración 2**



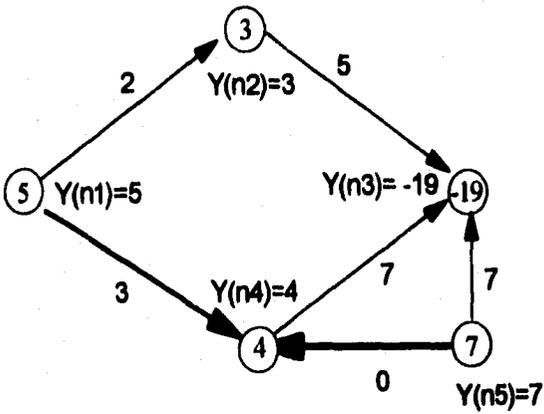
$N^+ = \{n1\}$   
 $N^- = \{n4, n5\}$   
 $P^+ = \{(n1, n4)\}$   
 $P^- = \emptyset$   
 $\alpha = \min\{2, 2, 1\} = 1$

**Iteración 3**



$N^+ = \{n1\}$   
 $N^- = \{n5\}$   
 $P^+ = \{(n1, n4)\}$   
 $P^- = \{(n5, n4)\}$   
 $\alpha = \min\{1, 2, 1, 1\} = 1$

**Iteración 4**



$N^+ = N^- = \emptyset$

FIN

El flujo de la figura es factible con respecto a capacidades y a oferta

## SECCION 1.9

### PROBLEMA DE DIFERENCIAL FACTIBLE ALGORITMO DE RECTIFICACION DE TENSION

Algunas veces se asocia un intervalo real no vacío  $D(j)$  a cada arco  $j$ . Este intervalo recibe el nombre de **intervalo de generación** del arco  $j$ . Se denotarán los extremos de  $D(j)$  con  $d^+(j)$ , límite inferior de generación y  $d^-(j)$ , límite superior de generación respectivamente. Estos intervalos señalarán una banda de números en los cuales la tensión es válida para un problema en particular. Los únicos requerimientos para estos números son :

$$\begin{aligned}d^-(j) &\leq d^+(j) \\d^-(j) &< \infty \\d^+(j) &> -\infty\end{aligned}$$

A partir de esto se define la **tensión factible con respecto a las generaciones** como una tensión  $V$  tal que  $V(j) \in D(j)$ , para todo  $j \in A$ .

Se define la **generación superior**  $d^+(P)$  de una trayectoria  $P$  como la suma de los términos  $d^+(j)$ , para  $j$  recorrido positivamente, menos la suma de los términos  $d^-(j)$ , para  $j$  recorrido negativamente. Análogamente se define la **generación inferior**  $d^-(P)$  de  $P$ . Para una trayectoria  $P$  sin multiplicidades se tiene :

$$\begin{aligned}d^+(P) &= \sum_{j \in P^+} d^+(j) - \sum_{j \in P^-} d^-(j) \\d^-(P) &= \sum_{j \in P^+} d^-(j) - \sum_{j \in P^-} d^+(j)\end{aligned}$$

Posteriormente nos veremos en la necesidad de encontrar un potencial tal que su diferencial se encuentre dentro del intervalo de generación correspondiente y a partir de la generación superior tendremos una forma de saber si existe una solución a dicho problema (Teorema de diferencial factible), este problema queda escrito más formalmente así :

### Problema de diferencial factible

Determinar un potencial  $U$  tal que  $V(j) = (\Delta U)(j) \in D(j)$  para todo  $j \in A$  en una red con intervalos de generación  $D(j) = [d^-(j), d^+(j)]$ .

### Teorema de diferencial factible

El problema de diferencial factible tiene solución si y sólo si  $d^+(P) \geq 0$  para todo circuito  $P$  (elemental).

De nuevo notamos las bondades de las coloraciones pues una vez que se tiene un potencial cualquiera y el diferencial que éste define, intuitivamente comenzaremos por formar grupos de arcos: uno constituido por aquéllos cuyo diferencial está contenido estrictamente en el intervalo de generación, otro por aquéllos cuyo diferencial está abajo de la cota mínima y uno más para aquéllos cuyo diferencial está arriba de la cota máxima; los grupos corresponden a colores, ahora bien para cada grupo se utilizará una estrategia particular, en el caso de los arcos con un diferencial menor a  $d^-$  se necesitará aumentar el potencial del extremo final mientras que para los arcos con diferencial por arriba de  $d^+$  se necesitará aumentar el potencial del extremo inicial del arco. Así utilizando la coloración para cortes compatibles asociamos negro al segundo grupo de ellos, blanco para el tercero, verde para los arcos con diferencial igual a la cota inferior y superior y rojo para los que pueden subir y bajar su diferencial. El conjunto de arcos que define al corte  $Q[S, N/S]$  compatible con la coloración será para el que se modifiquen los diferenciales de acuerdo a sus necesidades (planteadas con la coloración) al alterar el potencial del conjunto  $N/S$ . Nótese que en los arcos que se cumplen las restricciones, la modificación se hará de tal manera que continúe cumpliendo con las restricciones.

Para determinar un potencial tal que  $V(j) = (\Delta U)(j) \in D(j)$  para todo  $j \in A$  utilizamos el siguiente algoritmo:

### Algoritmo de rectificación de tensión

Su propósito es encontrar solución al problema de diferencial factible (si existe).

#### Algoritmo

##### Paso 1

Sea  $U$  cualquier potencial. Sea  $V = \Delta U$ .

##### Paso 2

Sean

$$A^+ = \{j \in A \mid V(j) < d^-(j)\}$$

$$A^- = \{j \in A \mid V(j) > d^+(j)\}$$

- Si  $A^+ = \emptyset = A^-$  terminar. El potencial  $U$  resuelve el problema.
- En otro caso, selecciónese  $\bar{j} \in A^+ \cup A^-$ .

##### Paso 3

Colorear los arcos de  $G$  de la siguiente manera :

rojo	si	$d^-(j) < V(j) < d^+(j)$
negro	si	$V(j) \leq d^-(j), V(j) < d^+(j)$
blanco	si	$V(j) \geq d^+(j), V(j) > d^-(j)$
verde	si	$d^-(j) = V(j) = d^+(j)$

Aplicar el Algoritmo de Minty para  $\bar{j}$

- Si se determina un circuito  $P$  compatible que contenga a  $\bar{j}$  terminar. En este caso  $d^+(P) < 0$  y por tanto no existe solución para el problema.
- Si se determina un corte  $Q$  compatible que contenga a  $\bar{j}$ , calcular :

$$\alpha = \min \begin{cases} d^+(j) - V(j) & \text{para } j \in Q^+ \\ V(j) - d^-(j) & \text{para } j \in Q^- \\ \bar{\alpha} \end{cases}$$

donde :

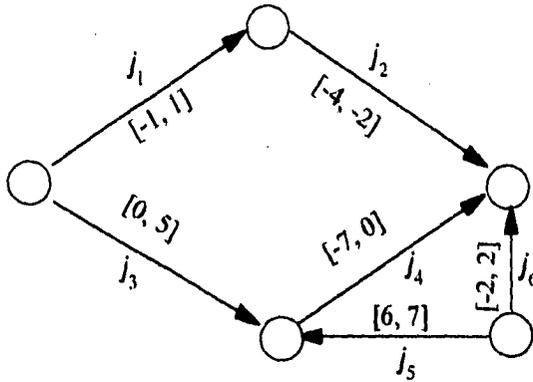
$$\bar{\alpha} = \text{dist}(V(\bar{j}), [d^-(\bar{j}), d^+(\bar{j})]) = \begin{cases} d^-(\bar{j}) - V(\bar{j}) & \text{si } \bar{j} \in A^+ \\ V(\bar{j}) - d^+(\bar{j}) & \text{si } \bar{j} \in A^- \end{cases}$$

Actualizar  $U = U + \alpha e_{NIS}$  e ir a 2, donde

$$e_{NIS} = \begin{cases} 1 & \text{si } i \in N \setminus S \\ 0 & \text{en otro caso} \end{cases}$$

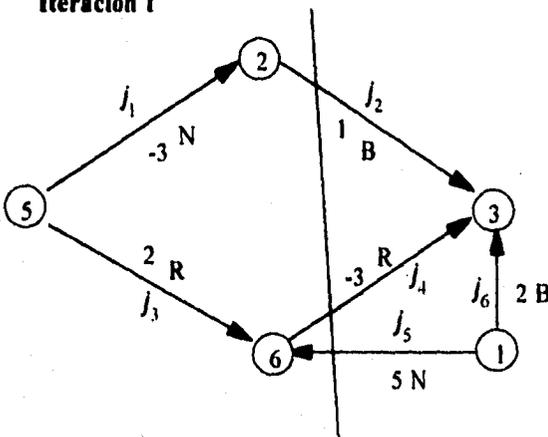
En seguida un ejemplo que ilustra el algoritmo:

Encontrar un potencial tal que el diferencial resultante de él cumpla con las restricciones asociadas a cada arco.



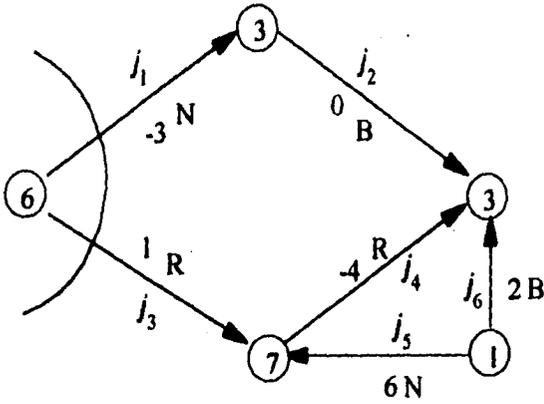
Se inicia la primera iteración con el potencial mostrado en la figura.

Iteración 1



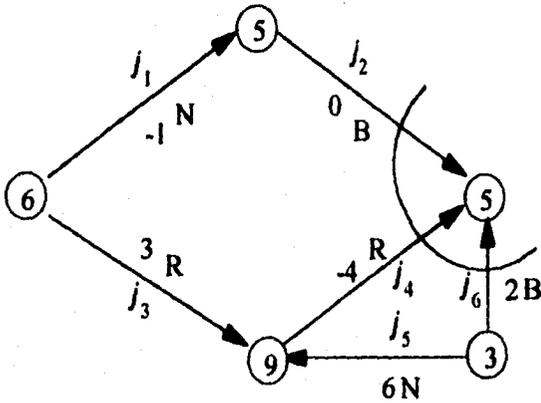
$$\begin{aligned} \bar{j} &= j_5 \\ A^+ &= \{j_1, j_5\} \\ A^- &= \{j_2\} \\ Q^+ &= \{\bar{j}\} \\ Q^- &= \{j_2, j_4\} \\ \frac{\bar{c}}{\alpha} &= 6 - 5 = 1 \\ \alpha &= \min\{1, 4, 5\} = 1 \end{aligned}$$

**Iteración 2**



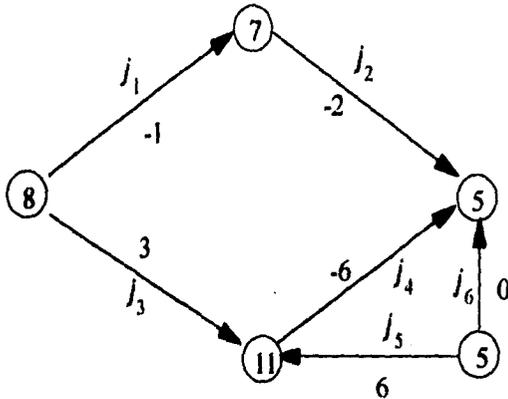
$$\begin{aligned} \bar{j} &= j_1 \\ A^+ &= \{\bar{j}\} \\ A^- &= \{j_2\} \\ Q^+ &= \{j_3, \bar{j}\} \\ Q^- &= \emptyset \\ \bar{\alpha} &= -1(-3) = 2 \\ \alpha &= \min\{4, 4, 2\} = 2 \end{aligned}$$

**Iteración 3**



$$\begin{aligned} \bar{j} &= j_2 \\ A^+ &= \emptyset \\ A^- &= \{\bar{j}\} \\ Q^+ &= \emptyset \\ Q^- &= \{\bar{j}, j_4, j_5\} \\ \bar{\alpha} &= 0(-2) = 2 \\ \alpha &= \min\{2, 3, 4\} = 2 \end{aligned}$$

#### Iteración 4



El potencial encontrado  
resuelve el problema de  
diferencial factible

#### SECCION 1.10

#### ARBOLES, BOSQUES

#### ALGORITMO PARA EL RECONOCIMIENTO DE ARBOLES Y BOSQUES

Lo último que se verá en este capítulo son los árboles y bosques que serán una de las principales herramientas en los capítulos dos y tres.

Un árbol es una gráfica con características especiales y el concepto de bosque es más general, que como su nombre indica, está formado por uno o más árboles.

#### Arboles

Se dice que una red es un árbol si es conexa y no tiene circuitos. En términos más generales se dice que un conjunto de arcos en una red  $G$  forma un árbol si la subred de  $G$  constituida por dichos arcos y los nodos alcanzados por ellos es un árbol.

Así una estructura muy particular es el árbol de expansión de una red  $G$  que es aquel árbol formado por una subred de  $G$  tal que contiene a todos los nodos de  $G$ .

Las siguientes definiciones son equivalentes :

- Un árbol.
- Una gráfica conexa con  $n$  nodos y  $n - 1$  arcos.
- Una gráfica sin circuitos con  $n$  nodos y  $n - 1$  arcos.
- Una gráfica tal que entre cada par de nodos existe una trayectoria única.

Un bosque es un concepto mucho más general y puede ser usado tanto para redes conexas como no conexas

**Bosque** es aquella red en la cual cada componente conexa forma un árbol. En particular, si la red es conexa se tiene un árbol. Un **bosque maximal** es aquel que no está contenido estrictamente en ningún otro bosque; en el caso de red conexa un bosque maximal es un árbol de expansión.

En algunos problemas surge la necesidad de verificar si una red es un árbol o no; para ello utilizaremos el siguiente algoritmo.

### Prueba de árboles

Sea  $G=(N, A)$  una red.

#### Paso 1

Seleccionar un nodo  $s \in N$

#### Paso 2

Colorear todos los arcos de color verde.

Aplicar el algoritmo de coloración con  $N^+ = \{s\}$ ,  $N^- = \emptyset$

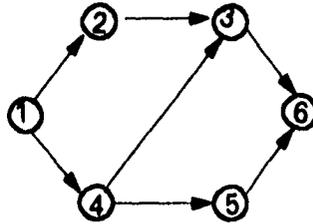
#### Paso 3

- Si  $\theta(i)$  está definido para todo  $i \in N - \{s\}$  y todo arco forma parte del enrutamiento,  $G$  es un árbol.
- En otro caso,  $G$  no es un árbol.

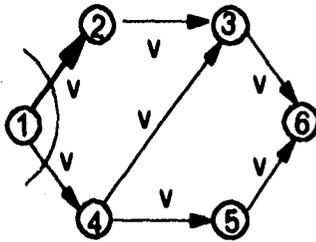
Este algoritmo sería el mismo que se utilizaría para probar si una red es un bosque o no, la única diferencia es que hay que aplicar dicho algoritmo a cada una de las componentes conexas de la red.

En seguida se aplicará el algoritmo a un ejemplo.

Determinar si la siguiente red es un árbol.

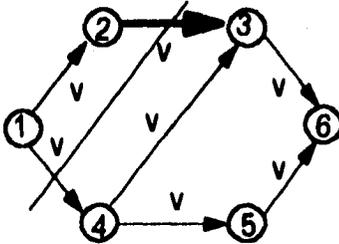


Iteración 1



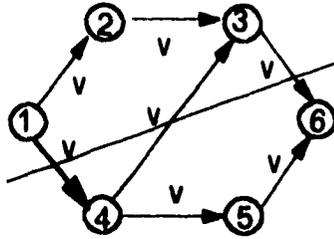
$s = \{1\}$   
 $N^+ = \{1\}, N^- = \emptyset$   
 $S = \{1\}$   
 $\theta(2) = (1, 2)$   
 $S = \{1, 2\}$

Iteración 2



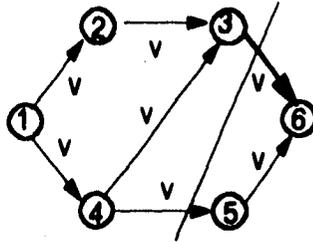
$S = \{1, 2\}$   
 $\theta(3) = (2, 3)$   
 $S = \{1, 2, 3\}$

Iteración 3



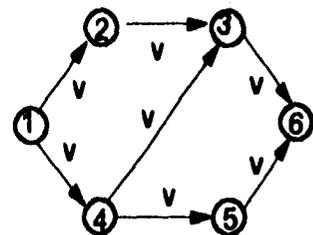
$S = \{1, 2, 3\}$   
 $\theta(4) = (1, 4)$   
 $S = \{1, 2, 3, 4\}$

Iteración 4



$S = \{1, 2, 3, 4\}$   
 $\theta(6) = (3, 6)$   
 $S = \{1, 2, 3, 4, 6\}$

Iteración 5



$S = \{1, 2, 3, 4, 6\}$   
 $\theta(5) = (5, 6)$   
 $S = \{1, 2, 3, 4, 5, 6\}$

En esta iteración  $\theta(i)$  está definido para toda  $i \in N - \{1\}$ ; el enrutamiento  $\{\theta(i) | i \in N - \{1\}\} = \{(1,2), (2,3), (1,4), (3,6), (5,6)\}$  está contenido estrictamente en el conjunto de arcos de la red original. Por tanto esta red no es un árbol.

# 2

## SEGUNDO CAPITULO

### FLUJOS

Este capítulo está dedicado a los flujos conformables, las representaciones de Tucker y los flujos extremos; estos conceptos muestran la cercanía entre el análisis de redes y la programación lineal sin perder las cualidades gráficas del primero.

De la sección 2.1 a la 2.5 se hablará de los flujos conformables y del teorema de realización conformeble, así como el algoritmo de realización que surge a raíz de él. A partir de la sección 2.6 y hasta la sección 2.10 se tratarán a las representaciones de Tucker para el espacio de circulaciones; este tema muestra una analogía sorprendente con el método simplex de programación lineal para resolver el problema primal, aunque es hasta el próximo capítulo en donde se hablará de una dualidad para la resolución del problema. Finalmente se comenzará a tratar a los flujos extremos a partir de la sección 2.11 y hasta finalizar el capítulo. El resultado más importante de flujos extremos es el teorema de representación extrema y el algoritmo que su demostración proporciona. Este teorema es un caso particular o resultado muy conocido en programación lineal llamado teorema de Representaciones.

#### SECCION 2.1

#### TRAYECTORIAS ELEMENTALES, ANTICIRCULACIONES FLUJOS CONFORMABLES

Los flujos conformables, son una manera muy natural de observar el comportamiento de un flujo a través de trayectorias; nos hace comprender más fácilmente como es el flujo en la red, pues nos permiten una descomposición en flujos más "sencillos".

Para realizar esta división en flujos más sencillos, el concepto de flujo elemental será fundamental.

El que un flujo sea elemental depende directamente de las trayectorias que se escojan; su nombre resulta obvio pues queda expresado como un escalar multiplicado por el vector de incidencia de una trayectoria elemental, así un flujo elemental  $X$  es un flujo de la forma :

$$X(j) = \alpha e_p(j) = \begin{cases} \alpha & \text{si } j \in P^+ \\ -\alpha & \text{si } j \in P^- \\ 0 & \text{en otro caso.} \end{cases}$$

donde  $P$  es una trayectoria elemental y  $\alpha$  un escalar mayor que cero.

Para expresar un flujo cualquiera como combinación lineal de flujos elementales, necesitaremos restringir la clase de trayectorias que serán admitidas; por lo que surge el concepto de trayectorias que conforman a un flujo  $X$ .

Se dice que una trayectoria  $P$  conforma a un flujo  $X$  si pasa lo siguiente:

- 1.-  $X(j) > 0$  para todo  $j \in P^+$  y  $X(j) < 0$  para todo  $j \in P^-$ ,  $j \in A$
- 2.- O bien  $P$  es un circuito elemental, o bien el nodo inicial y el final de  $P$  son origen y destino para  $X$  respectivamente.

La condición 1 puede ser expresada en términos de coloraciones.

Sea la coloración:

Blanco :	si $X(j) > 0$
Negro :	si $X(j) < 0$
Rojo :	si $X(j) = 0$

Entonces si  $P$  es una trayectoria compatible con la coloración anterior decimos que  $P$  conforma al flujo  $X$ .

Otra definición que será de suma importancia es la de anticirculación; por la palabra cualquiera pensaría que una anticirculación es aquel flujo que no es circulación pero esto no es cierto en general pues para empezar el flujo cero es tanto circulación como anticirculación, además de que existen flujos que no son ni circulaciones ni anticirculaciones; la definición es la siguiente:

Una anticirculación es un flujo que no contiene circuitos que lo conforman.

Se acaba de decir que el cero es tanto circulación como anticirculación y una pregunta muy natural sería: ¿Existe algún otro flujo que cumpla con eso? y la respuesta es no

#### Proposición

El único flujo que es circulación y anticirculación es el cero.

Sea  $X \neq 0$  una anticirculación. Coloreamos la red con la coloración para flujos conformables. No hay circuitos compatibles con esta coloración ya que  $X$  es una anticirculación. Existe  $\bar{j}$  blanco o negro ya que  $X \neq 0$  entonces por el algoritmo de Minty encontramos un corte compatible con la coloración que contiene a  $\bar{j}$  y

$j$  es negro o rojo  $\forall j \in Q^+$ , es decir  $X(j) \leq 0$  para todo  $j \in Q^+$

$j$  es blanco o rojo  $\forall j \in Q^-$ , es decir  $X(j) \geq 0$  para todo  $j \in Q^-$

Entonces

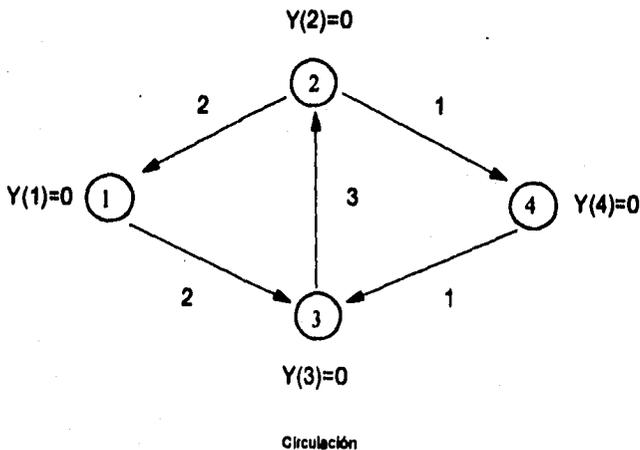
$$e_Q X = \sum_{j \in Q^+} X(j) - \sum_{j \in Q^-} X(j) < 0$$

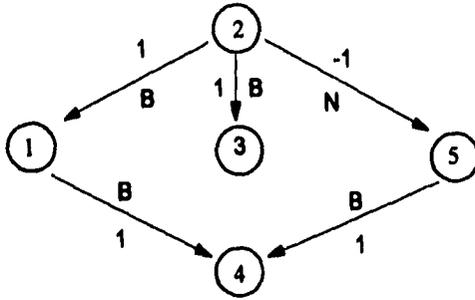
pero sabemos que  $e_Q X = 0$  en toda circulación para todo  $Q = [S, N/S]$  ya que

$$Y(S) = e_Q X = 0$$

$\therefore X$  no puede ser una circulación  $\otimes$

El concepto de anticirculación será fácilmente comprendido al ver los siguientes ejemplos :

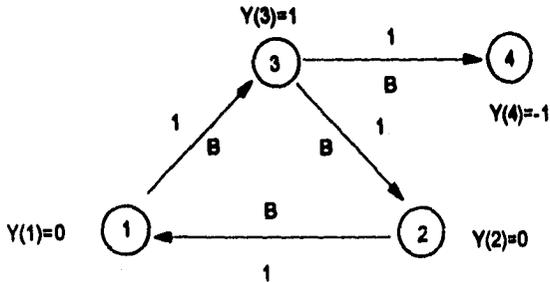




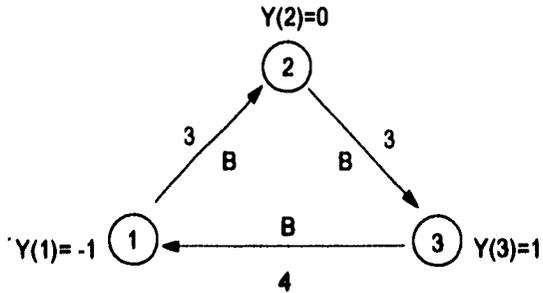
Anticirculación

Como vemos en la última gráfica no hay circuitos compatibles con la coloración, que es lo mismo que decir no existe un circuito que conforme al flujo dado.

El siguiente ejemplo no es ni una circulación ni una anticirculación :  
 La razón de que no sea una anticirculación es que tenemos al menos el siguiente circuito compatible con la coloración :  $1 \rightarrow 3 \rightarrow 2 \rightarrow 1$  y tampoco es una circulación ya que  $Y(3) \neq 0 \neq Y(4)$ .



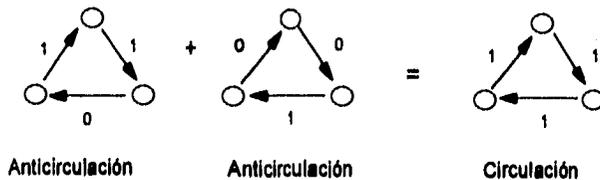
Otro ejemplo muy sencillo de este caso es el siguiente circuito :



Este es un circuito compatible con la coloración y  $Y(1) \neq 0 \neq Y(3)$

$\therefore$  no es ni circulación, ni anticirculación.

Como vimos en la sección 1.4 la suma de circulaciones es una circulación, sin embargo la suma de anticirculaciones no necesariamente es una anticirculación y de esto tenemos un ejemplo muy sencillo.



La importancia de las circulaciones y anticirculaciones radica en que cualquier flujo puede ser descompuesto como una suma de una circulación y una anticirculación tal como se enunciará en el teorema de realización conforme en la próxima sección.

## SECCION 2.2

### TEOREMA DE REALIZACION CONFORMABLE

En esta sección presentaremos el teorema de realización conformable en el cual se establece que un flujo cualquiera puede expresarse como combinación lineal de los vectores de incidencia de flujos elementales y como la suma de una circulación y una anticirculación. Abusando del lenguaje, en adelante nos referiremos a combinaciones lineales de los vectores de incidencia de flujos elementales como "combinación lineal de flujos elementales" para abreviar. La demostración de este teorema es constructiva por lo cual nos proporciona un algoritmo para realizar tal descomposición.

Este teorema y el algoritmo que de él resulta son una herramienta de gran utilidad especialmente para fines teóricos.

#### Teorema de Realización Conformable

Sea  $X \neq 0$  un flujo cualquiera, entonces existen trayectorias elementales  $P_1, \dots, P_r$  tales que conforman a  $X$  y  $\alpha_1, \dots, \alpha_r$  mayores que cero, los cuales pueden ser escogidos enteros si  $X$  es entero, tales que  $X = \alpha_1 e_{P_1} + \dots + \alpha_r e_{P_r}$ .

Además, las trayectorias (no necesariamente únicas) pueden ser escogidas de tal manera que si las que son circuitos son numeradas  $P_1, \dots, P_q$  entonces  $\alpha_1 e_{P_1} + \dots + \alpha_q e_{P_q}$  es una circulación y  $\alpha_{q+1} e_{P_{q+1}} + \dots + \alpha_r e_{P_r}$  es una anticirculación;  $X$  queda entonces representado como la suma de una circulación y una anticirculación.

Recordando que la suma de anticirculaciones no es cerrada veremos que esta expresión particular del flujo  $X$ , representado como la suma de una circulación y una anticirculación, se deberá a que primeramente serán localizados todos y cada uno de los circuitos conformables, por lo que, si la suma de cualquier número de anticirculaciones para cualesquiera de ellas fuera una circulación, eso significaría que algún circuito no fue incluido en un principio. La suma para el caso de circulaciones es cerrada por lo que es claro que  $\alpha_1 e_{P_1} + \dots + \alpha_q e_{P_q}$  es una circulación. El punto quedará más claro al ver el algoritmo de realización.

## SECCION 2.3

### ALGORITMO DE REALIZACION : DESCRIPCION

El propósito es encontrar la descomposición del flujo dado  $X$  como suma de una circulación y una anticirculación expresadas como "combinación lineal de trayectorias elementales".

El algoritmo se divide en dos fases, en la primera fase se encuentran las circulaciones detectando circuitos mediante el algoritmo de Minty con la coloración mencionada al inicio del capítulo; una vez encontrado el circuito se encuentra el mínimo de los valores absolutos de los flujos a través de los arcos del mismo para que obtengamos la circulación tal que los valores absolutos del flujo de cada arco perteneciente a tal circuito sean iguales y los valores del flujo a través de los arcos fuera del circuito sean cero. El mínimo encontrado se resta al circuito y vuelve a comenzar la búsqueda; si ya no hay circuitos pasamos a la segunda fase, en la cual se encuentran las anticirculaciones. Esta fase es muy sencilla pues dado que ya no hay circuitos compatibles con la coloración es suficiente con encontrar una trayectoria compatible.

El algoritmo detallado se presenta a continuación.

#### Algoritmo

##### FASE I

###### Paso 0

$k = 0, t = 0, M = \{ \}$ , donde  $M$  es un conjunto en el que se almacenan aquellos arcos que se sabe no pertenecen a ningún circuito. Sea  $X_0 = X$ .

###### Paso 1

Dar a la red la coloración para flujos conformables

Blanco : si  $X_t(j) > 0$

Negro : si  $X_t(j) < 0$

Rojo : si  $X_t(j) = 0$

Colorear de rojo todo  $j \in M$  con el fin de no repetir la búsqueda.

Ir al paso 2.

###### Paso 2

Buscar un arco  $\bar{j}$  blanco o negro.

-Si existe tal  $\bar{j}$  ir al paso 3.

-Si no existe dicho  $\bar{j}$  pasar a la fase 2.

### Paso 3

Aplicar el algoritmo de Minty para encontrar un circuito que contenga a  $\bar{j}$

-Si existe tal circuito, sea  $P_k$  éste, calcular el mínimo de:

$$a_k = \min |X(j)| \quad \forall j \in P_k$$

y actualizar el flujo de la siguiente manera :

$$X_{k+1} = X_k - a_k e_{P_k}; k = k + 1.$$

Ir al paso 1.

-Si no existe el circuito que contenga a  $\bar{j}$  entonces se encontró un corte.

Añadir los arcos del corte a  $M$ , pues estos arcos ya no pertenecen a ningún otro circuito, en otras palabras, hacer  $M = M \cup j \quad \forall j \in Q$ .

Ir al paso 1.

## FASE 2

### Paso 1

$Y_0 = X_k$  pues en adelante solo habrá anticirculaciones.

### Paso 2

Colorear con la coloración para flujos conformables.

Blanco : si  $Y_i > 0$

Negro : si  $Y_i < 0$

Rojo : si  $Y_i = 0$

### Paso 3

Buscar una trayectoria cualquiera compatible con la coloración de la siguiente manera (por ejemplo).

+ Sean  $n_1, \dots, n_q$  los nodos de la gráfica

a)  $i := 1$

b) Aplicar el algoritmo de coloración para  $N^+ = \{n_i\}$  y  $N^- = \emptyset$ .

c) - Si se encontró un corte diferente a  $Q = [n_i, N / n_i]$ , se ha encontrado una trayectoria, ir al paso 4

- Si se encontró un corte de la forma  $Q = [n_i, N / n_i]$ :

• Si  $i \neq q$ , hacer  $i := i + 1$  e ir al inciso b)

• Si  $i = q$  no existe trayectoria compatible con la coloración.  
Ir al paso 5.

**Paso 4**

Sea  $\bar{P}_i$  dicha trayectoria calcular

$$\beta_i = \min\{Y_i(j) \mid \forall j \in \bar{P}_i\}$$

Actualizar el flujo :

$$Y_{i+1} = Y_i - \beta_i e_{\bar{P}_i}$$

Hacer  $i = i + 1$ .

Ir al paso 2

**Paso 5**

Escribir

$$X = \sum_{i=0}^k \alpha e_{P_i} + \sum_{j=0}^l \beta e_{\bar{P}_j}$$

donde si  $s=0$ , el flujo es una circulación y si  $k=0$ , el flujo es una anticirculación.

**FIN** del algoritmo.

El flujo quedó expresado como "combinación lineal de trayectorias elementales" a la vez de quedar expresado como la suma de una circulación y una anticirculación. El algoritmo se muestra de una manera más gráfica, mediante un diagrama de flujo, en la siguiente sección.

## SECCION 2.4

### JUSTIFICACION DEL ALGORITMO.

### DIAGRAMA DE FLUJO DEL ALGORITMO

### EJEMPLOS

Como dijimos anteriormente, la demostración del teorema de realización conforme se realizaría de manera constructiva mediante el algoritmo; sin embargo, siempre quedan algunos detalles por aclarar, como lo es el verificar que el algoritmo termina en un número finito de pasos; éste es el objetivo de la siguiente justificación.

Para facilitar dicha justificación haremos uso del siguiente concepto.

El soporte de un flujo  $X$ , es el conjunto  $S$  formado por los arcos  $j \in A$  tal que  $X(j) \neq 0$ , formado por dos subconjuntos :

$$S^+ = \{j \mid X(j) > 0\} \text{ y } S^- = \{j \mid X(j) < 0\}$$

### Justificación:

En el algoritmo, cada vez que se calcula un nuevo flujo, como  $X_k(j) \geq \alpha_k \forall j \in P_k^+$  y  $-X_k(j) \geq \alpha_k \forall j \in P_k^-$ , entonces el color de los arcos de  $P$ , con respecto al flujo  $X_{k+1}$ , es o bien el mismo o rojo. Además, el nuevo flujo  $X_{k+1}$  tiene soporte estrictamente incluido en el de  $X_k$  pues  $X_{k+1}(j) = 0$  al menos para  $j$  tal que  $\alpha_k = |X_k(j)| > 0$ , por lo que al menos un arco ya sea de la parte positiva o negativa del conjunto tiene como flujo igual a cero y por tanto es rojo.

La primera fase termina en un número finito de pasos, ya que, en cada etapa y en cada fin de iteración, por lo menos un arco se vuelve de color rojo, ya sea porque su flujo se convierte en cero, o queda incluido en  $M$  pues no pertenece a ningún circuito conformable. En la segunda fase sólo se podrán detectar anticirculaciones con los arcos de  $M$ .

En la segunda fase, cada vez que se calcula un nuevo flujo, por las mismas razones que en la primera fase, cada arco conserva su color o se vuelve rojo, además de que el nuevo flujo tiene su soporte estrictamente incluido en el inmediato anterior.

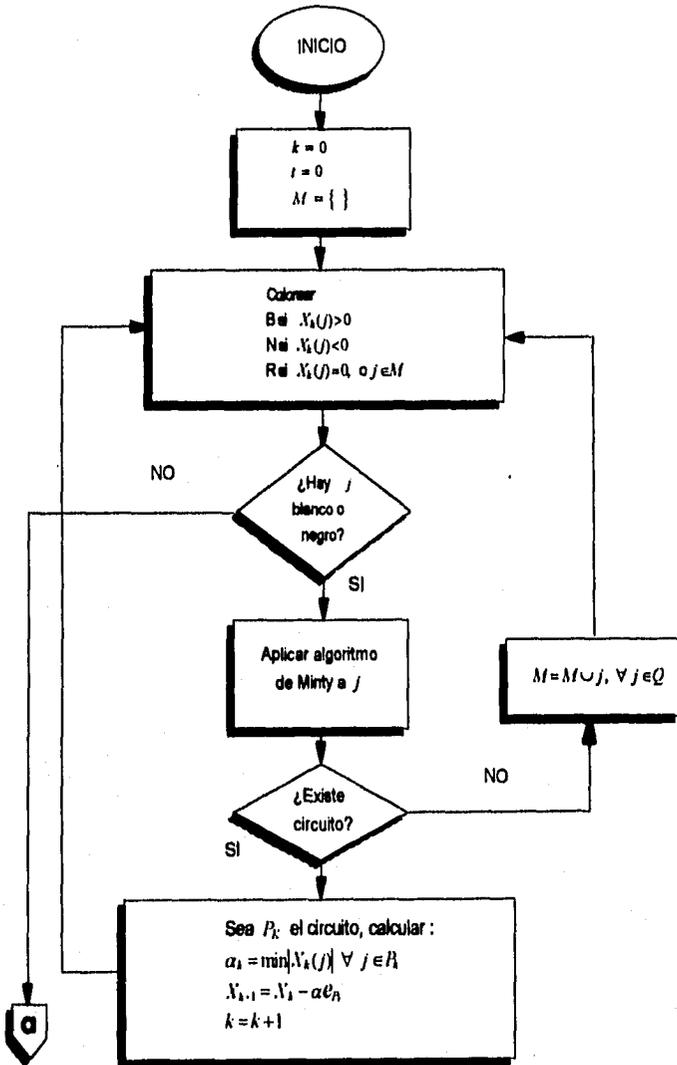
Si al entrar a la segunda fase, no se encuentra ninguna trayectoria compatible con la coloración, eso significa que  $Y_0 = 0$ , pues todos los arcos son rojos y entonces  $X$  es una circulación.

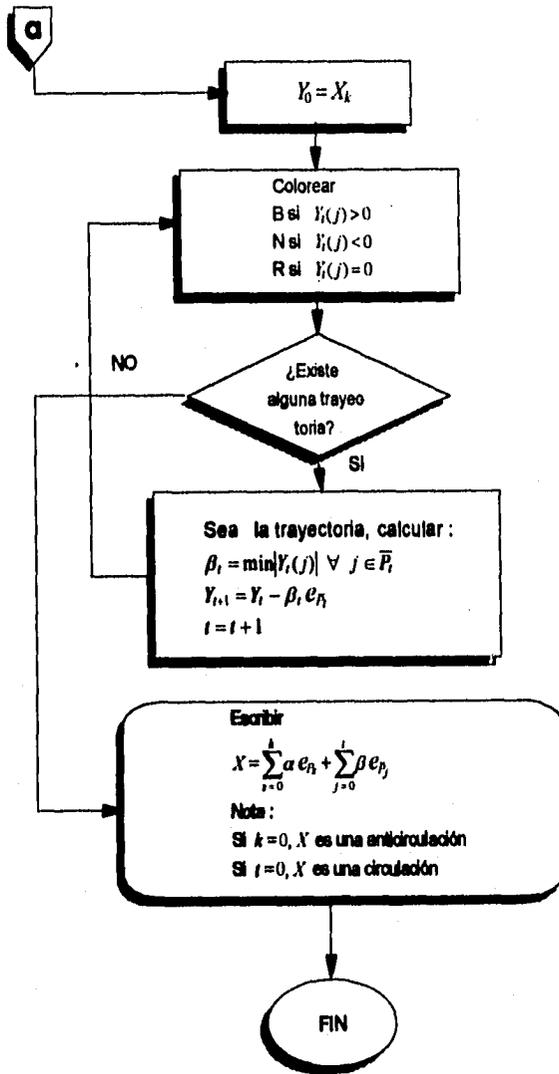
Si en la primera fase no fue encontrado ningún circuito, entonces por definición tenemos que  $X$  es una anticirculación ya que no hay circuitos que lo conformen.

La segunda fase también termina en un número finito de pasos ya que a lo más termina en  $|M|$  pasos.

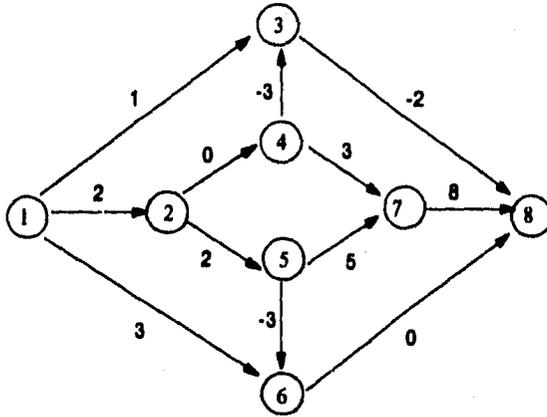
Notemos además que si  $X$  es entero, dada la definición de  $\alpha_0$  y  $\beta_0$ , estos también son enteros y como la función de incidencia está formada por ceros, unos y menos unos, dada la propiedad de cerradura en los enteros, los nuevos flujos  $X_k$  son también enteros, así todos los  $\alpha_i$  y  $\beta_m$  para  $i=0, \dots, K$  y  $m=0, \dots, T$  son enteros. ⊗

El diagrama de flujo que se presenta es el correspondiente al algoritmo de realización conformable.

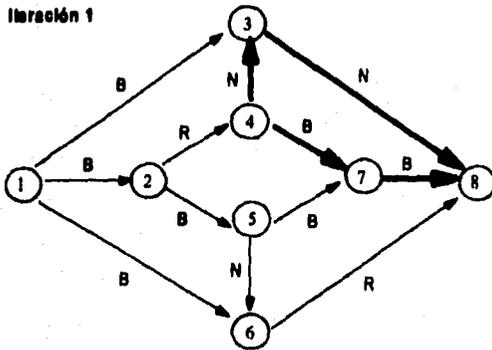




Para ilustrar el algoritmo daremos un ejemplo de su aplicación en la siguiente red :



Iteración 1

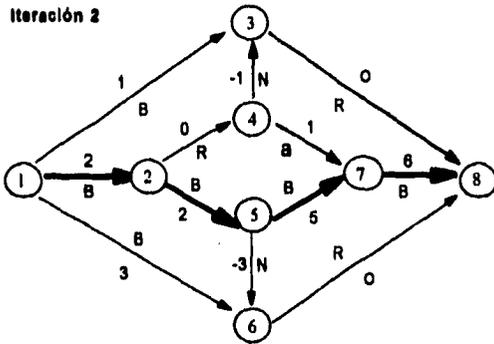


El arco (4, 3) de color negro y se aplica el algoritmo de Minty encontrando el siguiente circuito:

$$p_1: 3 \leftarrow 4 \rightarrow 7 \rightarrow 8 \leftarrow 3$$

$$\alpha_1 = \min \{ 3, 3, 8, 2 \} = 2$$

Iteración 2



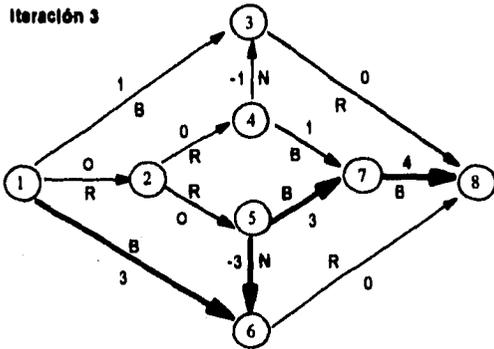
Se actualiza el flujo y ya no hay circuitos compatibles por lo que todos los arcos quedan de color rojo y se comienza con la fase 2 del algoritmo:

$$N^+ = 1, N^- = \phi$$

$$\bar{P}_1: 1 \rightarrow 2 \rightarrow 5 \rightarrow 7 \rightarrow 8$$

$$\beta_1 = \min \{ 2, 2, 5, 6 \} = 2$$

Iteración 3

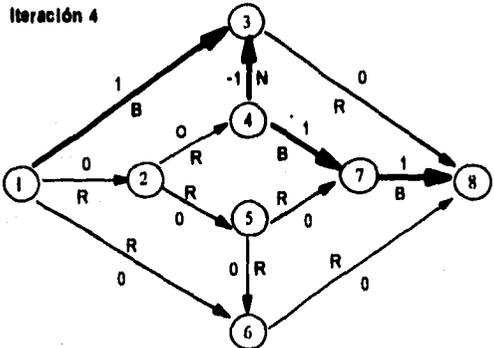


$$N^+ = 1, N^- = \phi$$

$$\bar{P}_2: 1 \rightarrow 6 \leftarrow 5 \rightarrow 7 \rightarrow 8$$

$$\beta_2 = \min \{ 3, 3, 3, 4 \} = 3$$

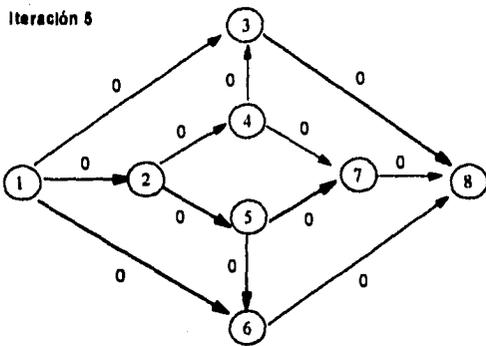
Iteración 4



$$N^+ = 1, N^- = \phi$$

$$\bar{P}_3: 1 \rightarrow 3 \leftarrow 4 \rightarrow 7 \rightarrow 8$$

$$\beta_3 = \min \{ 1, 1, 1, 1 \} = 1$$



y el flujo ha quedado expresado como :

$$X = 2 \cdot e_{P_1} + 2 \cdot e_{\bar{P}_1} + 3 \cdot e_{\bar{P}_2} + e_{\bar{P}_3}$$

Donde :

$$P_1: 3 \leftarrow 4 \rightarrow 7 \rightarrow 8 \leftarrow 3$$

$$\bar{P}_1: 1 \rightarrow 2 \rightarrow 5 \rightarrow 7 \rightarrow 8$$

$$\bar{P}_2: 1 \rightarrow 6 \leftarrow 5 \rightarrow 7 \rightarrow 8$$

$$\bar{P}_3: 1 \rightarrow 3 \leftarrow 4 \rightarrow 7 \rightarrow 8$$

## SECCION 2.5

### CASO ESPECIAL: CIRCULACIONES

Las circulaciones son un caso muy útil pues en la mayoría de los casos facilitan el trabajo ya que la teoría relacionada con ellas es más simple, además, cualquier flujo en una red puede ser visto como una circulación en la red aumentada (Primer capítulo, sección 1.4). Por estas razones muchos de los resultados serán obtenidos desde el punto de vista de las circulaciones.

En seguida enunciaremos el teorema de realización conformable para flujos para el caso especial de circulaciones; la demostración de este teorema es inmediata cuando se parte de una circulación por lo que el objetivo es demostrar su equivalencia con el caso general partiendo de un flujo cualquiera al que asociaremos una circulación en la red aumentada.

#### Teorema de realización conformable para el caso de circulaciones

Sea  $X$  una circulación distinta de cero, entonces existen  $P_1, \dots, P_r$  circuitos elementales que conforman a  $X$  y  $\alpha_1, \dots, \alpha_r, \alpha_i > 0 \forall i = 1, \dots, r$  que pueden ser escogidos enteros si  $X$  tiene componentes enteras, tales que

$$x = \alpha_1 e_{P_1} + \dots + \alpha_r e_{P_r}$$

#### Demostración (equivalencia con el teorema de realización conformable)

⇒

Supongamos que se tiene una solución expresada como en el teorema de realización conformable, ésta consta de dos sumandos: una circulación y una anticirculación. El primer sumando es de la misma forma que en el teorema de realización conformable de manera que

sólo resta demostrar que la anticirculación puede ser expresada como en el teorema de realización conformable para el caso de circulaciones. La anticirculación está expresada en términos de trayectorias elementales  $P$  y en particular cada término es un flujo elemental  $X_p$ , entonces en la red aumentada podemos construir un circuito  $C$  con esta trayectoria y dos arcos de distribución que conectan al nodo origen y al nodo destino de la trayectoria con el nodo de distribución; además el flujo a través de los arcos de distribución es igual a la divergencia de su extremo perteneciente a la red original por lo que el circuito resultante conforma al siguiente flujo:

$$\bar{X}_p(j) = \begin{cases} X_p(j) & \text{si } j \in C \\ 0 & \text{en otro caso} \end{cases}$$

$\bar{X}_p(j)$  resulta ser un flujo elemental que conforma a  $C$  y es una circulación por lo que está expresado como en el teorema de realización conformable para circulaciones.

←

Las expresiones como en el teorema de realización conformable para circulaciones no son únicas, de manera que se construirá una de la siguiente forma:

1. Encontrar la descomposición del flujo  $X$  como sigue

$$X = \sum_{j=1}^q \alpha_j e_{P_j} + X_a$$

donde  $\alpha_j e_{P_j}$ ,  $j = 1, \dots, q$  es una circulación que conforma al circuito  $P_j$ , que no contiene arcos de distribución y  $X_a$  un flujo sin circuitos conformables que no contengan arcos de distribución.

2. Encontrar la descomposición de  $X_a$ :

$$X_a = \sum_{k=q+1}^r \alpha_k e_{P_k}$$

$$X = \sum_{i=1}^q \alpha_i e_{h_i} + \sum_{k=q+1}^r \alpha_k e_{h_k}$$

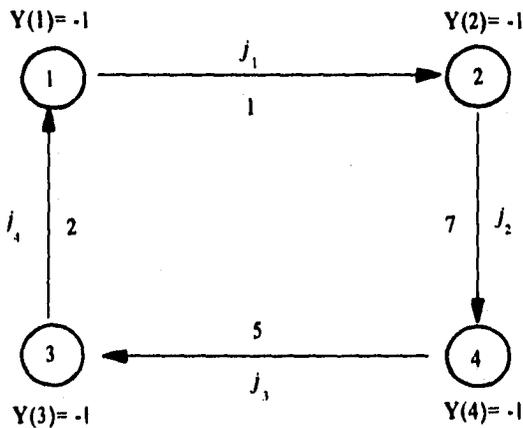
es una circulación y está expresada como en el teorema de realización conformable para circulaciones. Retiremos los arcos de distribución de la red. Sean  $P_i^k$   $i=1, \dots, s$ ,  $k=q+1, \dots, r$  las trayectorias resultantes de retirar dichos arcos del circuito  $P_k$   $k=q+1, \dots, r$  (al menos una por circuito), entonces

$$\sum_{k=q+1}^r \alpha_k \sum_{i=1}^s e_{P_i^k}$$

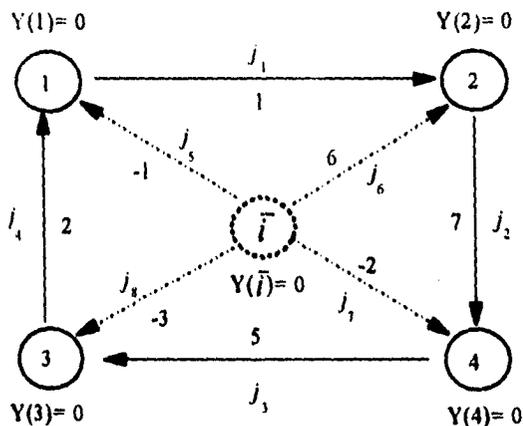
es una anticirculación puesto que la red original (sin arcos de distribución) no contiene circuitos que conformen a  $X_s$ . Esta expresión es igual a la del teorema de realización conformable. Por lo tanto tenemos al flujo expresado como la suma de una circulación y una anticirculación así como combinación lineal de trayectorias elementales como en el teorema de realización conformable. ⊗

En el siguiente ejemplo se obtendrá la descomposición como en el teorema de realización conformable para el caso de circulaciones de la forma que se expuso en la demostración de la equivalencia con el teorema para el caso general. Posteriormente procederemos a convertirla a la expresión del teorema de realización conformable.

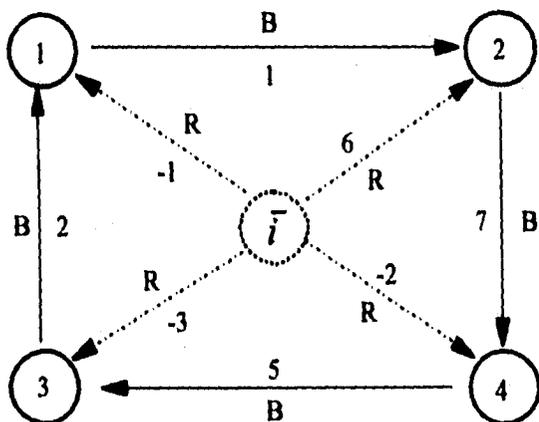
Sea la siguiente red:



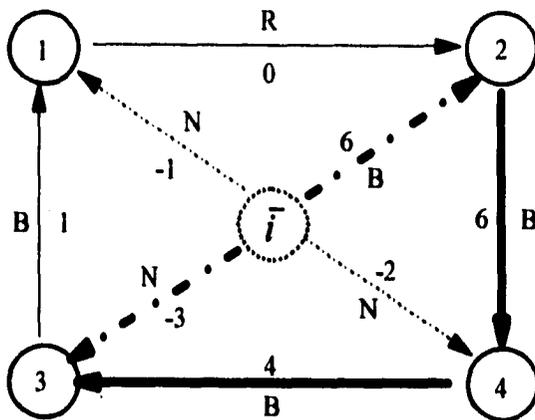
La circulación asociada a este flujo en la red aumentada es la siguiente:



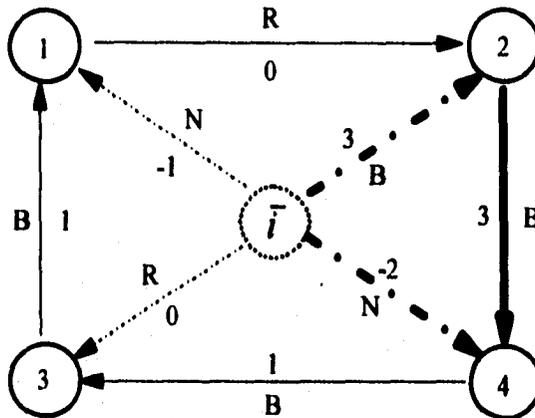
A continuación se muestran los pasos para encontrar la descomposición tal y como se expone en la demostración de la equivalencia de ambos teoremas:



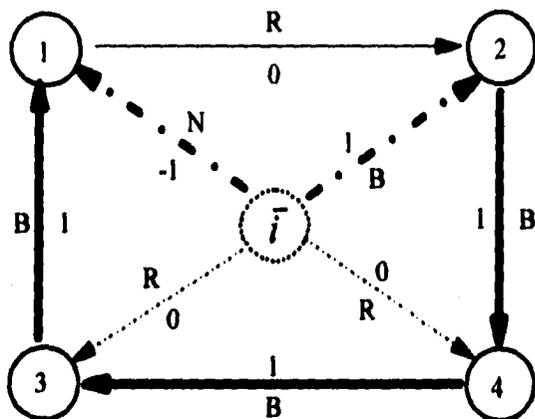
$$\alpha_1 = 1, \quad e_{p_1} = (1, 1, 1, 1, 0, 0, 0), \quad X_o = (0, 6, 4, 1, -1, 6, -2, -3); \quad X = \alpha_1 e_{p_1} + X_o$$



$$\alpha_2 = 3, \quad e_{p_2} = (0, 1, 1, 0, 0, 1, 0, -1)$$



$$\alpha_3 = 2, \quad e_{p_3} = (0, 1, 0, 0, 0, 1, -1, 0)$$



$$\alpha_4 = 1, \quad e_{p_4} = (0, 1, 1, 1, -1, 1, 0, 0)$$

Sólo el circuito  $P_1$  no contiene arcos de distribución; los vectores de incidencia de las trayectorias resultantes de eliminar los arcos de distribución son las siguientes

$$e_{p_1}^2 = (0, 1, 1, 0)$$

$$e_{p_1}^3 = (0, 1, 0, 0)$$

$$e_{p_1}^4 = (0, 1, 1, 1)$$

Entonces:

$$X = \alpha_1 e_{p_1} + \alpha_2 e_{p_1}^2 + \alpha_3 e_{p_1}^3 + \alpha_4 e_{p_1}^4$$

es una solución expresada como en el teorema de realización conformable, donde el primer término es una circulación y la suma de los tres restantes una anticirculación.

A partir de lo visto es fácil demostrar que una característica de una circulación  $X$  que es un flujo elemental es que su soporte es no vacío y no existe ninguna circulación  $X'$  cuyo soporte sea no vacío y esté estrictamente incluido en el soporte de  $X$ , este resultado queda así :

### Proposición

Una circulación  $X$  es un flujo elemental si y sólo si su soporte es no vacío y minimal, en el sentido de que no existe ninguna circulación  $X'$  cuyo soporte sea no vacío y esté estrictamente incluido en el soporte de  $X$ .

### Demostración

=> Dado que  $X$  es un flujo elemental, entonces  $X = \alpha e_P$ , donde  $P$  es una trayectoria elemental, de hecho es un circuito ya que  $X$  es una circulación.

Sea  $X'$  un flujo tal que su soporte está contenido estrictamente en el soporte de  $X$ , es obvio que el soporte de  $X'$  no contiene un circuito (ya que el soporte de  $X$  es un circuito elemental), y por lo mismo  $X'$  no es una circulación. Por lo tanto no existe ninguna circulación  $X'$  cuyo soporte sea no vacío y esté estrictamente incluido en el soporte de  $X$ .

<= Sea  $X$  una circulación no elemental, entonces dado el teorema de realización

$$X = \alpha_1 e_{P_1} + \dots + \alpha_k e_{P_k} \quad \text{donde } \alpha_i > 0 \text{ y } e_{P_i} \text{ es un circuito elemental } \forall i = 1, \dots, k, k \geq 2$$

entonces

$$X' = \alpha_1 e_{P_1} + \dots + \alpha_k e_{P_k} \quad \text{donde } \alpha_i > 0 \text{ y } e_{P_i} \text{ es un circuito elemental } \forall i = 1, \dots, k$$

Notemos que  $X' \neq X$  pues  $\alpha_1 > 0$  y además  $X'$  es una circulación cuyo soporte es no vacío y está estrictamente incluido en el soporte de  $X$ . Por lo tanto si no existe ninguna circulación  $X'$  cuyo soporte sea no vacío y estrictamente incluido en el soporte de  $X$ , entonces  $X$  es un flujo elemental. ⊗

De hecho, se puede probar que el único flujo cuyo soporte está incluido en el de una circulación elemental es un múltiplo de ésta como se demuestra a continuación.

### Corolario

Sea  $X \neq 0$  una circulación elemental y  $X' \neq 0$  una circulación cuyo soporte está incluido en el de  $X$ , entonces  $X' = \lambda X$  para alguna  $\lambda \neq 0$ .

Dado que  $X$  es circulación elemental, entonces  $X = \alpha e_c$ ,  $\alpha \neq 0$  donde  $C$  es un circuito elemental. Por el ejercicio anterior, el soporte de  $X'$  no puede estar estrictamente contenido en el de  $X$  pues dejaría de ser elemental, así el soporte de  $X'$  es igual al soporte de  $X$ , el soporte de  $X'$  contiene al mismo circuito elemental  $C$ . Además  $X'$  es circulación elemental pues de no serlo, aquella circulación cuyo soporte es no vacío y contenido en  $X'$  también lo estaría en  $X$  contradiciendo la hipótesis de que  $X$  es una circulación elemental.

Entonces

$$X' = \alpha' e_c$$

$$X = \alpha e_c$$

$\Rightarrow$

$$e_c = X'/\alpha'$$

$$e_c = X/\alpha$$

Así

$$\frac{X'}{\alpha'} = \frac{X}{\alpha}, X = \frac{\alpha}{\alpha'} X'$$

$\Rightarrow$

$$X = \lambda X' \text{ y } \lambda \neq 0 \text{ pues si no lo fuera}$$

$$\alpha = 0 \text{ y } X = \alpha e_c = 0. \otimes$$

## SECCION 2.6

### RESULTADOS DE LA TEORIA DE ARBOLES Y BOSQUES

#### TEOREMA DE EXPANSION

Veremos algunos resultados acerca de árboles y bosques que serán de total importancia para la comprensión y demostración de muchos resultados del capítulo en curso.

He aquí uno de ellos que caracteriza los bosques en términos de flujos.

**Proposición**

Sea  $F \subset A$  entonces  $F$  es un bosque  $\Leftrightarrow$  no existe una circulación  $X \neq 0$  tal que su soporte esté incluido en  $F$ .

**Demostración**

$\Rightarrow$

Esto es fácil de comprobar por contradicción

Sea una circulación tal que su soporte está incluido en  $F$ . Tal soporte contiene por lo menos un circuito si  $X \neq 0$ . Contradicción.

$\Leftarrow$

Recurriendo de nuevo a la demostración por contradicción, si tenemos que en esa gráfica no existe circulación alguna distinta de cero cuyo soporte esté incluido en  $F$  y suponemos que  $F$  no es un bosque, entonces la gráfica tiene al menos un circuito  $P$  por lo que podemos asociar a los arcos un flujo igual al valor de la función de incidencia  $e_p$  en tales arcos y llegamos a una contradicción pues el flujo resultante es una circulación distinta de cero.  $\otimes$

Así mismo, de la definición y del resultado anterior podemos concluir:

**Proposición**

Un conjunto de arcos  $F$  es un bosque en  $G$  si y sólo si las columnas correspondientes a la matriz de incidencia de  $G$  son linealmente independientes.

**Demostración**

$\Rightarrow$  Si  $X$  es una circulación distinta de cero, las columnas de la matriz de incidencia son linealmente dependientes ya que observando la definición de circulación

$$\sum_{j \in A} e(i, j) X(j) = 0 \quad \forall i \in N$$

tenemos una combinación lineal de las columnas de la matriz de incidencia que da como resultado el vector cero y cuyos coeficientes  $X(j)$  no son todos cero, ya que, la circulación es distinta de cero. Ahora probaremos la propiedad por contradicción. Supongamos que los arcos de la matriz de incidencia correspondientes a  $F$  forman un conjunto de vectores linealmente dependientes entonces tendríamos que existen escalares  $\alpha_j$ , no todos cero tales que:

$$\sum_{j \in F} e(i, j) \alpha_j = 0 \quad \forall i \in N$$

Si definimos el flujo  $X$  como  $X(j) = \alpha$ , si  $j \in F$  y  $X(j) = 0$  si  $j \notin F$  habríamos encontrado una circulación distinta de cero cuyo soporte está incluido en  $F$  y por el resultado anterior  $F$  no sería un bosque.

⇐ Para el regreso de la demostración tenemos que:

$$\sum_{j \in F} e(i, j) X(j) = 0 \Leftrightarrow X(j) = 0 \quad \forall j \in F$$

puesto que partimos de columnas linealmente independientes entonces la única circulación existente es el vector cero, por lo tanto  $F$  es un bosque. ⊗

El teorema de expansión enuncia la equivalencia de ciertos postulados que incluyen parte de la teoría expuesta dándonos una visión más amplia.

### Teorema de expansión

Para cualquier red  $G$  las siguientes propiedades de un conjunto de arcos  $F \subset A$  son equivalentes

1.  $F$  forma un bosque maximal en  $G$ .
2.  $F$  forma un árbol de expansión para cada componente conexa de  $G$
3.  $F$  forma un bosque tal que  $|F| = |N| - p$ , donde  $p$  es el número de componentes conexas de  $G$
4. Las columnas  $e^j$  de la matriz  $E$  de incidencia correspondientes a los arcos  $j \in F$  forman una base para el espacio de columnas de  $E$ .

### Demostración

1 ⇒ 4

Para que un cierto conjunto sea una base, se necesita que los elementos sean linealmente independientes y que tal conjunto sea maximal. Sea  $C = \{e^j \text{ de } E \mid j \in F\}$ , que los elementos de  $F$  sean independientes ya se mostró; ahora, sea un conjunto que contenga a  $C$  y al menos un elemento más que fuera linealmente independiente, este nuevo conjunto no sería una base, pues debido a la definición y propiedades de un bosque maximal, si añadimos al menos un arco más a un bosque, la nueva gráfica contendría al menos un circuito y entonces las columnas de  $C$  junto con la asociada al arco agregado formarían un conjunto linealmente dependiente, por lo tanto, es un conjunto maximal de elementos linealmente independientes por lo cual es base.

4 ⇒ 1

Dado que las columnas de la matriz de incidencia correspondientes a los arcos  $j \in F$  forman una base, entonces por el resultado visto al inicio de esta sección  $F$  es un bosque en  $G$  así lo único que resta demostrar es su maximalidad pero esto es obvio ya que el conjunto de columnas también lo es y están en relación uno a uno con los arcos del bosque.

1  $\Rightarrow$  2

Sean  $G_i = (N_i, A_i)$ ,  $i=1, \dots, c$  las componentes conexas de  $G$  sin incluir a los nodos aislados y  $F_i = G_i \cap F$  para toda  $i=1, \dots, c$ . Si  $F_i$  no forma un árbol de expansión para alguna  $i=1, \dots, c$ , entonces existe  $\bar{j} \in A_i$  tal que  $\bar{j} \in F \cap G_i$  no forma ningún circuito en la red  $F_i \cup \bar{j}$ ; por lo tanto  $F_i \cup \bar{j}$  es un bosque lo cual contradice el supuesto de maximalidad de  $F$ .

Que 2  $\Rightarrow$  3

Sea  $p$  el número de componentes de  $G$ ; sabemos que para cada una de ellas el número de arcos en el árbol de expansión correspondiente es el número de nodos en la componente  $G_i$  menos 1, entonces, la suma total es

$$|F| = \sum_{i=1}^{i=p} |N_i| - 1 = |N| - p$$

Que 3  $\Rightarrow$  1

De aquí también tenemos que 3 implica 1 ya que si el bosque no fuera maximal entonces al menos para alguna componente tendríamos que el número de arcos que pertenecen al árbol contenido en ella es menor al número de sus nodos menos uno y entonces la igualdad presentada arriba no se daría.  $\otimes$

Como conclusión obvia de lo que hemos visto se tiene al siguiente resultado,

**Proposición:**

En una red conexa un árbol de expansión es un bosque maximal.

## SECCION 2.7

### REPRESENTACIONES DE TUCKER DEL ESPACIO DE CIRCULACIONES

A partir de aquí comenzamos a notar una familiaridad con la programación lineal. Los problemas a resolver en programación lineal involucran un sistema de ecuaciones  $EX=b$  como restricción, donde  $E$  es una matriz de  $n \times m$  y  $X$  es un vector en  $R^m$ ; estas ecuaciones pueden ser escritas como

$$\sum_{j \in A} X(j) e^j$$

Si  $R$  es una red y  $A$  es su conjunto de arcos, entonces el sistema de ecuaciones anterior define condiciones de divergencia para el flujo  $X$ , en las redes nos interesan los conjuntos  $F \subset A$ , para los cuales las columnas  $e^j$  de la matriz de incidencia para  $j \in F$  forman una base para el espacio columna (suponiendo que  $b$  está contenido en el espacio generado por las columnas). Cada uno de estos conjuntos determina una solución  $X$  única para la cual  $X(j)=0$  para toda  $j \notin F$ . El método simplex genera una secuencia de estos conjuntos. Gráficamente, estos conjuntos pueden ser identificados con los árboles de expansión de la red, suponiendo que la red es conexa.

Las representaciones de Tucker que veremos a continuación, nos ayudarán a determinar las bases; una **representación de Tucker** del espacio  $\mathcal{C}$  es un sistema de ecuaciones que está resuelto para algunas de las variables  $X(j)$  en términos de las variables restantes  $X(k)$ . En lenguaje matemático se escribiría de la siguiente manera. para varios subconjuntos  $F \subset A$ , el sistema

$$\sum_{k \in F'} a(j,k) X(k) = X(j) \quad \forall j \in F, \quad F' = A \setminus F \quad (I)$$

es una representación de Tucker de  $\mathcal{C}$ . Si el flujo para los arcos de  $F'$  fue escogido arbitrariamente entonces esta ecuación nos dará la única combinación de valores del flujo a través de los arcos de  $F$  que hacen que el flujo resultante sea una circulación.

El siguiente resultado, es de gran utilidad por relacionar la teoría vista con los algoritmos de redes.

### Proposición

Un conjunto de arcos  $F \subset A$  corresponde a una representación de Tucker del espacio de circulaciones  $\mathcal{C}$  si y sólo si  $F$  forma un bosque maximal para la red  $G$ .

### Demostración

$F$  corresponde a una representación de Tucker del espacio de circulaciones  $\mathcal{C}$  si y sólo si para cada elección de números  $X(k)$ ,  $k \in F'$ , existen números únicos  $X(j)$ ,  $j \in F$  tales que

$$\sum_{j \in F} X(j)e^j = - \sum_{k \in F'} X(k)e^k \quad (2)$$

Ahora demostraremos que los vectores  $e^j$ ,  $j \in F'$  forman una base para el espacio de columnas de  $E$ .

Busquemos un flujo para esta red fijando  $X(k) = 0$ ,  $k \in F'$ ; entonces el lado derecho de la ecuación (2) se hace cero. Sabemos que mediante (1) encontramos los valores únicos de los flujos de  $F$  que hacen que el flujo a través de los arcos sea una circulación. Así mismo, si  $X(j) = 0$  para toda  $j \in F$  la igualdad se cumple; como los valores de  $X(j)$  para  $j \in F$  son únicos, entonces, la combinación lineal de las columnas  $e^j$  en la ecuación (2) es cero si y solo si los escalares de la misma son cero y por lo tanto son linealmente independientes, además, este conjunto genera el espacio columna ya que cada columna  $e_{k_i}$  de la matriz de incidencia  $E$  correspondiente a  $k_i \in F'$  puede ser expresada como combinación lineal de las columnas de la matriz  $E$  de incidencia, correspondientes a los arcos  $j \in F$ , mediante los siguientes valores arbitrarios

$$X(k) = \begin{cases} 1 & \text{si } k = k_i \\ 0 & \text{si } k \neq k_i \end{cases} \quad k \in F'$$

Encontramos los valores únicos de  $X(j)$  para  $j \in F$  que cumplen con la ecuación (2) y obtenemos la combinación lineal buscada:

$$\sum_{j \in F} X(j)e^j = - \sum_{k \in F'} X(k)e^k = -e^{k_i}$$

entonces

$$e^{k_i} = - \sum_{j \in F} X(j) e^j$$

Este procedimiento nos proporciona cada una de las columnas de la matriz  $E$  de incidencia correspondientes a los arcos  $k \in F^*$  expresados como combinación de las columnas correspondientes a los arcos  $j \in F$  de la misma matriz, entonces estas últimas generan el espacio columnar y por lo tanto forman una base para dicho espacio; de aquí mediante el teorema de expansión, encontramos que este conjunto es un bosque maximal para  $G$ , que sería equivalente a tener un conjunto de ecuaciones que restringen al flujo de manera que sea una circulación, pues están igualadas a cero. ⊗

Además, podemos conocer con anticipación la dimensión del espacio de circulaciones. En cualquier red, la dimensión del espacio  $\mathcal{C}$  de circulaciones está dada por  $\dim \mathcal{C} = |A| - |N| + p$ , donde  $p$  es el número de componentes conexas de la red. En particular, para una red conexa  $\dim \mathcal{C} = |A| - |N| + 1$ ; a la dimensión del espacio de circulaciones cuando se tiene una red conexa, se le llama número cicломático de la gráfica  $G$ .

Esta relación se da, dado que las variables básicas que corresponden a los arcos de un bosque maximal son expresadas de manera única por las demás variables, estas demás variables son  $|A| - (|N| - 1) = |A| - |N| + 1$ , ya que  $|N| - 1$  es el número de arcos en un árbol de expansión (ya que la red es conexa) y por tanto la dimensión del espacio, es igual al número de variables no básicas; así para una red con  $p$  componentes conexas, sabemos que el bosque tiene  $|N| - p$  arcos, y la dimensión de  $\mathcal{C}$  para una red de estas será  $|A| - |N| + p$ . ⊗

## SECCION 2.8

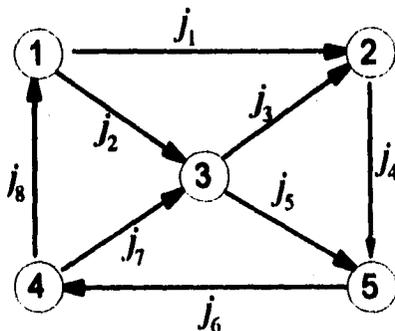
### REPRESENTACION INICIAL DE TUCKER

Obtener una representación inicial de Tucker para un subespacio  $\mathcal{C}$  es sencillo :

Se parte de la matriz de incidencia (arcos contra nodos) y se hacen operaciones elementales sobre renglones hasta obtener una matriz que tenga  $N - 1$  columnas con un elemento -1 y los demás elementos de la columna iguales a cero.

Los arcos representados por estas columnas serán los elementos de la base correspondientes a la representación de Tucker. Al final deberán quedar  $N - 1$  ecuaciones pues si hubiera más éstas serían redundantes. Para finalizar, los elementos de dicha base se expresarán del lado derecho de la matriz en el renglón correspondiente a la única ecuación en donde tienen coeficiente distinto de cero.

El ejemplo siguiente dejará el procedimiento más claro. Dada la siguiente red



### Paso 1

La matriz de incidencia

	$x(1)$	$x(2)$	$x(3)$	$x(4)$	$x(5)$	$x(6)$	$x(7)$	$x(8)$
$n_1$	1	1	0	0	0	0	0	-1
$n_2$	-1	0	-1	1	0	0	0	0
$n_3$	0	-1	1	0	1	0	-1	0
$n_4$	0	0	0	0	0	-1	1	1
$n_5$	0	0	0	-1	-1	1	0	0

Haciendo operaciones elementales sobre renglones a los cuales llamaremos  $R_i$ ,  $i=1, \dots, 5$   
 ( $R_3 + R_1 \rightarrow R_1$ )

	$x(1)$	$x(2)$	$x(3)$	$x(4)$	$x(5)$	$x(6)$	$x(7)$	$x(8)$
$n_1$	1	0	1	0	1	0	-1	-1
$n_2$	-1	0	-1	1	0	0	0	0
$n_3$	0	-1	1	0	1	0	-1	0
$n_4$	0	0	0	0	0	-1	1	1
$n_5$	0	0	0	-1	-1	1	0	0

(R2 + R1 → R1)

	x(1)	x(2)	x(3)	x(4)	x(5)	x(6)	x(7)	x(8)
$n_1$	0	0	0	1	1	0	-1	-1
$n_2$	-1	0	-1	1	0	0	0	0
$n_3$	0	-1	1	0	1	0	-1	0
$n_4$	0	0	0	0	0	-1	1	1
$n_5$	0	0	0	-1	-1	1	0	0

(R1 + R4 → R4)

	x(1)	x(2)	x(3)	x(4)	x(5)	x(6)	x(7)	x(8)
$n_1$	0	0	0	1	1	0	-1	-1
$n_2$	-1	0	-1	1	0	0	0	0
$n_3$	0	-1	1	0	1	0	-1	0
$n_4$	0	0	0	1	1	-1	0	0
$n_5$	0	0	0	-1	-1	1	0	0

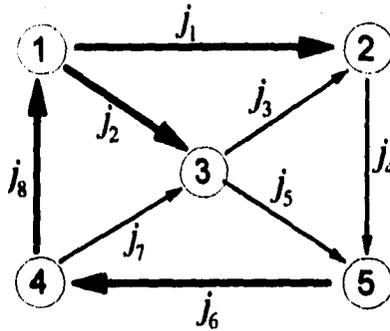
(R4 + R5 → R5)

	x(1)	x(2)	x(3)	x(4)	x(5)	x(6)	x(7)	x(8)
$n_1$	0	0	0	1	1	0	-1	-1
$n_2$	-1	0	-1	1	0	0	0	0
$n_3$	0	-1	1	0	1	0	-1	0
$n_4$	0	0	0	1	1	-1	0	0
$n_5$	0	0	0	0	0	0	0	0

El renglón  $n_5$  es redundante de manera que lo quitamos. Despejando las variables correspondientes a las columnas con una entrada igual a -1 y las demás iguales a cero obtenemos el sistema expresado en la siguiente tabla:

x(3)	x(4)	x(5)	x(7)	
0	1	1	-1	x(8)
-1	1	0	0	x(1)
1	0	1	-1	x(2)
0	1	1	0	x(6)

En esta representación, la base corresponde al conjunto de arcos  $F$  donde  $F = \{j_8, j_1, j_2, j_6\}$  y  $F' = \{j_3, j_4, j_5, j_7\}$ :



El siguiente resultado es de suma importancia y será fundamental para éste y el siguiente capítulo; la primera parte que es la que se utiliza principalmente en este capítulo será demostrada aquí, y la parte restante se dejará para el próximo capítulo.

## SECCION 2.9

### TEOREMA DE BASES

#### COMO OBTENER MAS FACILMENTE UNA REPRESENTACION INICIAL DE TUCKER

El teorema de bases que enunciaremos será una de nuestras mejores herramientas. En este capítulo sólo se demostrará la primera parte del mismo dejando para el próximo, en la sección 3.4, la demostración de la segunda parte pues es hasta entonces que se comienza a profundizar en el espacio de diferenciales.

#### Teorema de bases

Sea  $F \subset A$  el conjunto de arcos correspondiente a un bosque maximal en  $G$  y sean  $a(j, k)$  para  $j \in F, k \in F^c = A \setminus F$  los coeficientes de la representación de Tucker del espacio  $\mathcal{C}$  de circulaciones y  $\mathcal{D}$  de diferenciales

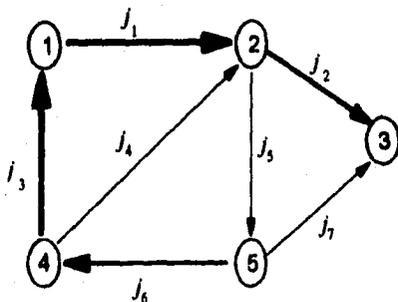
1. Para cada arco  $k \in F'$  existe un único circuito elemental  $P_k$  que contiene a  $k$  en su parte positiva y los arcos restantes del circuito pertenecen todos solamente a  $F$ . Los vectores de incidencia  $e_{P_k}$  forman una base para  $\mathcal{C}$  y están dados por

$$e_{P_k}(j) = \begin{cases} a(j, k) & \forall j \in F \\ 0 & \forall j \in F' \text{ para } j \neq k \\ 1 & \text{para } j = k \end{cases}$$

2. Para cada arco  $j \in F$ , existe un único corte elemental  $Q_j$  que contiene a  $j$  en su parte positiva y los demás arcos del corte pertenecen únicamente a  $F'$ . Los vectores de incidencia  $e_{Q_j}$  forman una base para  $\mathcal{D}$  y están dados por

$$e_{Q_j}(k) = \begin{cases} -a(j, k) & \forall k \in F' \\ 0 & \forall k \in F \text{ para } k \neq j \\ 1 & \text{para } k = j \end{cases}$$

Para ilustrar el primer caso del teorema de bases considérese la siguiente red con un bosque maximal asociado (en arcos oscuros):



Para este bosque maximal tenemos que  $F' = \{j_4, j_5, j_7\}$ , entonces encontraremos circuitos formados por un arco de  $F'$  (recorrido positivamente) y los restantes de  $F$ :

$F'$	Arcos del circuito
$j_4$	$j_4, j_3, j_7$
$j_5$	$j_5, j_6, j_3, j_1$
$j_7$	$j_7, j_2, j_1, j_3, j_6$

Estos circuitos son los que nos darán la representación de Tucker como veremos posteriormente

### Demostración

#### Parte 1

Sea  $k \in F'$ ,  $k \sim (i, i')$ , dado que  $F$  es un bosque maximal, sabemos que para toda componente conexa de él tenemos un árbol de expansión, si tomamos la componente conexa que contiene a  $k$  y el respectivo árbol de expansión, entonces existe una única trayectoria formada por arcos de  $F$  que va de  $i'$  a  $i$  formando, junto con  $k$ , el circuito  $P_k$ , su vector de incidencia  $e_{P_k}$  puede ser visto como una circulación que tiene un flujo +1 en el circuito y 0 en los arcos que se encuentran fuera de él. Entonces para cualquier elección de coeficientes  $\lambda_k$  el siguiente flujo es una circulación

$$X = \sum_{k \in F'} \lambda_k e_{P_k}$$

y satisface que  $X(k) = \lambda_k$  para toda  $k \in F'$ , la  $j$ -ésima componente del vector anterior para  $j \in F'$  es:

$$X(j) = \sum_{k \in F'} \lambda_k e_{P_k}(j) \quad \forall j \in F'$$

dado que existe una representación de Tucker para  $F$ , entonces la circulación acabada de mencionar es única, por lo que

$$\sum_{k \in F'} \lambda_k a(j, k) = \sum_{k \in F'} \lambda_k e_{P_k}(j) \quad \forall j \in F' \quad \text{y} \quad \forall \lambda_k \in \mathbb{R}.$$

ésto muestra que los vectores  $e_{P_k}$  forman una base para  $\mathcal{C}$ , y  $e_{P_k}(j) = a(j, k)$ .  $\otimes$

Directamente de este teorema, obtenemos un algoritmo mucho más sencillo (al menos manualmente) para encontrar una representación inicial de Tucker, a partir de un bosque maximal cualquiera. El algoritmo queda bajo escrito como sigue

### Algoritmo

#### Paso 1

Encontrar un bosque maximal  $F$

#### Paso 2

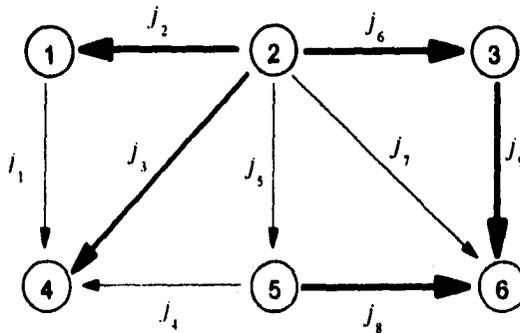
Para todo arco  $k$  no perteneciente al bosque  $F$ , encontrar el circuito  $P_k$  tal que  $k$  es recorrido positivamente, donde  $P_k$  está compuesto únicamente por arcos de  $F$  a excepción de  $k$ . Hallar el vector de incidencia  $e_{P_k}$  correspondiente a  $P_k$

#### Paso 3

Formar la representación de Tucker para la base compuesta por los arcos de  $F$ , como sigue :

Para cada  $k \in F'$ , la columna correspondiente en la representación, queda determinada por  $a(j,k) = e_{P_k}(j) \quad \forall j \in F$

Para dejar más claro el procedimiento se calculará, mediante el algoritmo, la representación de Tucker de  $\mathcal{Q}$  y  $\mathcal{D}$  correspondiente al árbol de expansión  $F$ , que es el formado por los arcos más oscuros, en la siguiente red.



Vectores de incidencias de los circuitos  $P_k$ , con  $k \in F'$

$k \in F'$	$P_k^+$	$P_k^-$	$e_{P_k}$									
			$j_1$	$j_2$	$j_3$	$j_4$	$j_5$	$j_6$	$j_7$	$j_8$	$j_9$	
$j_1$	$j_1, j_2$	$j_3$	1	1	-1	0	0	0	0	0	0	0
$j_4$	$j_4, j_6, j_9$	$j_3, j_8$	0	0	-1	1	0	1	0	0	-1	1
$j_5$	$j_5, j_8$	$j_9, j_6$	0	0	0	0	1	-1	0	0	1	-1
$j_7$	$j_7$	$j_9, j_6$	0	0	0	0	0	0	-1	1	0	-1

Representación de Tucker  
 $(a(j, k) = e_{P_k}(j))$  para toda  $j \in F$

$X(1)$	$X(4)$	$X(5)$	$X(7)$
1	0	0	0
-1	-1	0	0
0	1	-1	-1
0	-1	1	0
0	1	-1	-1

SECCION 2.10

PIVOTEO : CAMBIO DE UNA BASE A OTRA

A las operaciones necesarias para lograr pasar de una representación de Tucker a otra se le llama pivoteos, de la misma manera que en programación lineal con el método simplex para pasar de una base a otra. En lugar de hacer operaciones manualmente renglón por renglón podemos ver la ecuación final que resume los pasos para obtener una base a partir de otra intercambiando un arco  $\bar{j} \in F$  por uno  $\bar{k} \in F'$ , si es que existe dicha base.

Empezaremos con una tabla de Tucker como la que vimos en la sección 2.7 y consideraremos cualquier par de arcos  $\bar{j} \in F$  y  $\bar{k} \in F'$  tal que  $a(j, k) \neq 0$ . La ecuación representada por el renglón  $\bar{j}$  de la tabla es el siguiente :

$$\sum_{k \in F'} a(j, k) X(k) = X(\bar{j})$$

Resolviendo para  $X(\bar{k})$

$$X(\bar{k}) = \frac{1}{a(\bar{j}, \bar{k})} X(\bar{j}) - \sum_{\substack{k \in F \\ k \neq \bar{k}}} \frac{a(\bar{j}, k)}{a(\bar{j}, \bar{k})} X(k) \quad (1)$$

Sustituyendo en las ecuaciones representadas por los otros renglones se obtiene

$$X(j) = \frac{a(j, \bar{k})}{a(\bar{j}, \bar{k})} X(\bar{j}) + \sum_{\substack{k \in F \\ k \neq \bar{k}}} \left[ a(j, k) - \frac{a(j, \bar{k})a(\bar{j}, k)}{a(\bar{j}, \bar{k})} \right] X(k) \text{ para cada } j \in F, j \neq \bar{j} \quad (2)$$

Esta es una nueva representación de Tucker del espacio de circulaciones y es equivalente a la que teníamos ya que hicimos únicamente operaciones elementales sobre renglones. Esta nueva representación es aquella que tiene como base a  $\bar{F} = [F \setminus \bar{j}] \cup \bar{k}$

A la transformación hecha en la tabla asociada a  $F$  para obtener la tabla asociada a  $\bar{F}$  se le llama pivoteo en  $(\bar{j}, \bar{k})$ . Este pivoteo es posible solamente si  $a(\bar{j}, \bar{k})$  es distinto de cero.

El teorema del pivoteo relaciona condiciones necesarias para lograr un pivoteo con el teorema de bases, dándonos un panorama más amplio de lo que se está haciendo.

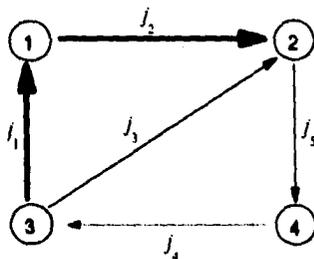
Además si se pone atención y se relacionan conceptos, se podrá notar que en cada base y después de cada pivoteo, las columnas de la tabla son los vectores de incidencia de los circuitos mencionados en el teorema de bases. (Esto se puede entender fácilmente a partir de la demostración del teorema de bases cuando llegamos a  $\sum_{k \in F'} \lambda_k a(j, k) = \sum_{k \in F'} \lambda_k e_{P_k}(j)$ ).

### Teorema del pivoteo

Sea  $F$  un bosque maximal y sean  $\bar{j}$  y  $\bar{k}$  arcos tales que  $\bar{j} \in F$  y  $\bar{k} \notin F$ , entonces, las siguientes condiciones son equivalentes:

1.  $a(\bar{j}, \bar{k}) \neq 0$  en la representación de Tucker correspondiente a  $F$ .
2.  $\bar{j}$  se encuentra en el circuito  $P_{\bar{k}}$ , correspondiente a  $\bar{k}$ , mencionado en la parte 1 del teorema de bases.
3.  $\bar{k}$  se encuentra en el corte  $Q_{\bar{j}}$ , correspondiente a  $\bar{j}$ , mencionado en la parte 2 del teorema de bases.

Un pequeño esquema muestra, a continuación lo que sucede:



$\bar{k}$	$\bar{j}$	Circuito	$e_{P_{\bar{k}}}(\bar{j})$
$J_3$	$J_2$	$3 \rightarrow 2 \leftarrow 1 \rightarrow 3$	-1

Recordando el sistema para obtener una representación inicial de Tucker (en base al teorema de bases) tenemos que, dado que  $e_{P_{\bar{k}}}(\bar{j}) = -1 = a(\bar{j}, \bar{k}) \neq 0$

### Demostración

#### 2 implica 1

$\bar{j}$  pertenece a  $F$  y a  $P_{\bar{k}}$ , por el teorema de bases

$$e_{P_{\bar{k}}}(\bar{j}) = a(\bar{j}, \bar{k})$$

dado que  $\bar{j}$  pertenece al circuito,  $e_{P_{\bar{k}}}(\bar{j})$  es distinto de cero y por ende  $a(\bar{j}, \bar{k})$  es distinto de cero en la representación de Tucker correspondiente a  $F$ .

#### 1 implica 2

Sea  $e_{P_{\bar{k}}}$  el vector de incidencia correspondiente al único circuito elemental que contiene solamente arcos pertenecientes a  $F$  a excepción de  $\bar{k}$  (el cual es recorrido positivamente).  $\bar{j} \in F$  entonces por el teorema de bases  $e_{P_{\bar{k}}}(\bar{j}) = a(\bar{j}, \bar{k})$ , pero  $a(\bar{j}, \bar{k})$  es distinto de cero, por lo tanto  $e_{P_{\bar{k}}}(\bar{j})$  es distinto de cero y por ende  $\bar{j}$  pertenece al circuito  $P_{\bar{k}}$ .

La equivalencia con la parte 3 del teorema de pivoteo se dejará para el siguiente capítulo. ©

Cuando estas condiciones se satisfacen, el conjunto  $\bar{F} = [F \setminus \bar{j}] \cup \bar{k}$  es de nuevo un bosque maximal.

A  $F$  y  $\bar{F}$  se les llama bosques adyacentes al igual que a sus respectivas representaciones de Tucker.

La manera de pasar de una tabla de Tucker a otra se verá gráficamente a continuación.

$$\begin{array}{c}
 F \\
 \begin{array}{cc}
 X(\bar{k}) & X(k) \\
 \hline
 \begin{array}{|c|c|}
 \hline
 a(\bar{j}, \bar{k}) & a(\bar{j}, k) \\
 \hline
 a(j, \bar{k}) & a(j, k) \\
 \hline
 \end{array}
 & \\
 \hline
 \end{array}
 \end{array}
 \begin{array}{l}
 = X(\bar{j}) \\
 = X(j)
 \end{array}$$

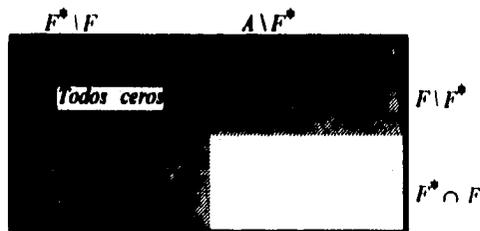
$$\begin{array}{c}
 \bar{F} \\
 \begin{array}{cc}
 X(\bar{j}) & X(k) \\
 \hline
 \begin{array}{|c|c|}
 \hline
 \frac{1}{a(\bar{j}, \bar{k})} & \frac{a(\bar{j}, k)}{a(\bar{j}, \bar{k})} \\
 \hline
 \frac{a(j, \bar{k})}{a(\bar{j}, \bar{k})} & a(j, k) - \frac{a(j, \bar{k})a(\bar{j}, k)}{a(\bar{j}, \bar{k})} \\
 \hline
 \end{array}
 & \\
 \hline
 \end{array}
 \end{array}
 \begin{array}{l}
 = X(\bar{k}) \\
 = X(j)
 \end{array}$$

Como se puede observar esta última tabla simplemente contiene los despejes que hicimos anteriormente (en las ecuaciones (1) y (2)) para obtener la nueva representación de Tucker, sin embargo es una manera mucho más sencilla de ver los resultados.

El objetivo del pivoteo es obtener la representación de Tucker correspondiente a un bosque maximal en particular, al que llamaremos  $F^*$ , a partir de una representación de Tucker correspondiente a otro bosque maximal cualquiera, al que llamaremos  $F$ . Debe quedar claro que mientras que para obtener la representación de Tucker para  $F^*$  puede ser necesario obtener varias representaciones a menos que  $F^* = \bar{F}$ .

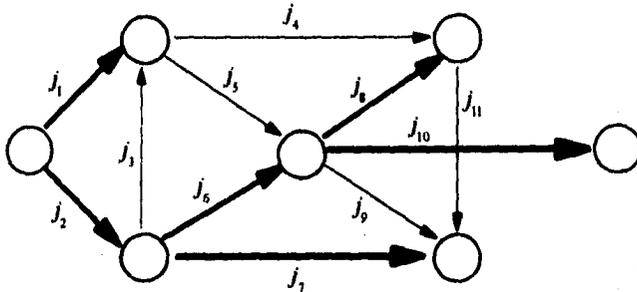
Para hacer el pivoteo comenzamos con la tabla de Tucker para  $F$  pivoteando en cualquier elemento de la forma  $(\bar{j}, \bar{k})$ , tal que  $\bar{j} \in F \setminus F^* \vee \bar{k} \in F^* \setminus F$ , y cuidando de no volver a pivotar sobre el mismo elemento hasta que ya no sea posible (es decir, los elementos columna pivoteados se convierten elementos renglón que pertenecen a  $\bar{F}$ , los cuales no queremos que salgan de la base)

Puede ser que los arcos con los que nosotros pretendemos formar una base no lo sean, ya que deben de cumplir la restricción de formar un bosque maximal. En la tabla podremos reconocer que un cierto conjunto de arcos no puede ser base al obtener una tabla como la que sigue:



En esta tabla todos los arcos que queremos meter a la base tienen un coeficiente cero. Remontándonos al teorema de bases encontramos la razón; cada columna es el vector de incidencia de un circuito formado únicamente por arcos de la base a excepción del arco que encabeza la columna que es recorrida positivamente; si pretendemos meter a la base el arco que encabeza la columna este debe de entrar en lugar de otro arco que se encuentre en el circuito del que estamos hablando, de lo contrario tendríamos un conjunto de arcos que contiene un circuito contradiciendo el que sea un bosque. Así si los coeficientes son cero los arcos que pretendemos sacar de la base no pertenecían al circuito y por lo tanto no pueden ser intercambiados por el arco que encabeza a la columna.

En el siguiente ejemplo aparece enseguida de la gráfica la representación de Tucker correspondiente al bosque maximal formado por los arcos de color más oscuro. El objetivo es llegar a la representación de Tucker correspondiente a  $F^* = \{j_1, j_2, j_4, j_7, j_9, j_{10}\}$ .



**Antes de pivotar**

(el arco sobre el que se pivotará es el sombreado en la tabla)

$X(3)$	$X(4)$	$X(5)$	$X(9)$	$X(11)$	
-1	1	1	0	0	$X(1)$
1	-1	-1	0	0	$X(2)$
0	-1	-1	1	1	$X(6)$
0	0	0	-1	-1	$X(7)$
0	-1	0	0	1	$X(8)$
0	0	0	0	0	$X(10)$

**Después de pivotar**  
 (el arco sobre el que se pivotará es el sombreado en la tabla)

X(3)	X(6)	X(5)	X(9)	X(11)	
-1	-1	0	1	1	X(1)
1	1	0	-1	-1	X(2)
0	-1	-1	1	1	X(4)
0	0	0	-1	-1	X(7)
0	1	1	[REDACTED]	0	X(8)
0	0	0	0	0	X(10)

$$\bar{F} = \{j_1, j_2, j_4, j_7, j_8, j_{10}\}$$

**Después de pivotar**

X(3)	X(6)	X(5)	X(8)	X(11)	
-1	0	1	-1	1	X(1)
1	0	-1	1	-1	X(2)
0	0	0	-1	1	X(4)
0	-1	-1	1	-1	X(7)
0	1	1	-1	0	X(9)
0	0	0	0	0	X(10)

$$F^* = \{j_1, j_2, j_4, j_7, j_9, j_{10}\}$$

## SECCION 2.11

### FLUJOS EXTREMOS

Un problema de distribución factible a menudo tiene muchas soluciones; sin embargo, todas ellas pueden ser representadas como una combinación ponderada de ciertas "soluciones extremas" más flujos elementales alrededor de circuitos especiales de capacidad ilimitada. El teorema de representación extrema de flujos expresa formalmente lo anterior, este teorema es casi una copia del teorema de representación para el caso general que se utiliza en programación lineal.

Geoméricamente, como veremos al final de esta sección, el conjunto de soluciones forma un cierto poliedro convexo en  $R^{|A|}$  y las "soluciones extremas" corresponden a los puntos extremos (vértices) de dicho poliedro. En general podrían ser utilizados muchos teoremas más de programación lineal familiares a nosotros.

Sea  $G$  una red con intervalos de capacidad  $[c^-(j), c^+(j)]$ , para cada arco  $j \in A$  y oferta  $b(i)$  para cada  $i \in N$ . Un flujo extremo  $X$ , relativo al correspondiente problema de distribución factible, es una solución factible a dicho problema, tal que el conjunto de arcos

$$F_X = \{j \in A \mid c^-(j) < X(j) < c^+(j)\}$$

forma un bosque.

En el caso de que  $b=0$ , los flujos extremos son llamados *circulaciones extremas* relativas a los intervalos de capacidad dados.

#### Proposición:

Un flujo  $X$  es extremo si y sólo si existe un conjunto  $F \subset A$  que forma un bosque maximal tal que, para todo arco  $j \in F$ , o bien  $X(j) = c^+(j)$  o bien  $X(j) = c^-(j)$ .

La demostración de lo anterior es sencilla. Supongamos que el conjunto  $F_X$  es un bosque, entonces  $F_X \subset F$  para algún bosque maximal  $F$ . Sea  $B_X = \{k \in A \mid k \notin F_X\}$  y  $B = \{k \in A \mid k \notin F\}$ . Sabemos que  $B \subset B_X$  y como se cumple que para toda  $k \in B_X$ ,  $X(k) = c^+(k)$  o  $X(k) = c^-(k)$ , entonces se cumple que, para toda  $k \in B$ ,  $X(k) = c^+(k)$  o  $X(k) = c^-(k)$ .

Hacia el otro lado, supongamos que  $F$  es un bosque maximal tal que, para cada  $k \in F^+$   $X(k) = c^+(k)$  o  $X(k) = c^-(k)$ , entonces cualquier subconjunto  $F_X$  de  $F$  tal que

$$F_X = \{j \in A \mid c^-(j) < X(j) < c^+(j)\}$$

es un bosque. (El vacío es un bosque, ya que, un bosque es una subgráfica acíclica y por vacuidad no hay ciclos) pues cualquier conjunto de arcos tal que el flujo a través de ellos no esté en los extremos del intervalo de capacidad está contenido en  $F_X$ .

Los circuitos de capacidad ilimitada mencionados en la introducción de esta sección son el equivalente a las direcciones extremas en programación lineal, este conjunto de circuitos podrá ser vacío y, cuando no lo es, se trata de un conjunto convexo de soluciones no acotado.

Un *circuito de capacidad ilimitada* es un circuito  $P$  tal que  $c^+(j) = +\infty$  para toda  $j \in P^+$  y  $c^-(j) = -\infty$  para toda  $j \in P^-$ , y un *circuito de capacidad doblemente ilimitada* es aquél con intervalo de capacidad  $(-\infty, \infty) \forall j \in P$ .

Un circuito elemental es de capacidad doblemente ilimitada si y sólo si el conjunto de arcos

$$\{j \in A \mid c^+(j) = +\infty \text{ y } c^-(j) = -\infty\}$$

no es un bosque. Cuando este conjunto está incluido en  $F_X$  para cada  $X$ , es claro que se pierde la posibilidad de que exista algún flujo extremo pues el circuito siempre estaría contenido en  $F_X$  y por lo tanto el conjunto no formaría un bosque. Como siempre, los circuitos de capacidad ilimitada juegan un papel básico en la representación extrema de flujos.

**Teorema de la representación extrema de flujos.**

Supongamos que el problema de distribución factible tiene al menos una solución y no existen circuitos de capacidad doblemente ilimitada, entonces

1. Existe un número finito de flujos extremos (al menos uno). Todos los flujos extremos son enteros si las capacidades  $c^+(j)$  y  $c^-(j)$  son enteras.
2. Un flujo  $X$  es una solución al problema de distribución factible si y sólo si ésta puede ser representada mediante la siguiente fórmula

$$X = \sum_{k=1}^r \lambda_k X_k + \sum_{i=1}^q \mu_i e_{P_i}$$

donde cada  $X_k$  es un flujo extremo,  $\lambda_k \geq 0$ ,  $\sum_{k=1}^r \lambda_k = 1$ , cada  $P_i$  es un circuito elemental de capacidad ilimitada, y  $\mu_i \geq 0$ . (El último sumando puede ser cero).

La demostración se desarrollará como sigue:

En primera instancia demostraremos finitud, posteriormente, en la sección 2.12, mediante el algoritmo para la representación extrema de flujos se demostrará la condición necesaria de la segunda parte del teorema. En la siguiente sección (sección 2.13) se presenta la justificación del algoritmo y dentro de ella se demuestra la condición suficiente que faltaba por demostrar y también que si las capacidades son enteras, los flujos extremos también lo serán, finalizando así con la demostración.

Para demostrar finitud nos valdremos de las representaciones de Tucker por lo que consideremos primero el caso en el cual se tiene  $b \cdot 0$  (circulaciones extremas) ya que si no se está en este caso, se puede llegar a él mediante la red aumentada cuyos arcos de distribución tendrán intervalos de capacidad  $[b(i), b(i)]$ ,  $i \in N$ , entonces  $F_X$  de cualquier solución extrema en esta red, no podrá contener ningún arco de distribución ya que al flujo no le quedará otra que tener un valor igual a alguna cota superior de ellos.

Para cada representación de Tucker, el conjunto  $B_k = \{k | k \in F\}$  tiene un número finito de posibles valores del flujo a través de sus arcos tales que para cada  $k \in B_k$ ,  $X(k) = c^+(j)$  o  $X(k) = c^-(j)$ , y dado que los valores  $X(j)$  son únicos cuando los valores de  $k \in B_k$  son elegidos (ya que son valores básicos) entonces los flujos extremos son finitos ya que  $F_x$  es un subconjunto del bosque maximal asociado a la representación de Tucker en cuestión. Además el número de representaciones de Tucker es finito pues está en relación uno a uno con los bosques maximales.

## SECCION 2.12

### ALGORITMO PARA LA REPRESENTACION EXTREMA DE FLUJOS

Dada una solución al problema la representaremos como en la parte dos del teorema de representaciones extremas de flujos, demostrando así tanto la existencia de al menos un flujo extremo, como la condición necesaria de la segunda parte del mismo. El algoritmo dice así :

#### Algoritmo

Sea  $X$  una solución al problema de distribución factible que no contiene circuitos de capacidad doblemente ilimitada.

1. Formar el conjunto  $F_x$  antes definido.
2. Verificar si  $F_x$  es un bosque o no.
  - Si  $F_x$  es un bosque, entonces también es un flujo extremo.
  - Si  $F_x$  no es un bosque entonces ir a 3

3. Dado que  $F_x$  no es un bosque, entonces tenemos un circuito elemental  $P$  tal que  $c^-(j) < X(j) < c^+(j)$  para todo arco  $j$  en  $P$ . Por hipótesis,  $P$  no es de capacidad doblemente ilimitada.

Calcular :

$$\mu' = \max \left\{ \mu \in \mathbb{R} \mid c^-(j) \leq X(j) - \mu e_p(j) \leq c^+(j), \forall j \in A \right\} \in (0, \infty)$$

Si  $\mu' = \infty$  calcular la función anterior pero recorriendo el circuito en sentido inverso, el circuito  $P$  es de capacidad ilimitada,  $X = X' + \mu e_p$ .

Si  $\mu' \neq \infty$  entonces  $X' = X - \mu' e_p$  es otra solución al problema de distribución factible con  $F_x$ , estrictamente contenido en  $F_x$ , y por supuesto  $X = X' + \mu e_p$ .

4. Si el circuito  $P$  no es de capacidad ilimitada entonces calcular

$$\mu'' = \max \left\{ \mu \in \mathbb{R} \mid c^-(j) \leq X(j) + \mu e_p(j) \leq c^+(j), \forall j \in A \right\} \in (0, \infty)$$

Así, tenemos que  $X'' = X + \mu'' e_p$ , también es una solución, con  $F_x$  estrictamente contenido en  $F_x$  y si  $\mu'' = \infty$   $X = X'' - \mu'' e_p$ , donde el circuito es de capacidad ilimitada.

Si  $\mu', \mu'' \neq \infty$ , ahora tenemos un sistema de dos ecuaciones

$$X' = X - \mu' e_p$$

$$X'' = X + \mu'' e_p$$

entonces,  $e_p = \frac{X - X'}{\mu'}$ , sustituyendo en la segunda ecuación llegamos a :

$$X = \frac{\mu''}{\mu' + \mu''} X' + \frac{\mu'}{\mu' + \mu''} X''$$

Dado que  $\mu'$  y  $\mu''$  son mayores que cero, entonces  $\mu'', \mu' < \mu' + \mu''$ , así

$$0 < \frac{\mu''}{\mu' + \mu''} < 1 \quad \text{y} \quad 0 < \frac{\mu'}{\mu' + \mu''} < 1$$

Sean

$$\lambda' = \frac{\mu''}{\mu' + \mu''} \quad \text{y} \quad \lambda'' = \frac{\mu'}{\mu' + \mu''}$$

entonces  $X = \lambda'X' + \lambda''X''$  con  $0 < \lambda' < 1$  y  $0 < \lambda'' < 1$

5. Seguir el algoritmo de la siguiente manera :

- Si no se encontró circuito de capacidad ilimitada, aplicar el algoritmo a  $X'$  y  $X''$
- Si se encontró circuito de capacidad ilimitada, aplicar el algoritmo al nuevo flujo ( $X'$  o  $X''$  según sea el caso).

### SECCION 2.13

#### JUSTIFICACION DEL ALGORITMO

#### EJEMPLO

Dado que en cada iteración se disuelve algún circuito del conjunto de interés, el algoritmo termina con éxito en un número finito de pasos.

Si lo que se quiere es únicamente encontrar una solución al problema de distribución factible que sea un flujo extremo, en lugar de representar la solución será suficiente con hacer todas las reducciones a partir de  $X'$  (ya no serán necesarias las de  $X''$ )

Falta demostrar que si tenemos a un flujo expresado como la parte dos del teorema entonces dicho flujo es una solución al problema de distribución factible.

Como los flujos extremos  $X_k$  por definición son factibles entonces

$$c^-(j) \leq X_k(j) \leq c^+(j) \quad \forall j, k$$

$$\lambda_k \geq 0 \Rightarrow$$

$$\begin{aligned} \lambda_1 c^-(j) &\leq \lambda_1 X_k(j) \leq \lambda_1 c^+(j) \quad \forall j, k \\ &\vdots \quad \quad \quad \vdots \\ \lambda_r c^-(j) &\leq \lambda_r X_k(j) \leq \lambda_r c^+(j) \quad \forall j, k \end{aligned}$$

Si sumamos las ecuaciones tenemos

$$c^-(j) \leq \sum_{k=1}^r \lambda_k X_k(j) \leq c^+(j) \quad \forall j, k$$

pues  $\sum_{k=1}^r \lambda_k = 1$

Además si para un arco  $j$  en particular se tiene que  $e_{P_i}(j) = 1$  donde  $P_i$  es un circuito de capacidad ilimitada, entonces  $c^+(j) = \infty$ , análogamente, si  $e_{P_i}(j) = -1$ , entonces  $c^-(j) = -\infty$ .

De aquí que, si

$$\begin{aligned} \sum_{i=1}^q \mu_i e_{P_i}(j) > 0 &\Rightarrow c^+(j) = \infty \\ \sum_{i=1}^q \mu_i e_{P_i}(j) < 0 &\Rightarrow c^-(j) = -\infty \end{aligned}$$

Como la cantidad que se le asigna a todo el circuito  $P_i$  de capacidad ilimitada es  $\mu_i$ , entonces la divergencia en cada nodo del mismo es cero, por lo tanto

$$(\text{div} X)(i) = \sum_{k=1}^r \lambda_k (\text{div} X_k)(i) = \sum_{k=1}^r \lambda_k b(i) = b(i). \otimes$$

Finalmente, si las cotas de los intervalos de capacidad son enteras, es obvio que los flujos extremos lo son pues todos los flujos extremos pueden ser encontrados aplicando el algoritmo a la solución adecuada, y los flujos extremos son obtenidos a partir de la expresión  $X' = X \pm \mu' e_p$ , donde todos los elementos son enteros. ⊗

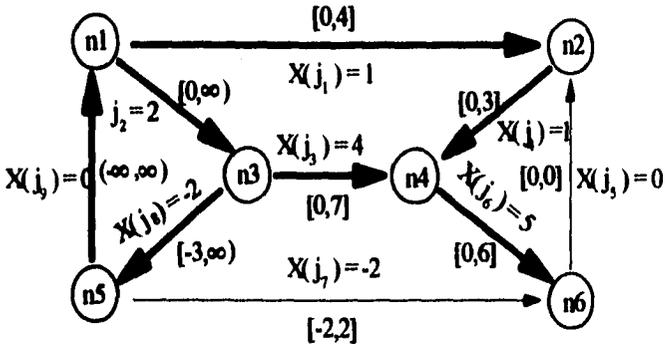
Al igual que en el caso del algoritmo de realización presentamos un diagrama de flujo del algoritmo para representación de flujos extremos con el fin de tener otra alternativa tanto para su comprensión como para tener una guía práctica.

No olvidemos que el algoritmo debe repetirse hasta que cada uno de los flujos que conforman la combinación lineal de  $X$  junto con los circuitos de capacidad ilimitada, sea un flujo extremo.



Nota: Este es el diagrama de un segmento del algoritmo, éste debe ser aplicado a todos los flujos resultantes hasta que todos sean extremos y entonces el flujo inicial pueda ser expresado como combinación lineal de flujos extremos y circuitos de capacidad limitada.

Como ejemplo, se calculará la representación extrema para el flujo y el problema de distribución mostrado en la siguiente red



$$X' = (1, 2, 4, 1, 0, 5, -2, -2, 0)$$

$$F_X = \{j_1, j_2, j_3, j_4, j_6, j_8, j_9\}$$

$F_X$  no es un bosque

Encontramos el siguiente circuito  $P$

$$P = 1 \rightarrow 2 \rightarrow 4 \leftarrow 3 \leftarrow 1$$

$$\mu' = \min\{1, 1, 3, \infty\} = 1$$

$$X'' = (0, 3, 5, 0, 0, 5, -2, -2, 0)$$

$$\mu'' = \min\{3, 2, 4, 2\}$$

$$\lambda' = \frac{2}{3}$$

$$\lambda'' = \frac{1}{3}$$

$$X^t = (3, 0, 2, 3, 0, 5, -2, -2, 0)$$

$$X = 2/3X^t + 1/3X^s$$

$E'_3 = \{j_2, j_3, j_6, j_8, j_9\}$  no es un bosque

Encontramos el siguiente circuito

$$C_2 = 1 \rightarrow 3 \rightarrow 5 \rightarrow 1$$

$$\mu^t = \min\{2, 1, \infty\} = 1$$

$$(X^t)^t = (0, 2, 5, 0, 0, 5, -2, -3, -1)$$

$\mu^s = \min\{\infty, \infty, \infty\} = \infty$  circuito de capacidad ilimitada

$$X = 2/3(X^t)^t + 2/3C_2 + 1/3X^s$$

Con esto hemos terminado toda la teoría que se verá acerca de flujos y los algoritmos que se relacionan con ella y se dará paso a los diferenciales en el siguiente capítulo.

# 3

## TERCER CAPITULO

### DIFERENCIALES

En este capítulo son relevantes los diferenciales conformables, las representaciones de Tucker para el espacio de diferenciales y los diferenciales extremos, a partir de los cuales encontramos una gran analogía con el segundo capítulo, y lo más importante, el concepto dual del problema de circulaciones dado por los diferenciales que nos permite obtener soluciones alternas sin tener que recurrir necesariamente a la programación lineal.

Las secciones 3.1 a 3.4 tratan el problema de diferenciales conformables y el teorema de realización conformable para diferenciales. A raíz de este teorema resulta un algoritmo que nos expresa a cualquier diferencial como combinación lineal de cortes elementales. De la sección 3.5 a la 3.7 se ven las representaciones de Tucker para el espacio de diferenciales y es hasta la sección 3.8 donde se habla de los diferenciales extremos y del teorema de representación extrema de diferenciales. En la sección 3.9 se muestra una tabla que presenta las analogías de los conceptos, resultados teóricos y algoritmos entre el segundo y tercer capítulo.

#### SECCION 3.1

#### DIFERENCIALES CONFORMABLES

En el caso de flujos y circulaciones la atención se centraba alrededor de las trayectorias y los circuitos, sabemos que un concepto dual a ellos es el de corte, por lo que en adelante la atención será puesta en ellos.

Con objeto de definir el concepto de diferencial conformable se requiere el de corte elemental.

Un corte  $Q$  es elemental si al remover sus arcos de la red, el número de componentes conexas se incrementa en una unidad.

El siguiente resultado será de gran utilidad más adelante:

#### Proposición

Todo corte  $Q$  puede ser expresado como la unión de cortes elementales ajenos, es decir, existen cortes elementales  $Q_1, \dots, Q_k$  los cuales no tienen arcos en común, tales que

$$Q^+ = Q_1^+ \cup \dots \cup Q_k^+ \quad \text{y} \quad Q^- = Q_1^- \cup \dots \cup Q_k^-$$

#### Demostración:

Sea  $Q$  un corte no elemental y  $S_1, \dots, S_k$  las componentes conexas resultantes de retirar los arcos de  $Q$  de la red, entonces  $Q_i = [S_i, N \setminus S_i]$  es un corte elemental para toda  $i=1, \dots, k$ ;

además  $\bigcup_{i=1}^k Q_i = Q$

Se dice que un corte elemental  $Q$  conforma a un diferencial  $V$  si:

1.  $V(j) > 0$  para todo  $j \in Q^+$  y
2.  $V(j) < 0$  para todo  $j \in Q^-$ .

Notemos que, como en el caso de flujos esta definición puede ser expresada en términos de coloraciones. En efecto, sea la coloración:

Negro : si  $V(j) > 0$

Blanco : si  $V(j) < 0$

Verde : si  $V(j) = 0$

Entonces si  $Q$  es un corte compatible con la coloración anterior, decimos que  $Q$  conforma al diferencial  $V$ .

Los diferenciales también pueden descomponerse en unidades más simples de analizar que se llaman **diferenciales elementales**, el resultado que demuestra esto quedará firmemente asentado en el teorema de realización conformable para diferenciales.

Un diferencial  $V$  es elemental, si es de la forma  $\alpha e_Q$ , donde  $Q$  es un corte elemental y  $\alpha > 0$ , que es lo mismo que :

$$V(j) = \alpha e_Q(j) = \begin{cases} \alpha & \text{si } j \in Q^+ \\ -\alpha & \text{si } j \in Q^- \\ 0 & \text{en otro caso.} \end{cases}$$

## SECCION 3.2

### TEOREMA DE REALIZACION CONFORMABLE PARA DIFERENCIALES

En esta sección se presenta el teorema de realización conformable para el caso de diferenciales que establece que cualquier diferencial puede expresarse como combinación lineal de cortes elementales de forma similar al capítulo anterior. Este resultado es muy similar al visto en la sección 2.2 para el caso de flujos.

La demostración es también constructiva proporcionándonos de esta manera el algoritmo que realiza la descomposición del diferencial y que será visto en la siguiente sección.

#### Teorema de realización conformable para diferenciales

Sea  $V \neq 0$  un diferencial cualquiera, entonces existen cortes elementales  $Q_1, \dots, Q_r$  tales que conforman a  $V$  y  $\alpha_1, \dots, \alpha_r$  mayores que cero, los cuales pueden ser escogidos enteros si  $V$  es entero, tales que

$$V = \alpha_1 e_{Q_1} + \dots + \alpha_r e_{Q_r}$$

### SECCION 3.3

#### ALGORITMO DE REALIZACION PARA DIFERENCIALES: DESCRIPCION

El algoritmo es sumamente sencillo; se parte de un diferencial factible  $V$ . El objetivo es expresar dicho diferencial como combinación lineal de los vectores de incidencia de cortes elementales que lo conformen. A partir de este momento, como en el caso de flujos, abreviaremos refiriéndonos a la combinación lineal de los vectores de incidencia de cortes elementales como "combinación lineal de cortes elementales". Se utiliza la coloración que se mencionó en la sección 3.2 y el algoritmo de Minty como herramientas principales para encontrar los cortes.

A continuación se detalla el algoritmo:

##### Paso 0

Sea  $V$  un diferencial factible, hacer

$$V_1 = V, k=1$$

##### Paso 1

Colorear de la siguiente forma

Negro si  $V(j) > 0$

Blanco si  $V(j) < 0$

Verde si  $V(j) = 0$

##### Paso 2

Buscar un arco  $j^*$  blanco o negro

- Si no hay terminar, el diferencial  $V$  ha sido expresado como combinación lineal de cortes elementales que lo conforman.

$$V = \sum_{m=1}^k \alpha_m e_{Q_m}$$

- En otro caso, aplicar el algoritmo de Minty a  $j^*$  e ir a 3

### Paso 3

- Si resulta un corte  $Q_k$ , obtener la descomposición del corte  $Q_k$  en cortes elementales ( $Q_k = Q_k^1 \cup \dots \cup Q_k^q$ ),  $i=1$

Repetir los siguientes pasos hasta que  $i=q$  (inclusive):

- $\alpha_k^i = \min |V(j)|$  para todo  $j \in Q_k^i$
- Actualizar el diferencial mediante la siguiente fórmula
- $V_{k+1} = V_k - \alpha_k^i e_{Q_k^i}$ ,  $i=i+1$

$k=k+1$ , regresar a 1.

- Si resulta circuito terminar, la función asociada a los arcos no es un diferencial, ya que  $d^+(C_k) < 0$

### SECCION 3.4

#### JUSTIFICACION DEL ALGORITMO

#### DIAGRAMA DE FLUJO DEL ALGORITMO

#### EJEMPLOS

El algoritmo de realización conformable para diferenciales es el pilar de la demostración del teorema que lo inspira. En seguida se tratarán algunos detalles que no se tratan con profundidad en el algoritmo, como por ejemplo, que los diferentes cortes encontrados conforman al diferencial, convergencia del algoritmo, etc..

Al aplicar el Algoritmo de Minty y obtener un corte este resulta ser compatible con la coloración; para cada  $j \in Q^+$  el arco es negro y por lo tanto su diferencial es positivo y si  $j \in Q^-$  el arco es blanco y por ende su diferencial negativo. Así cada corte  $Q_k$  encontrado conforma al diferencial  $V_k$ . Cuando se calcula  $\alpha_k$  se toma el mínimo del valor absoluto de los diferenciales de los arcos correspondientes al corte (por lo que si  $V$  es entero,  $\alpha_1, \dots, \alpha_r$  también) asegurando, en cada iteración, convertir al menos un arco blanco o negro en verde, por lo tanto el algoritmo termina en un número finito de pasos.

Si se encuentra un circuito  $C$ , tenemos que para cada arco recorrido positivamente (debe ser blanco o verde) el diferencial es negativo o cero y para cada arco recorrido negativamente (negro o verde) el diferencial es positivo o cero. Por otro lado,  $C$  contiene a  $j^*$  que es blanco o negro y por lo tanto el diferencial es mayor que cero o menor que cero en  $j^*$  respectivamente, así si ponemos como restricciones de generación  $d^+(j) = V(j) = d^-(j)$ , tenemos que

$$D^+(C) = \sum_{j \in P^+} d^+(j) - \sum_{j \in P^-} d^-(j) < 0$$

y por lo tanto no es un diferencial factible con respecto a los intervalos de generación (por el teorema del diferencial factible) lo que se traduce en este caso como  $V$  no es un diferencial.

Una vez modificado el diferencial, continuamos teniendo un diferencial pues es equivalente a restar  $\alpha_k$  unidades a los nodos pertenecientes a  $S$  y así garantizamos en cada iteración la construcción de un nuevo diferencial. ⊗

En esta ocasión no fue necesaria la utilización del concepto de soporte como en el caso de flujos. Sin embargo, con el objeto de mostrar una analogía más entre estos problemas consideraremos lo siguiente

El soporte de un diferencial  $V$ , es el conjunto  $S$  formado por los arcos  $j \in A$  tal que  $V(j) \neq 0$ , formado por dos subconjuntos :

$$S^+ = \{j | V(j) > 0\} \text{ y } S^- = \{j | V(j) < 0\}.$$

A partir de este último concepto y el teorema de realización conformable para diferenciales podemos obtener los siguientes resultados.

### Proposición

Un diferencial  $V$  es elemental si y sólo si el soporte de  $V$  es no vacío y minimal, en el sentido de que no existe un diferencial  $V'$  cuyo soporte es no vacío y esté estrictamente incluido en el soporte de  $V$ .

### Demostración

⇒

Supongamos que  $V$  es de la forma  $\alpha e_Q$  y existe  $V'$  cuyo soporte es no vacío y contenido en el de  $V$ , entonces por el teorema de realización conformable para diferenciales,  $V' = \alpha'_1 e_{Q'_1} + \dots + \alpha'_r e_{Q'_r}$ , donde los cortes  $Q'_i$  para  $i=1, \dots, r$  son elementales y conforman al diferencial  $V'$ . Dado que el soporte de  $V'$  está contenido estrictamente en el de  $V$ , entonces no existe ningún arco  $j$  perteneciente a algún corte tal que  $V'(j) = 0$  (pues son conformables), por lo que el conjunto de arcos pertenecientes a cualquier corte  $Q'_i$  para  $i=1, \dots, r$  está contenido en el conjunto de arcos que forman a  $Q$ ; dado que  $Q$  es un corte elemental, entonces cualquier subconjunto estricto de  $Q$  no forma un corte, o  $Q$  no era elemental, por lo tanto contradicción. Así hemos demostrado la parte necesaria del corolario.

⇐

Como  $V$  es un diferencial, entonces lo podemos expresar como  $V = \alpha_1 e_{Q_1} + \dots + \alpha_r e_{Q_r}$ , con  $\alpha > 0$  y  $Q_1, \dots, Q_r$  cortes elementales.

Sin pérdida de generalidad hagamos  $V' = \alpha_2 e_{Q_2} + \dots + \alpha_r e_{Q_r}$ ,  $V'$  también es un diferencial.

Por hipótesis no existe  $V''$  diferencial no vacío cuyo soporte esté incluido estrictamente en el de  $V'$ , por la forma en que fue obtenida la expresión de  $V'$ ,  $V'$  tiene su soporte estrictamente incluido en el de  $V$ , por lo tanto tiene que ser vacío, así  $V' = \alpha e_{Q_1}$ , con  $\alpha > 0$  y  $Q_1$  un corte elemental. ⊗

### Proposición:

Si tenemos un diferencial elemental  $V \neq 0$  y  $V'$  un diferencial cuyo soporte está incluido en el de  $V$  entonces  $V'$  es de la forma  $\lambda V$  para alguna  $\lambda$ .

### Demostración

Por el ejercicio anterior, el soporte de  $V'$  no puede estar estrictamente incluido en el de  $V$ , entonces, es igual al de  $V$ .

Como  $V$  es diferencial elemental,  $V = \alpha e_Q$  donde  $Q$  es un corte elemental y  $\alpha > 0$ . Así podemos formar en el soporte de  $V'$  el mismo corte  $Q$ .

Además  $V'$  es un diferencial elemental pues de no serlo, dicho diferencial cuyo soporte es no vacío y estar contenido en  $V'$  también lo estaría en  $V$  contradiciendo la hipótesis de que  $V$  es un diferencial elemental.

Entonces tenemos que

$$V' = \alpha' e_Q$$

$$V = \alpha e_Q$$

$\Rightarrow$

$$e_Q = V'/\alpha'$$

$$e_Q = V/\alpha$$

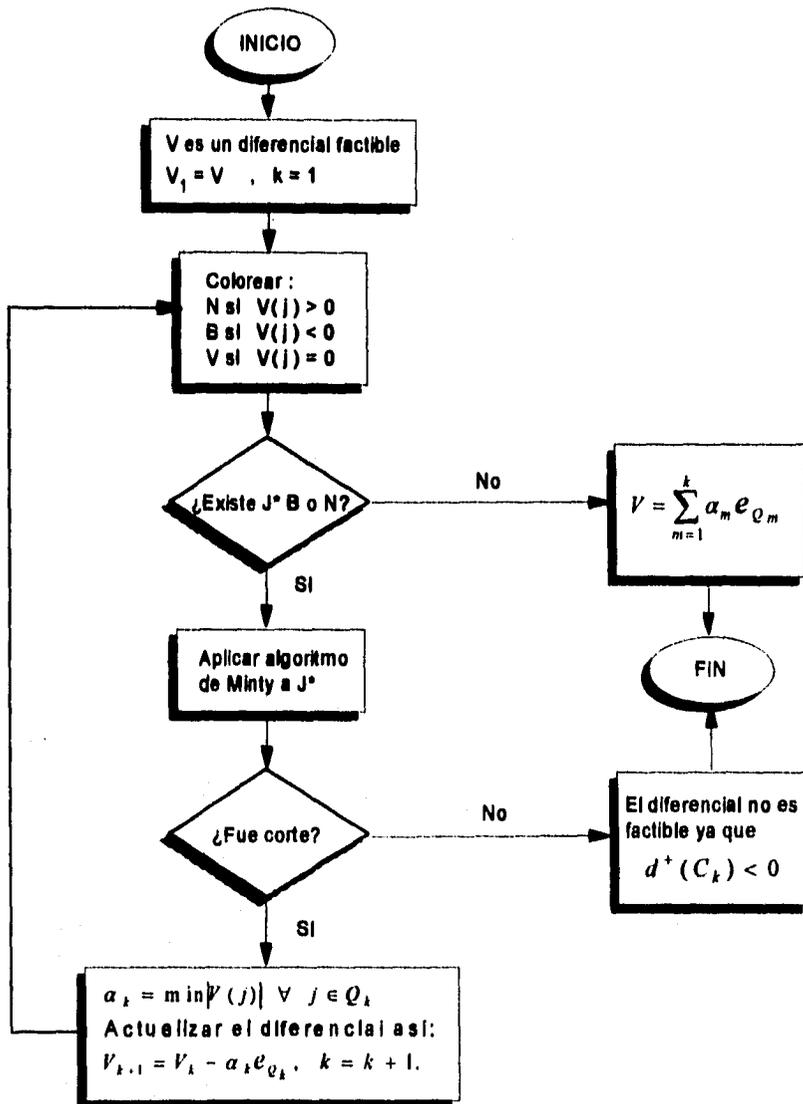
$$\text{Así } \frac{V'}{\alpha'} = \frac{V}{\alpha}, \quad V = \frac{\alpha}{\alpha'} V'$$

$$\therefore V = \lambda V' \quad \text{y } \lambda \neq 0 \text{ pues si no lo fuera}$$

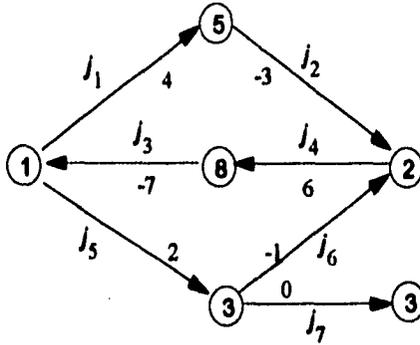
$$\alpha = 0 \text{ y } V = \alpha e_Q = \bar{0}$$

⊗

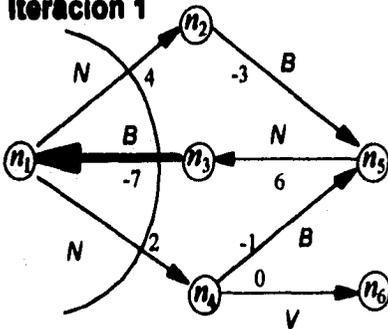
En seguida el diagrama de flujo del algoritmo de realización para diferenciales, que sirve como una referencia rápida:



Para finalizar esta sección se ejemplifica el algoritmo mediante la siguiente red. Los números asociados a los arcos corresponden al diferencial resultado de los potenciales asociados a los nodos.



**Iteración 1**

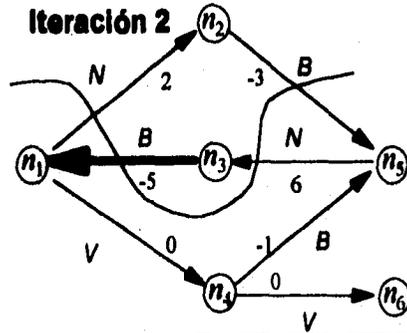


Se aplica algoritmo de Minty al arco  $j_3$

$$Q_1[\{1\} | N - \{1\}] = \{j_1, j_3, j_5\}$$

$$\alpha_1 = \min\{4, 7, 2\} = 2$$

**Iteración 2**

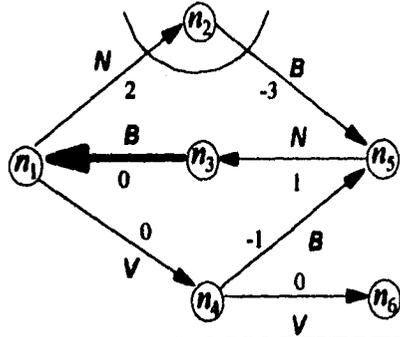
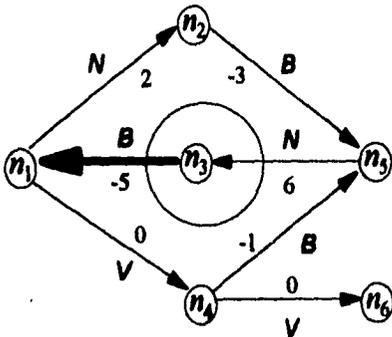


Se aplica algoritmo de Minty al arco  $j_3$

$$Q_2[\{1, 4, 5, 6\} | \{2, 3\}] = \{j_1, j_2, j_3, j_4\}$$

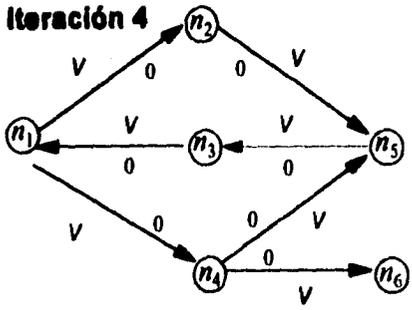
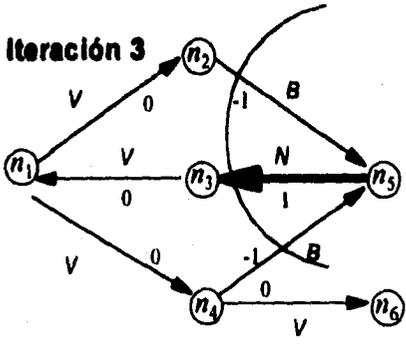
$Q_2$  no es elemental

Obtenemos la descomposición de  $Q_2$  dado que no es un corte elemental:  $Q_2 = Q_2^1 \cup Q_2^2$   
 donde  $Q_2^1 = [\{1,4,5,6,2\} | \{3\}]$  y  $Q_2^2 = [\{1,4,5,6,3\} | \{2\}]$



$Q_2^1 = [\{1,4,5,6,2\}   \{3\}] = (J_3, J_4)$
Obtenemos $\alpha_2^1$ para $Q_2^1$
$\alpha_2^1 = \min \{ 5, 6 \} = 5$

$Q_2^2 = [\{1,4,5,6,3\}   \{2\}] = (J_1, J_2)$
Obtenemos $\alpha_2^2$ para $Q_2^2$
$\alpha_2^2 = \min \{ 2, 3 \} = 2$



Se aplica algoritmo de Minty al arco $j_4$
$Q_3[\{5\}   N - \{5\}] = (J_2, J_4, J_6)$
$\alpha_3 = \min \{ 1, 1, 1 \} = 1$

<b>No existe arco blanco o negro.</b>
$V = \alpha_1 e_{Q_1} + \alpha_2^1 e_{Q_2^1} + \alpha_2^2 e_{Q_2^2} + \alpha_3 e_{Q_3}$
$\alpha_1 = 2, e_{Q_1} = (1, 0, -1, 0, 1, 0, 0)$
$\alpha_2^1 = 3, e_{Q_2^1} = (0, 0, -1, 1, 0, 0, 0)$
$\alpha_2^2 = 2, e_{Q_2^2} = (1, -1, 0, 0, 0, 0, 0)$
$\alpha_3 = 1, e_{Q_3} = (0, -1, 0, 1, 0, -1, 0)$
$V = \sum_{i=1}^3 \alpha_i e_{Q_i} = (4, -3, -7, 6, 2, -1, 0)$
<b>FIN</b>

### SECCION 3.5

#### REPRESENTACIONES DE TUCKER PARA DIFERENCIALES

(Problema dual a las representaciones de Tucker para flujos)

A través de este trabajo siempre se han relacionado cada uno de los conceptos con su dual, así las representaciones de Tucker no son la excepción, cada representación de Tucker de  $\mathcal{Q}$  puede ser inmediatamente trasladada en una representación de Tucker del espacio de diferenciales  $\mathcal{D}$ ; la analogía entre el procedimiento seguido para encontrar una representación de Tucker de  $\mathcal{Q}$  y  $\mathcal{D}$  es evidente, así como los procedimientos para cambiar de una base a otra, como resultado observaremos como podemos trabajar con ambas representaciones al mismo tiempo.

Dado que el espacio de diferenciales es el ortogonal complementario del espacio de circulaciones,

$$V \in \mathcal{D} \Leftrightarrow \\ 0 = \sum_{j \in A} V(j) X(j) \quad \forall X \in \mathcal{C}$$

Tenemos para una  $f \in A$  que corresponde a una representación de Tucker de  $\mathcal{C}$ , el flujo a través de los arcos de este conjunto puede ser expresado de la siguiente forma:

$$\sum_{k \in F'} X(k) a(j, k) = X(j) \quad \forall j \in F$$

sustituyendo en la ecuación, nos queda

$$0 = \sum_{j \in F} V(j) \left[ \sum_{k \in F'} a(j, k) X(k) \right] + \sum_{k \in F'} V(k) X(k) \\ = \sum_{k \in F'} X(k) \left[ V(k) + \sum_{j \in F} V(j) a(j, k) \right]$$

Y así tenemos que una representación de Tucker del espacio de diferenciales  $\mathcal{D}$  es un sistema de la siguiente forma:

$$-\sum_{j \in F} V(j) a(j, k) = V(k) \quad \forall k \in F'$$

Siempre es de una gran utilidad el poder resolver un problema por más de un punto de vista, además de que, en algunas ocasiones será muy claro plantear el problema para el caso de circulaciones, pero en otras, resultará más natural plantearlo como un diferencial. De aquí la importancia del resultado de dualidad.

A diferencia de las representaciones de Tucker para el espacio de circulaciones en las que todo giraba alrededor de los bosques maximales, en las representaciones de Tucker para el espacio de diferenciales, las variables básicas son las pertenecientes a un cobosque.

El concepto de cobosque, como podrá imaginar es lo opuesto a un bosque, la definición dice así:

Un **cobosque** es un conjunto de arcos  $F'$  en la red  $G$  tal que al borrar todos estos de la red, el número de componentes conexas de  $G$  no aumenta.

Un resultado equivalente es el siguiente

**Proposición**

$F'$  es un cobosque si y sólo si no existen cortes elementales en  $G$  incluidos en  $F'$  (de hecho ningún corte no vacío).

Entonces, si  $G$  es conexa los cobosques de ella serían los conjuntos  $F' \subset A$  tales que  $A \setminus F'$  contiene un árbol de expansión para  $G$ .

Las representaciones de Tucker de  $\mathcal{C}$  y  $\mathcal{D}$  siempre se dan en pares con una correspondencia uno a uno con el bosque maximal  $F$  de  $G$ .

Es interesante y puede ser de mucha utilidad observar que la tabla que define la relación entre flujos, divergencias, potenciales y diferenciales en  $G$ , puede ser también interpretada en términos de representaciones de Tucker (mediante simples operaciones por renglón), al utilizar la red aumentada. Es de notar que los arcos de distribución constituyen en particular un árbol de expansión para la nueva red. Las tablas a las que hacemos mención se presentan a continuación

$$\begin{array}{ccc}
 & X(j) \text{ para } j \in A & \\
 -U(i) \text{ para } i \in N & \boxed{e(i, j)} & = Y(i) \text{ para } i \in N \\
 & = V(j) \text{ para } j \in A & 
 \end{array}$$

$$\begin{array}{ccc}
 & X(k) \text{ para } k \in F' & \\
 -V(j) \text{ para } j \in A & \boxed{a(j, k)} & X(k) \text{ para } k \in F' \\
 & = V(k) \text{ para } k \in F' & 
 \end{array}$$

Por lo anterior, para cada representación de Tucker para el espacio de circulaciones tenemos también una para el espacio de diferenciales formada por el complemento de los arcos en la base para circulaciones, y viceversa.

A partir de aquí pondremos ambas bases en cada representación de Tucker obtenida.

### SECCION 3.6

#### TEOREMA DE BASES

La primera parte de este teorema fue demostrada en el capítulo anterior, ahora se demostrará la parte restante que es la que corresponde realmente al presente capítulo.

El teorema nos da una base para el espacio de circulaciones  $\mathcal{C}$ , la cual utiliza un conjunto de circuitos muy particulares, así como una base para el espacio de diferenciales, construida a partir de cortes. Estas bases tienen la particularidad de ser muy sencillas de encontrar y el teorema nos proporciona un algoritmo para determinar una base inicial a partir de la cual podemos utilizar el pivoteo.

Estos circuitos y cortes son elegidos en base a un bosque maximal  $F$  y ciertas restricciones; los vectores de incidencia son los elementos de las bases que quedan expresados en términos de los elementos  $a(j, k)$  de la tabla de Tucker correspondiente a  $F$ .

### Teorema de bases

Sea  $F \subset A$  el conjunto de arcos correspondiente a un bosque maximal en  $G$ , y sean  $a(j, k)$  para  $j \in F, k \in F' = A \setminus F$  los coeficientes de la representación de Tucker del espacio  $\mathcal{C}$  de circulaciones y  $\mathcal{D}$  de diferenciales

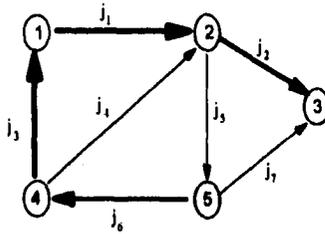
1. Para cada arco  $k \in F'$  existe un único circuito elemental  $P_k$  que contiene a  $k$  en su parte positiva y los arcos restantes del circuito pertenecen todos solamente a  $F$ . Los vectores de incidencia  $e_{P_k}$  forman una base para  $\mathcal{C}$  y están dados por

$$e_{P_k}(j) = \begin{cases} a(j, k) & \forall j \in F \\ 0 & \forall j \in F' \quad j \neq k \\ 1 & \text{para } j = k \end{cases}$$

2. Para cada arco  $j \in F$ , existe un único corte elemental  $Q_j$  que contiene a  $j$  en su parte positiva y los demás arcos del corte pertenecen únicamente a  $F'$ . Los vectores de incidencia  $e_{Q_j}$  forman una base para  $\mathcal{D}$  y están dados por

$$e_{Q_j}(k) = \begin{cases} -a(j, k) & \forall k \in F' \\ 0 & \forall k \in F \quad k \neq j \\ 1 & \text{para } k = j \end{cases}$$

Antes de proceder a demostrar la segunda parte del teorema (la primera parte se demostró en el segundo capítulo), considérese la siguiente red (la misma que en la sección 2.9) con un bosque maximal  $F$  correspondiente a los arcos de color más oscuro.



Ahora se encuentran los cortes (únicos) que cumplen con la condición de contener un arco de  $F$  en  $Q^+$  y que todos los demás arcos del corte pertenecen a  $F^+$ .

$F$	$Q = [S \mid N/S]$	
$J_1$	$S = \{1, 5, 4\}$ ,	$N/S = \{2, 3\}$
$J_2$	$S = \{1, 2, 4, 5\}$ ,	$N/S = \{3\}$
$J_3$	$S = \{4, 5\}$ ,	$N/S = \{1, 2, 3\}$
$J_6$	$S = \{5\}$ ,	$N/S = \{1, 2, 3, 4\}$

Estos cortes son los que nos proporcionarán la representación de Tucker para  $\mathcal{D}$  como veremos posteriormente.

### Demostración

#### PARTE 2

Sea  $j \in F$ ,  $j = (i, i')$ ,  $G'$  la componente conexa a la que pertenece  $j$  y  $T$  el árbol de expansión contenido en  $F$  correspondiente a  $G'$ . Si  $i$  o  $i'$  tienen grado 1, entonces el corte es de la siguiente manera  $Q = [i, N \setminus i]$  o  $Q = [i', N \setminus i']$  respectivamente, ahora bien si ninguno de los dos nodos que componen el arco en cuestión tiene grado 1, aplicar el algoritmo de enrutamiento a  $i$  y a  $i'$  en  $T - j$ , dado que es sobre  $T$  jamás llegará el enrutamiento a él pues habría un circuito. sea  $S$  el conjunto de nodos formado por el enrutamiento de  $i$  y sea  $N'$  el conjunto de nodos que componen a  $G'$ , entonces, el conjunto de nodos que compone al enrutamiento de  $i'$  es  $N' \setminus S$  ya que de lo contrario no tendríamos un árbol de expansión.

Afirmamos que si el algoritmo fuera aplicado con  $N^+ = i'$  y  $N^- = i$   $Q = [S, N' \setminus S]$  es un corte tal que para todo arco que pertenece a  $Q$  también pertenece a  $F'$  a excepción de  $j$  ya que, dada la construcción del corte, si tuvieramos al menos un arco  $j \sim (k, k')$  que pertenezca a  $T$  a parte de  $j$  en el corte tendríamos un circuito formado por arcos del bosque y esto sería una contradicción. El circuito podría formarse de la siguiente manera. Tomamos  $k$  como inicio,  $k$  perteneciente sin pérdida de generalidad a  $S$ , entonces dado el enrutamiento de  $i$  existe una trayectoria de  $k$  a  $i$ . Análogamente construimos una trayectoria que una a  $k'$  con  $i'$ , dado que  $j$  une a  $i$  con  $i'$ , entonces uniendo ambas trayectorias con  $j$  tenemos un circuito. La unicidad de este corte se da gracias a la unicidad de trayectorias dadas por un árbol de expansión

Supongamos que encontramos el corte  $e'_{a_j} = [S, N' \setminus S]$  en la componente conexa  $G'$ , entonces  $e_{a_j} = [S, N \setminus S]$  en la red original también es un corte con las características buscadas.

Asignando un potencial de 1 a cada nodo de  $N \setminus S$  y un potencial de 0 a cada nodo de  $S$  obtenemos un diferencial que es igual al vector de incidencia de  $e_{a_j}$ , así para cualquier elección de  $\lambda$

$$V = \sum_{j \in F} \lambda_j e_{a_j} \quad \forall j \in F$$

es un diferencial que satisface que  $V(j) = \lambda_j \quad \forall j \in F$  y

$$V(k) = \sum_{j \in F} \lambda_j e_{a_j}(k) \quad \forall k \in F'$$

Por la representación dual de Tucker existe un diferencial  $V$  único que satisface que  $V(j) = \lambda_j \quad \forall j \in F$ , y este es

$$V(k) = - \sum_{j \in F} \lambda_j a(j, k) \quad \forall k \in F'$$

Dado que cada elemento de  $\mathcal{D}$  puede ser representado de manera única como combinación lineal de los vectores  $e_{Q_j}$  y tenemos que

$$\sum_{j \in F'} \lambda_j e_{Q_j} = - \sum_{j \in F'} \lambda_j a(j, k) \quad \forall k \in F'$$

entonces  $e_{Q_j}$  es una base para  $\mathcal{D}$  y  $e_{Q_j} = -a(j, k)$  como  $V(j) = \lambda_j$ , entonces  $e_{Q_j}(j) = 1$  y  $e_{Q_j}(k) = 0 \quad \forall k \in F', k \neq j$ .  $\otimes$

Sabemos que la dimensión del espacio  $\mathcal{C}$  de circulaciones está determinada por la siguiente relación  $\dim \mathcal{C} = |A| - |N| + p$  (sección 2.7), como las variables básicas para el espacio de diferenciales son el complemento de las del espacio de circulaciones, la dimensión del espacio de diferenciales  $\mathcal{D}$  será  $|N| - p$ , para  $p$  componentes conexas y, en el caso particular de tratarse de una red conexa será  $|N| - 1$ .

De la misma manera que en flujos, el teorema de bases nos proporciona una forma bastante sencilla para encontrar una representación de Tucker para diferenciales con la cual iniciar, para posteriormente pivotear y encontrar la base deseada. El algoritmo es el siguiente

#### Algoritmo:

##### Paso 1

Encontrar un bosque maximal  $F'$

##### Paso 2

Para todo arco  $j$  perteneciente a  $F$ , encontrar el corte  $Q_j$ , tal que  $j$  se encuentra en  $Q_j^+$ , donde  $Q_j$  está formado únicamente por arcos de  $F'$  con excepción de  $j$ . Hallar el vector de incidencia  $e_{Q_j}$ , correspondiente a  $Q_j$ .

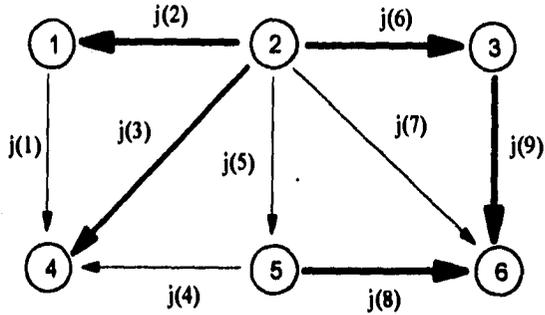
##### Paso 3

Formar la representación de Tucker para este diferencial con base en los arcos pertenecientes a  $F'$ , como sigue

Para cada renglón correspondiente a  $j$ , calcular

$$a(j, k) = -e_{Q_j}(k) \quad \forall k \in F'$$

Para ilustrar el algoritmo utilizaremos el mismo ejemplo que se vió en el tema correspondiente para flujos(sección 2.9)



$j \in F$	$Q_i^+$	$Q_i^-$	$e_{0j}$									
			$j_1$	$j_2$	$j_3$	$j_4$	$j_5$	$j_6$	$j_7$	$j_8$	$j_9$	
$j_2$	$j_2$	$j_1$	-1	1	0	0	0	0	0	0	0	0
$j_3$	$j_3, j_1, j_4$		1	0	1	1	0	0	0	0	0	0
$j_6$	$j_6, j_7, j_5$	$j_4$	0	0	0	-1	1	1	1	0	0	0
$j_8$	$j_8, j_4$	$j_5$	0	0	0	1	-1	0	0	0	1	0
$j_9$	$j_9, j_7, j_5$	$j_4$	0	0	0	-1	1	0	1	0	0	1

	$X(1)$	$X(4)$	$X(5)$	$X(7)$	
$-V(2)$	1	0	0	0	$= X(2)$
$-V(3)$	-1	-1	0	0	$= X(3)$
$-V(6)$	0	1	-1	-1	$= X(6)$
$-V(8)$	0	-1	1	0	$= X(8)$
$-V(9)$	0	1	-1	-1	$= X(9)$
	$= V(1)$	$= V(4)$	$= V(5)$	$= V(7)$	

Nota : como habíamos dicho, se obtiene exactamente la misma tabla al usar un cierto bosque maximal (caso circulaciones) que al usar su complemento como cobosque (caso diferenciales). El ejemplo es el mismo que el de la sección 2.9 para poder hacer comparaciones.

## SECCION 3.7

### PIVOTEO : CAMBIO DE UNA BASE A OTRA

Para cambiar de una base a otra, una vez que ya tenemos una base inicial, la cual podemos obtener mediante el algoritmo resultante del teorema de bases, utilizamos el pivoteo. Las representaciones de Tucker para el espacio de diferenciales corresponden a cobosques maximales. Para pasar de una base B1 a otra B2 se hace uso del pivoteo. El pivoteo es la serie de operaciones necesarias para sacar una variable de B1 (que no pertenece a B2) y meter en su lugar una variable de B2 (que no pertenecía a B1). El resultado de este pivoteo también es una base y tiene una variable más en común (con respecto a B1) con la deseada (B2). La variable que entra a la base deberá tener posibilidad de ser básica, porque, como veremos deberá cumplir que el coeficiente que tiene asociado en la representación de Tucker,  $a(\bar{j}, \bar{k})$  sea distinto de cero, donde  $\bar{j}$  es el arco que va a entrar, y  $\bar{k}$  es el arco que deseamos que salga.

Los pasos a seguir son exactamente los mismos que en el pivoteo de flujos.

Empezaremos con una tabla de Tucker para diferenciales y consideraremos cualquier par de arcos  $\bar{j} \in F$  y  $\bar{k} \in F'$  tal que  $a(j, k) \neq 0$ ,  $F$  es, en este caso, el complemento de un cobosque maximal  $F'$ . La ecuación representada por la columna  $\bar{k}$  de la tabla es la siguiente :

$$-\sum_{j \in F} a(j, \bar{k})V(j) = V(\bar{k})$$

Resolviendo para  $V(\bar{j})$

$$V(\bar{j}) = -\frac{1}{a(\bar{j}, \bar{k})}V(\bar{k}) - \sum_{\substack{j \in F \\ j \neq \bar{j}}} \frac{a(j, \bar{k})}{a(\bar{j}, \bar{k})}V(j) \quad (1)$$

stituyendo en las ecuaciones representadas por los otros renglones se obtiene

$$V(k) = \frac{a(\bar{j}, k)}{a(\bar{j}, \bar{k})} V(\bar{k}) + \sum_{\substack{j \in F \\ j \neq \bar{k}, \bar{j}}} \left[ \frac{a(j, \bar{k}) a(\bar{j}, k)}{a(j, \bar{k})} - a(j, k) \right] V(j) \text{ para cada } k \in F', k \neq \bar{k} \quad (2)$$

que es una nueva representación de Tucker del espacio de diferenciales y tiene como base a  $\bar{F}' = [F' \setminus \bar{k}] \cup \bar{j}$

A la transformación hecha en la tabla  $F'$  para obtener la tabla  $\bar{F}'$  se le llama pivoteo en  $(\bar{j}, \bar{k})$  y se dice que los bosques  $\bar{F}'$  y  $F'$  son adyacentes, así como a sus representaciones de Tucker correspondientes. Este pivoteo es posible solamente si  $a(\bar{j}, \bar{k})$  es distinto de cero.

El teorema del pivoteo relaciona al teorema de bases con las condiciones que se necesitan para que un arco que no pertenecía a la base, entre a la misma en lugar de otro arco en particular.

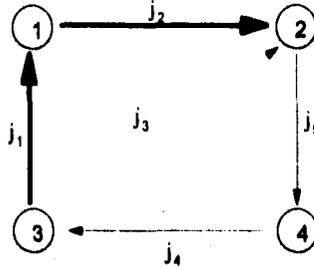
La equivalencia entre las dos primeras condiciones del teorema fue demostrada en la sección 2.10, rastando al capítulo en curso la demostración de la equivalencia de cualquiera de ellas con la tercera condición. En seguida se enuncia dicho teorema.

### Teorema del pivoteo

Sea  $F$  un bosque maximal y sean  $\bar{j}$  y  $\bar{k}$  arcos tales que  $\bar{j} \in F$  y  $\bar{k} \notin F$ , entonces, las siguientes condiciones son equivalentes:

1.  $a(\bar{j}, \bar{k}) \neq 0$  en la representación de Tucker correspondiente a  $F$ .
2.  $\bar{j}$  se encuentra en el circuito  $P_{\bar{k}}$ , correspondiente a  $\bar{k}$ , mencionado en la parte 1 del teorema de bases.
3.  $\bar{k}$  se encuentra en el corte  $Q_{\bar{j}}$ , correspondiente a  $\bar{j}$ , mencionado en la parte 2 del teorema de bases.

Un pequeño esquema en donde los arcos de color más oscuros representan a  $F$  muestra, a continuación lo que sucede:



$\bar{j}$	$\bar{k}$	Corte	$e_{Q_j}(k)$
$J_1$	$J_3, J_4$	$Q_{J_1} = \{U_1, J_3, U_4\}$	1, -1
$J_2$	$J_3, J_5$	$Q_{J_2} = \{U_2, J_3, U_5\}$	1, -1

Recordando el sistema para obtener una representación inicial de Tucker (en base al teorema de bases) tenemos que, dado que  $e_{Q_j}(\bar{k}) = -1, 1 = a(\bar{j}, \bar{k}) \neq 0$

### Demostración

$3 \Rightarrow 1$

$\bar{k} \in F^1, \bar{k} \in Q_j$  entonces por el teorema de bases  $e_{Q_j}(\bar{k}) = -a(\bar{j}, \bar{k})$ . Dado que  $\bar{k}$  pertenece al corte,  $e_{Q_j}(\bar{k}) \neq 0$ , por ende  $a(\bar{j}, \bar{k})$  es distinto de cero en la representación de Tucker para diferenciales correspondiente a  $F^1$ .  $\otimes$

La manera de pasar de una tabla de Tucker a otra se verá gráficamente a continuación.

$$\begin{array}{cc}
 & \begin{array}{c} \mathbf{F}' \\ \mathbf{X}(\bar{k}) \end{array} & \begin{array}{c} (\mathbf{F}) \\ \mathbf{X}(k) \end{array} \\
 \begin{array}{c} -\mathbf{V}(\bar{j}) \\ -\mathbf{V}(j) \end{array} & \begin{array}{|c|c|} \hline \mathbf{a}(\bar{j},k) & \mathbf{a}(\bar{j},k) \\ \hline \mathbf{a}(j,k) & \mathbf{a}(j,k) \\ \hline \end{array} & \begin{array}{c} = \mathbf{X}(\bar{j}) \\ = \mathbf{X}(j) \end{array} \\
 & \begin{array}{c} = \mathbf{V}(\bar{k}) \end{array} & \begin{array}{c} = \mathbf{V}(k) \end{array}
 \end{array}$$

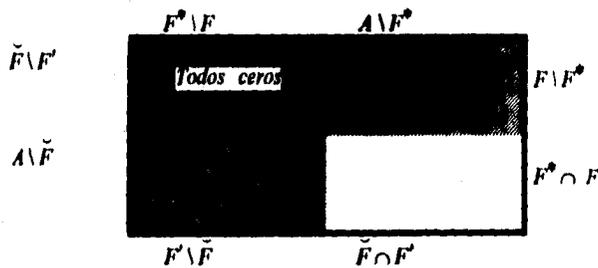
$$\begin{array}{cc}
 & \begin{array}{c} \tilde{\mathbf{F}}' \\ \mathbf{X}(\bar{j}) \end{array} & \begin{array}{c} (\tilde{\mathbf{F}}) \\ \mathbf{X}(k) \end{array} \\
 \begin{array}{c} -\mathbf{V}(\bar{k}) \\ -\mathbf{V}(j) \end{array} & \begin{array}{|c|c|} \hline \frac{1}{\mathbf{a}(\bar{j},\bar{k})} & \frac{\mathbf{a}(j,k)}{\mathbf{a}(\bar{j},\bar{k})} \\ \hline \frac{\mathbf{a}(j,k)}{\mathbf{a}(\bar{j},k)} & \mathbf{a}(j,k) - \frac{\mathbf{a}(j,\bar{k})\mathbf{a}(\bar{j},k)}{\mathbf{a}(\bar{j},k)} \\ \hline \end{array} & \begin{array}{c} = \mathbf{X}(\bar{k}) \\ = \mathbf{X}(j) \end{array} \\
 & \begin{array}{c} = \mathbf{V}(\bar{j}) \end{array} & \begin{array}{c} = \mathbf{V}(k) \end{array}
 \end{array}$$

Como se puede observar esta última tabla simplemente contiene los despejes que hicimos anteriormente (en las ecuaciones (1) y (2)) para obtener la nueva representación de Tucker, sin embargo es una manera mucho más sencilla de ver los resultados.

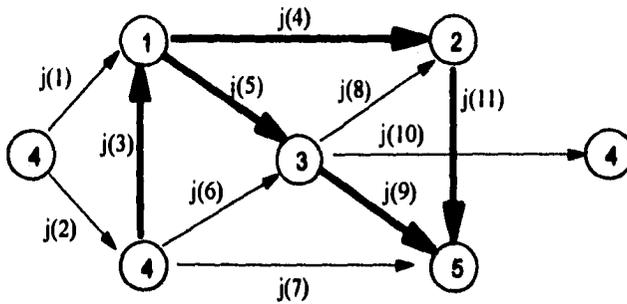
Es muy importante notar que con las mismas operaciones se obtienen, en una misma tabla, tanto la representación de Tucker para el espacio de circulaciones como la representación de Tucker para el espacio de diferenciales aprovechándose así al máximo la dualidad existente entre ambos problemas.

Así en esta tabla, para pivotar, se escoge un elemento cuyo coeficiente sea distinto de cero (dado el teorema del pivoteo) y el arco que se encuentra en el renglón correspondiente dejará de pertenecer a la base para el espacio de circulaciones para pertenecer a la del espacio de diferenciales. El arco correspondiente a la columna del elemento seleccionado dejará de pertenecer al cobosque asociado a la representación de Tucker para el espacio de diferenciales para pertenecer al bosque asociado a la representación de Tucker para el espacio de circulaciones.

Sean:  $\tilde{F}$  el cobosque correspondiente a la representación de Tucker para el espacio de diferenciales deseada ( $F^*$  el bosque correspondiente a la representación de Tucker para el espacio de circulaciones deseada),  $F'$  el cobosque asociado a la representación de Tucker para el espacio de diferenciales que se tiene actualmente ( $F$  el bosque asociado a la representación de Tucker que se tiene actualmente), y  $A$  el conjunto de arcos de la gráfica, entonces, si se llega en algún momento dado a una situación como la que sigue, ya sea se esté tratando un problema de diferenciales o circulaciones, la representación de Tucker deseada para el espacio de diferenciales (circulaciones) no es factible, es decir, no corresponde a un cobosque maximal (bosque maximal) y por lo tanto no puede ser base para dicho espacio.



En el siguiente ejemplo aparece enseguida de la gráfica la representación de Tucker correspondiente al cobosque maximal formado por los arcos de color más obscuro. El objetivo es llegar a la representación de Tucker correspondiente a  $\tilde{F} = \{j_3, j_4, j_5, j_9, j_{11}\}$  ( $F^* = \{j_1, j_2, j_4, j_7, j_9, j_{10}\}$ ).



**Antes de pivotar**  
(el arco sobre el que se pivotará es el sombreado en la tabla)

	$X(3)$	$X(4)$	$X(5)$	$X(9)$	$X(11)$	
$-V(1)$	-1	1	1	0	0	$X(1)$
$-V(2)$	1	-1	-1	0	0	$X(2)$
$-V(6)$	0	1	-1	1	1	$X(6)$
$-V(7)$	0	0	0	-1	-1	$X(7)$
$-V(8)$	0	-1	0	0	1	$X(8)$
$-V(10)$	0	0	0	0	0	$X(10)$
	$V(3)$	$V(4)$	$V(5)$	$V(9)$	$V(11)$	

**Después de pivotar**  
(el arco sobre el que se pivotará es el sombreado en la tabla)

	$X(3)$	$X(6)$	$X(5)$	$X(9)$	$X(11)$	
$-V(1)$	-1	-1	0	1	1	$X(1)$
$-V(2)$	1	1	0	-1	-1	$X(2)$
$-V(4)$	0	-1	-1	1	1	$X(4)$
$-V(7)$	0	0	0	-1	-1	$X(7)$
$-V(8)$	0	1	1	1	0	$X(8)$
$-V(10)$	0	0	0	0	0	$X(10)$
	$V(3)$	$V(6)$	$V(5)$	$V(9)$	$V(11)$	

$$\tilde{F}' = \{j_3, j_5, j_6, j_9, j_{11}\} \quad (\tilde{F} = \{j_1, j_2, j_4, j_7, j_8, j_{10}\})$$

Después de pivotar

	X(3)	X(6)	X(5)	X(8)	X(11)	
-V(1)	-1	0	1	-1	1	X(1)
-V(2)	1	0	-1	1	-1	X(2)
-V(4)	0	0	0	-1	1	X(4)
-V(7)	0	-1	-1	1	-1	X(7)
-V(9)	0	1	1	-1	0	X(9)
-V(10)	0	0	0	0	0	X(10)
	V(3)	V(6)	V(5)	V(8)	V(11)	

$$\tilde{F} = \{j_3, j_5, j_6, j_8, j_{11}\} \quad (F^* = \{j_1, j_2, j_4, j_7, j_9, j_{10}\})$$

SECCION 3.6

DIFERENCIALES EXTREMOS

Cualquier solución al problema de diferencial factible puede ser expresada como combinación lineal de ciertas "soluciones extremas" y cortes de generación ilimitada según nos dice el teorema de representación extrema para el caso de diferenciales. De nuevo encontramos analogías con la teoría de flujos, donde nos encontramos también flujos extremos y su correspondiente teorema de representación.

Un diferencial  $v$  es extremo (con respecto a los intervalos de generación) si  $d^-(j) \leq v(j) \leq d^+(j)$  para todos los arcos  $j$ , y el conjunto

$$E_v = \{j \in A \mid d^-(j) < v(j) < d^+(j)\}$$

es un cobosque.

**Proposición**

Un diferencial  $V$  es extremo si y solo si, existe un bosque maximal  $F$  tal que para todo  $j \in F$  o  $V(j) = d^+(j)$  o  $V(j) = d^-(j)$  (correspondencia biunívoca con representaciones de Tucker para el espacio de diferenciales).

Un corte de generación ilimitada es aquel que  $d^+(j) = +\infty \forall j \in Q^+$  y  $d^-(j) = -\infty \forall j \in Q^-$ . Si  $Q$  y su reverso son de generación ilimitada, entonces  $Q$  es un corte de generación doblemente ilimitada

**Proposición**

Un corte es de generación doblemente ilimitada si y sólo si el conjunto de arcos

$$F_v' = \{j \in A \mid d^+(j) = \infty \text{ y } d^-(j) = -\infty\}$$

no es un cobosque. Es claro que de existir un corte de generación doblemente ilimitada, para cualquier diferencial el conjunto  $F_v'$  no sería un cobosque.

El teorema de representación extrema de diferenciales tiene el mismo fin que el teorema de representación extrema de flujos y este es, expresar cualquier diferencial como combinación lineal de flujos extremos y cortes de generación ilimitada. Dichos flujos extremos siempre existen si existe solución al problema y su número es finito. Así pues el teorema dice

### Teorema de la representación extrema de diferenciales

Suponiendo que el problema de diferencial factible tiene al menos una solución y no existen cortes de generación doblemente ilimitada, entonces.

1. Existe un número finito de diferenciales extremos (al menos uno). Estos diferenciales son enteros, si las cotas de los intervalos de generación  $d^+(j)$ ,  $d^-(j)$  son todas enteras.
2. Un diferencial  $V$  es solución al problema de diferencial factible si y solo si, este puede ser expresado de la siguiente manera

$$V = \sum_{k=1}^r \lambda_k V_k + \sum_{l=1}^q \mu_l e_{Q_l},$$

donde  $V_k$  es un diferencial extremo,  $\lambda_k \geq 0$ ,  $\sum_{k=1}^r \lambda_k = 1$ , donde  $Q_l$  es un corte de generación ilimitada, y  $\mu_l \geq 0$  (la segunda sumatoria puede ser cero).

La demostración se desarrollará como sigue:

En primer lugar demostraremos finitud, después mediante el algoritmo para la representación extrema de diferenciales se demostrará la existencia de al menos un diferencial extremo, así como la condición necesaria de la segunda parte del teorema, todo esto en la sección 3.9. En la siguiente sección (3.10) la justificación del algoritmo y finalmente la condición suficiente además de demostrar, que si las cotas de los intervalos de generación son enteras, los diferenciales extremos también lo serán.

#### Demostración

Si  $V$  es un diferencial extremo, existe un bosque maximal  $F$  fuera de  $F'$ , entonces su representación de Tucker correspondiente es

$$V_k = - \sum_{l \in F} V(l) a(l, k) \quad \forall k \in F,$$

donde los coeficientes  $a(j, k)$  son todos 1, -1 o 0. Entonces o  $V(j) = d^+(j)$  o  $V(j) = d^-(j)$  para todo  $j \in F$ . Existen únicamente una cantidad finita de diferenciales que se pueden expresar de esta manera para una  $F$  dada, todos ellos son enteros cuando las cotas de los intervalos de generación son enteras. Además existe un número finito de bosques  $F'$ .

### SECCION 3.9

#### ALGORITMO PARA LA REPRESENTACION EXTREMA DE DIFERENCIALES

Dada una solución al problema de distribución factible el siguiente algoritmo la expresa como en la segunda parte del teorema de bases demostrando así tanto la condición necesaria de la misma, como la existencia de al menos un flujo extremo.

#### Algoritmo

##### Paso 1

Dado un diferencial factible que no contiene cortes de generación doblemente ilimitada, formar el conjunto

$$F'_v = \{j \mid d^-(j) < V(j) < d^+(j), \forall j \in A\}$$

##### Paso 2

- Si  $F'_v$  forma un cobosque, terminar,  $V$  es un diferencial extremo.
- Si  $F'_v$  no forma un cobosque ir al paso 3

##### Paso 3

Existe un corte elemental  $Q$  tal que  $d^-(j) < V(j) < d^+(j)$ , para todo  $j$  que pertenezca a  $Q$ ,  $Q$  no es un corte de generación doblemente ilimitada.

##### Paso 4

Calcular:

$$\mu' = \min_{j \in Q} \left\{ \mu \in R \mid \max \left\{ m \in R \mid d^-(j) < V(j) - m e_Q < d^+(j) \right\} \right\}$$

- Si  $\mu'$  es igual a infinito, tenemos un corte de generación ilimitada, ir al paso siguiente.
- Si  $\mu'$  es finita, tenemos que  $V' = V - \mu' e_Q$ . Entonces esta es otra solución al problema de diferencial factible, con  $F'_v$  incluido estrictamente en  $F'_v$ .

### Paso 5

Calcular:

$$\mu'' = \min_{j \in J} \left\{ \mu \in R \mid \max \left\{ m \in R \mid d^-(j) < V(j) + m e_Q < d^+(j) \right\} \right\}$$

- Si  $\mu''$  es igual a infinito y  $\mu'$  es finita, tenemos un corte de generación ilimitada, ir al paso siguiente.
- Si  $\mu''$  es igual a infinito y  $\mu'$  es infinita, FIN, no existe solución. Existen cortes de generación doblemente ilimitada.
- Si  $\mu'$  es finita, tenemos que  $V'' = V + \mu'' e_Q$ . Entonces esta es otra solución al problema de diferencial factible, con  $F''_V$  incluido estrictamente en  $F'_V$ .

### Paso 6

Si  $\mu', \mu'' \neq \infty$ , ahora tenemos un sistema de dos ecuaciones

$$V' = V - \mu' e_Q$$

$$V'' = V + \mu'' e_Q$$

entonces,  $e_Q = \frac{V - V'}{\mu'}$ , sustituyendo en la segunda ecuación llegamos a la siguiente ecuación

$$V = \frac{\mu''}{\mu' + \mu''} V' + \frac{\mu'}{\mu' + \mu''} V''$$

Dado que  $\mu'$  y  $\mu''$  son mayores que cero, entonces  $\mu'', \mu' < \mu' + \mu''$ , así

$$0 < \frac{\mu''}{\mu' + \mu''} < 1 \quad \text{y} \quad 0 < \frac{\mu'}{\mu' + \mu''} < 1$$

Sean

$$\lambda' = \frac{\mu''}{\mu' + \mu''} \quad \text{y} \quad \lambda'' = \frac{\mu'}{\mu' + \mu''}$$

entonces  $V = \lambda' V' + \lambda'' V''$  con  $0 < \lambda' < 1$  y  $0 < \lambda'' < 1$

### Paso 7

Aplicar el algoritmo a  $V'$  y  $V''$ .

## SECCION 3.10

### JUSTIFICACION DEL ALGORITMO

#### EJEMPLO

Restaría a esta justificación la demostración de convergencia del algoritmo. Dado que en cada iteración por lo menos un arco deja de pertenecer al cobosque, en un número finito de pasos, aquellos arcos que al quitarlos hacen que la red tengan una componente conexa más desaparecen de  $E'$ . Si en este conjunto se convierte en el vacío se llega también a la convergencia pues en particular éste es un cobosque.

En seguida se demuestra que si se tiene un diferencial expresado como en la segunda parte del teorema de representación para diferenciales extremos, entonces este diferencial es una solución al problema de distribución factible.

Si  $v$  tiene esta forma, entonces  $d^-(j) \leq V_k(j) \leq d^+(j)$  para  $k=1, \dots, r$  pues  $V_k$  es extremo y así

$$d^-(j) \leq \sum_{k=1}^r \lambda_k V_k(j) \leq d^+(j), \quad \forall j, k$$

$$\lambda_k \geq 0 \Rightarrow$$

$$\begin{aligned} \lambda_1 d^-(j) &\leq \lambda_1 v_k(j) \leq \lambda_1 d^+(j) \quad \forall j, k \\ \lambda_r d^-(j) &\leq \lambda_r v_k(j) \leq \lambda_r d^+(j) \quad \forall j, k \end{aligned}$$

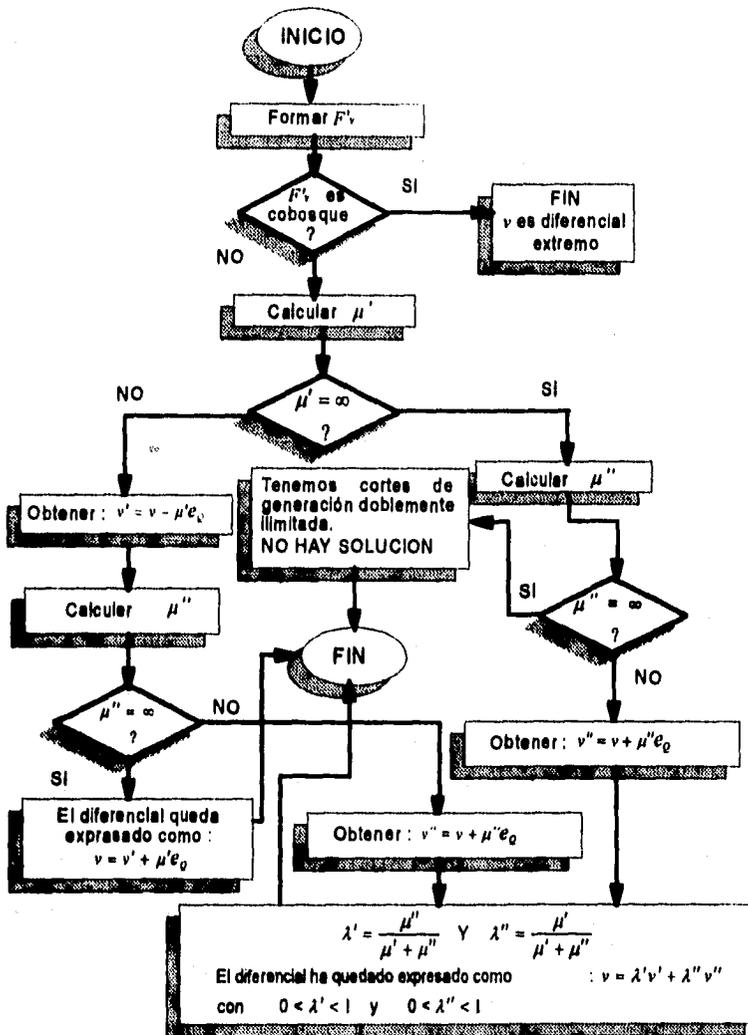
Si sumamos las ecuaciones tenemos

$$d^-(j) \leq \sum_{k=1}^r \lambda_k v_k(j) \leq d^+(j), \forall j, k$$

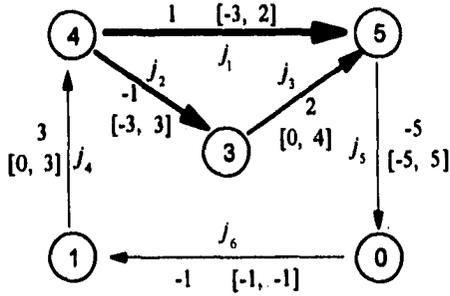
pues  $\sum_{k=1}^r \lambda_k = 1$

El término  $\mu_j e_{0,j}(j)$  puede incrementar la suma si y solo si  $d^+(j) = +\infty$ , y puede decrementarla si y solo si  $d^-(j) = -\infty$ . Entonces no puede haber violación alguna a las cotas, así que  $d^-(j) \leq v(j) \leq d^+(j)$  para todo  $j \in A$ .

El siguiente diagrama de flujo representa la parte esencial del algoritmo y debe ser aplicado a cada uno de los diferenciales resultantes hasta que cada una de las soluciones que forman parte de la combinación lineal sean diferenciales extremos.



En seguida haremos uso de este algoritmo para descomponer el diferencial factible de la siguiente red, donde las cantidades asociadas a los nodos son los potenciales y los intervalos en cada arco son los intervalos de generación



$$V = 1, -1, 2, 3, -5, -1$$

$$F\psi = j_1, j_2, j_3$$

$F\psi$  no es un cobosque, entonces existe un corte  $Q$  cuyo vector de incidencia es el siguiente:

$$e_Q = 0, -1, 1, 0, 0, 0$$

$$\mu' = \min\{4, 2\} = 2$$

$$V' = 1, 1, 0, 3, -5, -1$$

$$F\psi' = j_1, j_2$$

$F\psi'$  es un cobosque

$$\mu'' = \min\{2, 2\} = 2$$

$$V'' = 1, -3, 4, 3, -5, -1$$

$$F\psi'' = j_1$$

$F\psi''$  es un cobosque

$$\lambda' = \frac{1}{2}, \quad \lambda'' = \frac{1}{2}$$

$$V = \frac{1}{2}V' + \frac{1}{2}V''$$

### Nota

Los diferenciales extremos pueden ser identificados con los puntos extremos del poliedro convexo que consiste de todos los diferenciales factibles con respecto a generaciones.

**SECCION 3.11****ANALOGIAS**

En seguida aparece una tabla con los conceptos y resultados de mayor importancia tanto para flujos como para diferenciales. Si estos conceptos o resultados son la misma idea se pondrán uno frente al otro, asentando así las similitudes y diferencias de ambos.

<b>FLUJOS</b>	<b>DIFERENCIALES</b>
Flujo elemental	Diferencial elemental
Traectoria que conforma a un flujo $X$	Corte que conforma a un diferencial $Y$
Anticirculación	
Suma de anticirculaciones = circulación o anticirculación	
Cero, único flujo que es circulación y anticirculación	
Teorema de realización conformable para flujos	Teorema de realización conformable para diferenciales
Algoritmo de realización para flujos	Algoritmo de realización para diferenciales
Teorema de realización conformable, caso circulaciones	

FLUJOS	DIFERENCIALES
Circulación es flujo elemental $\Leftrightarrow$ su soporte es no vacío y minimal	Un diferencial $V$ es elemental si y solo si el soporte de $V$ es no vacío y minimal.
Sea $X \neq 0$ una circulación elemental y $X' \neq 0$ una circulación cuyo soporte está incluido en el de $X$ , entonces $X' = \lambda X$ para alguna $\lambda \neq 0$ .	Si tenemos un diferencial elemental $V \neq 0$ y sea $V'$ un diferencial cuyo soporte está incluido en el de $V$ entonces $V'$ es de la forma $V' = \lambda V$ para alguna $\lambda$ .
Sea $F \subset A$ , $F$ es un bosque $\Leftrightarrow$ no existe circulación $X \neq 0$ tal que su soporte esté incluido en $F$	
$F$ es bosque en $G \Leftrightarrow$ las columnas correspondientes a la matriz de incidencia de $G$ son linealmente independientes	
Teorema de expansión	
Bosque	Cobosque
Representación de Tucker para flujos	Representación de Tucker para diferenciales
$F \subset A$ corresponde a una representación de Tucker de $\mathcal{C} \Leftrightarrow F$ forma un bosque maximal para la red $G$	$F' \subset A$ corresponde a una representación de Tucker de $\mathcal{D} \Leftrightarrow F'$ forma un cobosque maximal para la red $G$
$\dim \mathcal{C} =  A  -  N  + p$ , donde $p$ es el número de componentes conexas en la gráfica	$\dim \mathcal{D} =  A  - p$ , donde $p$ es el número de componentes conexas en la gráfica

<b>FLUJOS</b>	<b>DIFERENCIALES</b>
Obtención de base inicial de Tucker para el espacio de circulaciones.	Obtención de una base inicial de Tucker para el espacio de diferenciales.
Teorema de bases	Teorema de bases
Pivoteo	Pivoteo
Teorema del pivoteo	Teorema del pivoteo
Flujo extremo	Diferencial extremo
Circuito de capacidad ilimitada	Corte de generación ilimitada
Circuito de capacidad doblemente ilimitada	Corte de generación doblemente ilimitada
Teorema de la representación extrema de flujos	Teorema de la representación extrema de diferenciales
Algoritmo para la representación extrema de flujos	Algoritmo para la representación extrema de diferenciales

# 4

## CUARTO CAPITULO

### EL PAQUETE COMPUTACIONAL

Este capítulo consta de cuatro secciones, la primera da a conocer la estructura utilizada para la representación de las redes y los aspectos por los cuales fue elegida ésta para el paquete computacional; en la segunda sección se describen la estructura general del programa y los procedimientos de mayor relevancia. La tercera sección es muy pequeña pues sólo manifiesta las principales variables y constantes utilizadas a lo largo de los programas, en la cuarta y última sección, se brinda un manual que describe las opciones y los menús para poder resolver el problema deseado; al final de la misma se darán ejemplos resueltos con el paquete con el fin de ilustrar los procedimientos y las pantallas de salida.

#### SECCION 4.1

##### REPRESENTACION DE REDES EN EL PAQUETE COMPUTACIONAL.

La primera decisión que se toma al desarrollar un programa es la estructura en la cual se representarán los datos; una forma muy utilizada para representar las redes en la computadora es mediante la matriz de adyacencia; sin embargo, tiene un gravísimo problema: ocupa mucha memoria a pesar de que es suficiente con tan sólo la mitad de los elementos de la matriz para conocer los arcos existentes en la red así como su dirección.

A pesar de que el objetivo de este programa es principalmente didáctico, se trató que la estructura utilizada fuera eficiente y tuviera más probabilidades de ser usada para fines científicos o comerciales.

Debido a éste segundo objetivo, la utilización de las matrices de incidencia para la representación de redes no es muy conveniente ya que, para redes con muchos nodos se ocupa una gran cantidad de memoria (recurso muy colizado en estos tiempos) y en caso de tener pocos arcos muchas casillas de la matriz contendrían ceros. Como siempre en los casos de computación, muchas veces la reducción en cantidad de memoria utilizada se hace a costa de la velocidad con la que se ejecuta el programa y viceversa.

Así, se decidió utilizar listas para representar las redes; cada uno de los elementos de tal lista representan a los arcos de la red y los datos que se asocian a ellos. Su representación es la siguiente

Número	Nodo Inicial	Nodo Final	Flujo	Color	M	Cota Inferior	Cota Superior	→
--------	--------------	------------	-------	-------	---	---------------	---------------	---

Donde :

**Número.** Etiqueta para el arco. Estas etiquetas no las asigna el usuario sino la máquina al leer los archivos de datos, ya sean generados en la opción del paquete para crear archivos o desde cualquier editor de texto; el usuario puede saber qué etiqueta se asigna a cada arco al ver el archivo de datos en pantalla pues la máquina asigna números naturales consecutivos a los arcos por orden de aparición.

**Nodo Inicial.** El del arco.

**Nodo final.** El del arco.

**Flujo.** El del arco.

**Color.** El que se asigna al arco de acuerdo a la coloración utilizada en los algoritmos de Realización Conformable (en este algoritmo, además de la coloración usual, toma el valor de cero (rojo) si en la siguiente casilla hay un uno ( la de M ) pues esto significa que el arco pertenece al conjunto M según la fase I del mismo) y el de Representación Extrema de Flujos.

**M.** En el algoritmo de Representación Extrema de Flujos toma el valor de uno si el arco se encuentra en este conjunto y cero en otro caso.

**Cota inferior.** Este campo es la cota inferior del intervalo de capacidad y sólo se utiliza en el segundo algoritmo; no necesita ser llenado en caso de utilizarse el primer algoritmo.

**Cota superior.** Este campo es la cota superior del intervalo de capacidad y sólo se utiliza en el segundo algoritmo; no necesita ser llenado en caso de utilizarse el primer algoritmo.

→ Apuntador que contiene la dirección del siguiente elemento de la lista.

Como se acaba de decir, en la mayoría de las ocasiones se sacrifica memoria por velocidad o viceversa. Este caso no fue la excepción, al utilizar listas para dar posibilidad de usar más arcos se pierde más tiempo en búsquedas para encontrar el registro deseado; sin embargo en la actualidad se desarrollan muchos algoritmos para optimizar búsquedas, asunto que podría mejorar la velocidad de ejecución.

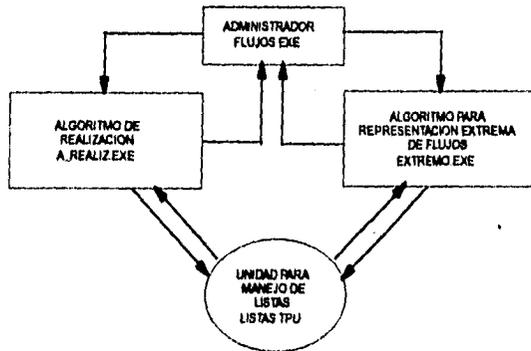
## **SECCION 4.2**

### **ESTRUCTURA GENERAL DEL PROGRAMA PROCEDIMIENTOS DE RELEVANCIA**

El paquete computacional fue desarrollado en el lenguaje de programación estructurado Turbo-Pascal en su versión 5.5 y consta de 3 programas ejecutables, una unidad que contiene la estructura de cada una de las listas utilizadas así como todas las funciones y procedimientos que se relacionan con ellas y un editor de texto de uso público cuya documentación se encuentra también dentro del conjunto de programas.

Hay tres programas ejecutables, uno por cada algoritmo (algoritmo de Realización y algoritmo para la Representación Extrema de Flujos) y un tercero para la administración de ambos y la presentación inicial.

Gráficamente los programas se relacionan de la siguiente forma :



Una vez que hemos visto la estructura general del programa describiremos ahora los principales procedimientos y funciones que se utilizan en ambos algoritmos.

#### Procedimiento ENRUTAR

(A\_realiz.pas, Extremo.pas)

Dado que la única rutina de teoría de redes que tienen en común ambos algoritmos es ésta se decidió ponerla en ambos programas con el fin de no ocupar memoria del stack, además de evitar perder más el tiempo en búsquedas innecesarias de rutinas fuera del programa. Esta rutina realiza el algoritmo de Enrutamiento y nos da como resultado dos variables booleanas (Tray, Corte) que nos indican si el resultado fue una trayectoria o un corte. De haber resultado una trayectoria ésta se encuentra en la lista llamada trayec y de haber sido un corte guarda los arcos que se encuentran en la parte positiva del corte en la lista Qm y la parte negativa en la lista Qme.

En ocasiones se llega a utilizar el vacío como conjunto inicial o final de manera que en caso de haber un corte la lista trayec contiene el enrutamiento resultante.

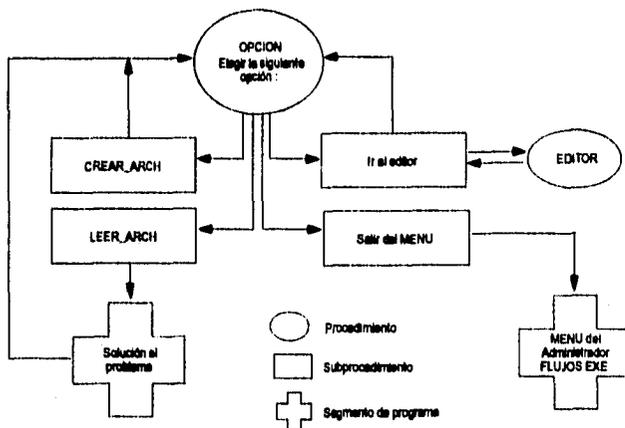
**Procedimiento OPCION**

(A\_realiz.pas, Extremo.pas)

Este procedimiento es el encargado de crear el archivo de datos (Procedimiento CREAR\_ARCH) con la información de la red y lo hace de una manera amigable. A partir de un archivo ya creado el procedimiento LEER\_ARCH se encarga de verificar la información contenida en ella con el fin de evitar la existencia de bucles, flujos fuera del intervalo de capacidad, más nodos de los permitidos, números fuera de rango, etc.

El procedimiento varía de un algoritmo a otro únicamente a causa del intervalo de capacidad que se utiliza en el algoritmo para Representación Extrema de Flujos pero no en el algoritmo de Realización.

Finalmente el Procedimiento OPCION nos da la alternativa de crear nuestro archivo de datos o modificarla desde un editor de texto.



**Procedimiento COLOREA**

(A\_realiz.pas)

Asigna la coloración para flujos conformables.

#### Procedimiento COLOREA2

(Extremo.pas)

Asigna el color rojo si el flujo se encuentra en la cota inferior o superior del intervalo de capacidad y el color verde en otro caso. Esta coloración es la que se usa en el algoritmo para Representación Extrema de Flujos.

#### Procedimiento MINTY

(A\_realiz.pas)

Este procedimiento realiza el algoritmo de Minty.

#### Procedimiento MINTY2

(Extremo.pas)

Este procedimiento realiza una versión modificada del algoritmo de Minty donde éste es aplicado a arcos verdes y no necesariamente blancos o negros como en el original.

#### Procedimiento ACTUALIZA\_M

(A\_realiz.pas)

Es propio del algoritmo de Realización; tiene como función asignar el color rojo a los arcos que pertenecen al corte encontrado indicando que pertenecen al conjunto M.

#### Procedimiento ACTUALIZAFLUJO

(A\_realiz.pas)

Procedimiento del algoritmo de Realización que actualiza el flujo una vez encontrada una circulación o anticirculación mediante el valor de alfa( $\alpha$ ) a través de la función MINIMO.

**Procedimientos FASE\_UNO y FASE\_DOS**  
(A\_realiz.pas)

Procedimientos principales del algoritmo de Realización que realizan la primera y segunda fases respectivamente.

**Procedimiento ENCUESTRA\_CIRCUITO**  
(Extremo.pas)

Procedimiento de gran importancia para el algoritmo de Representación Extrema de Flujos que tiene por finalidad detectar un arco verde que pertenezca a un circuito formado por arcos verdes, si es que existe dicho circuito en el flujo que se está analizando. La existencia de este circuito es detectada mediante la variable booleana circuito. Si este circuito no existe, entonces se trata de un flujo extremo.

**Procedimiento MAXIMO**  
(Extremo.pas)

Este procedimiento es muy parecido a la función MINIMO del algoritmo de Realización, calcula las  $\mu$ 's con las cuales se obtendrán los nuevos flujos más cercanos a ser flujos extremos durante el algoritmo de Representación.

**Procedimiento FORMA\_2HIJOS**  
(Extremo.pas)

Es el que se encarga de formar los dos nuevos flujos más cercanos a ser extremos a partir de uno que no lo es; en caso de tener un circuito de capacidad ilimitada crea un flujo más cercano a ser extremo y guarda el circuito de capacidad ilimitada.

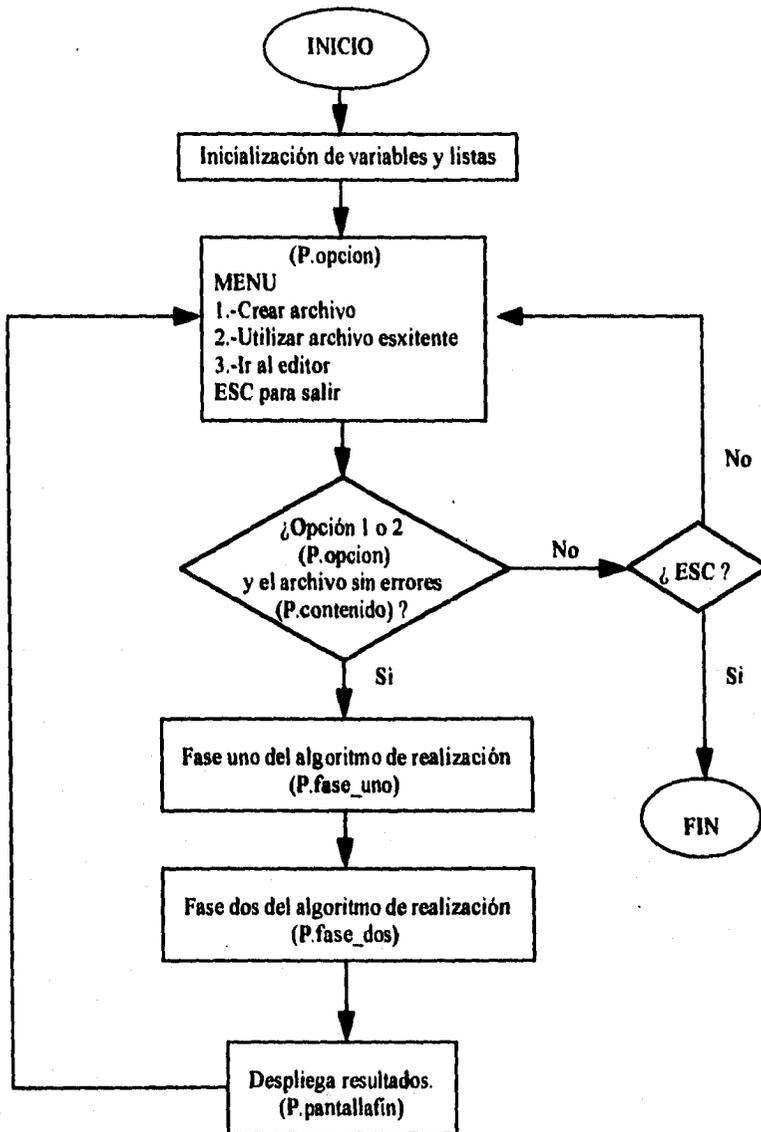
**Procedimientos para escritura de resultados :**

- Procedimiento **ESCRIBETRAYEC** : algoritmo de Realización.  
Despliega la combinación lineal de trayectorias elementales
- Procedimiento **PANTALLAFIN** : algoritmo de Realización.  
Despliega el mensaje final
- Procedimiento **MUESTRA** : algoritmo para Representación Extrema de Flujos.  
Despliega los flujos extremos en pantalla y el mensaje final.
- Procedimiento **MUESTRA2** : algoritmo para Representación Extrema de Flujos.  
Despliega los circuitos de capacidad ilimitada y el mensaje final.

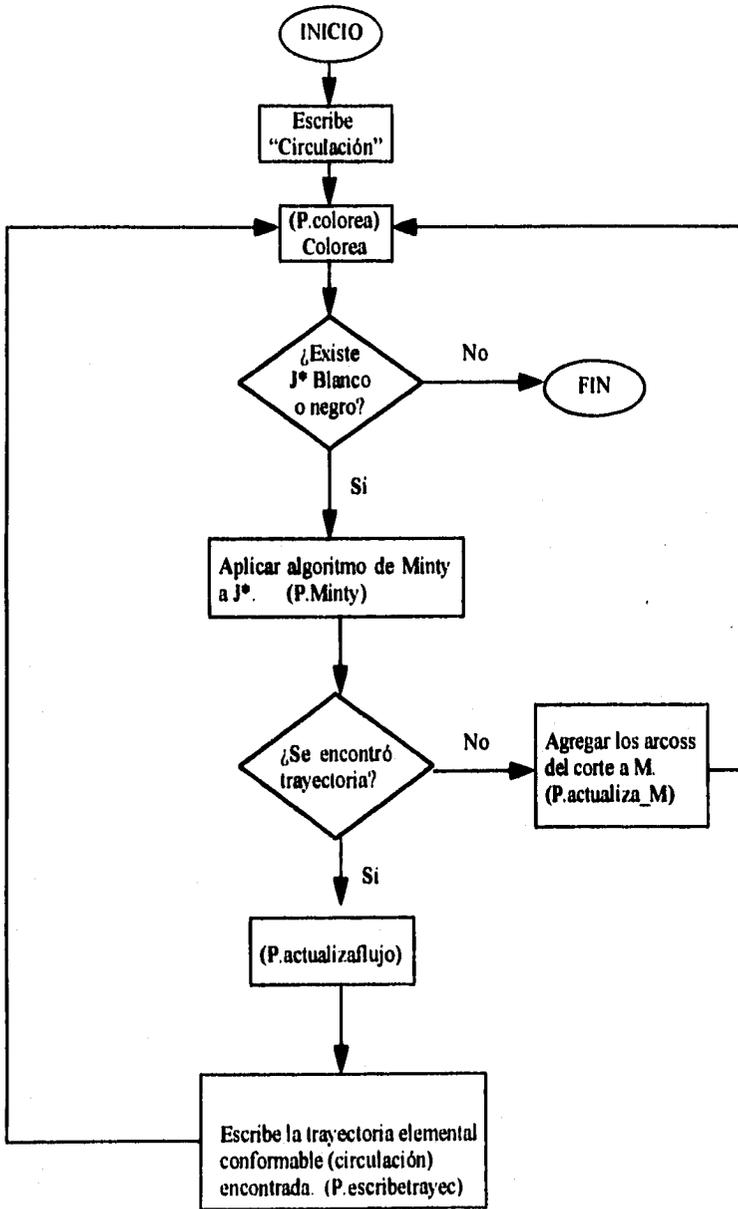
El cuerpo principal del programa que realiza el algoritmo de Realización es muy sencillo y se presenta a continuación :

```
Begin (Programa Principal A_realiz.pas)
  Lineas:=0;
  trayectoria:=nil;
  Repeat
    opcion;
    fase_uno;
    fase_dos;
    pantallaFin;
    destruye_lista(x);
  Until key='s';
End. (Programa principal A_realiz.pas)
```

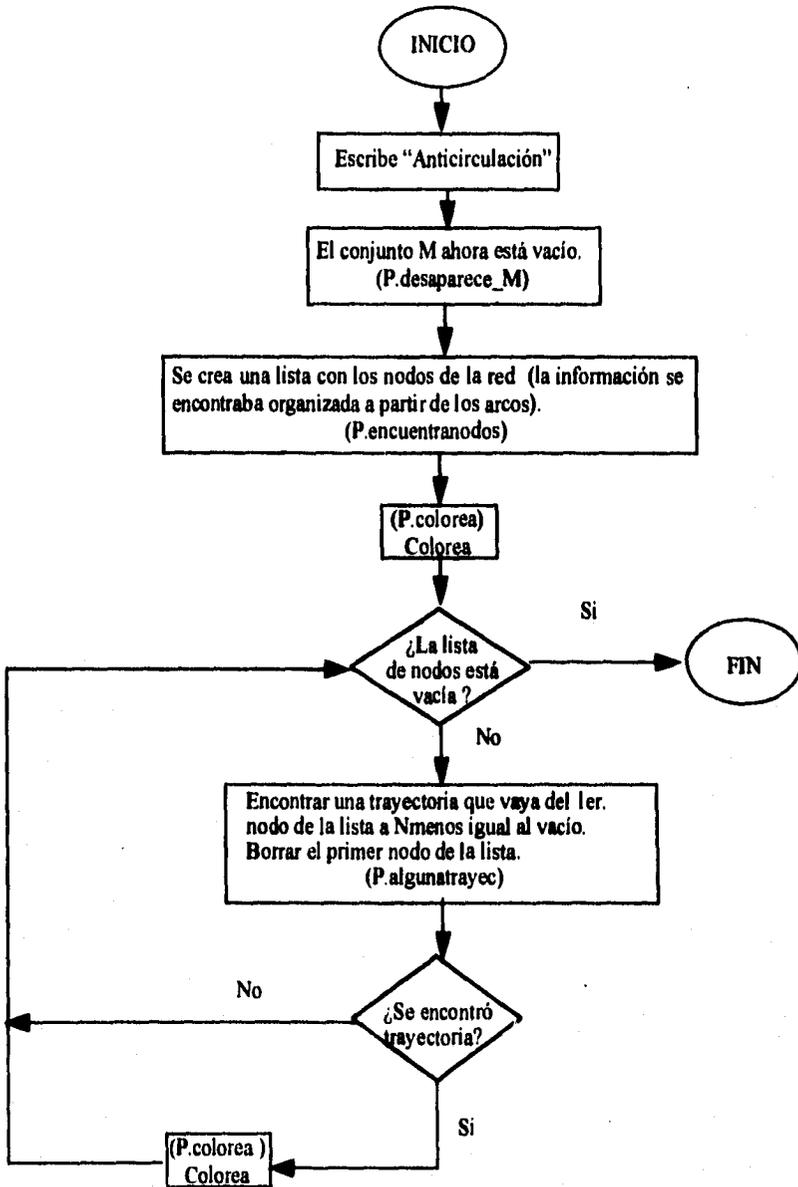
# Cuerpo del programa para el ALGORITMO DE REALIZACION



# PROCEDURE FASE\_UNO



## PROCEDURE FASE\_DOS



El código del algoritmo para Representación Extrema de Flujos se muestra a continuación junto con el diagrama de flujo :

```
Begin (Program Principal Extremo.pas)
  Repeat
    opcion;
    inicializa;
    While cola1<>nil do
      Begin
        colorea2;
        encuentra_circuito;(circuit_arco);
        If not(circuit) then
          {Si no hubo circuito tenemos un flujo extremo}
          Begin
            Fextrem:=const_extre(cola1,Fextre);
            ayuda:=cola1^.sig;
            dispose(cola1);
            cola1:=ayuda;
          End
        Else
          Begin
            Minty2;
            {Una vez encontrado el circuito procede a calcular las  $\mu$ 's y a
            encontrar dos flujos a partir del dado que están más cerca de
            ser extremos}
            maximo;
            forma_2hljos;
          End;
        End;
      End;
    muestra;
    muestra2;
    destruye_lista(x);
  Until nunca;
End. (Programa Principal Extremo.pas)
```

# Cuerpo del programa que realiza el ALGORITMO PARA LA REPRESENTACION EXTREMA DE FLUJOS

**Variables:**

Cola1, Fextrem: listas conformadas de la manera siguiente:

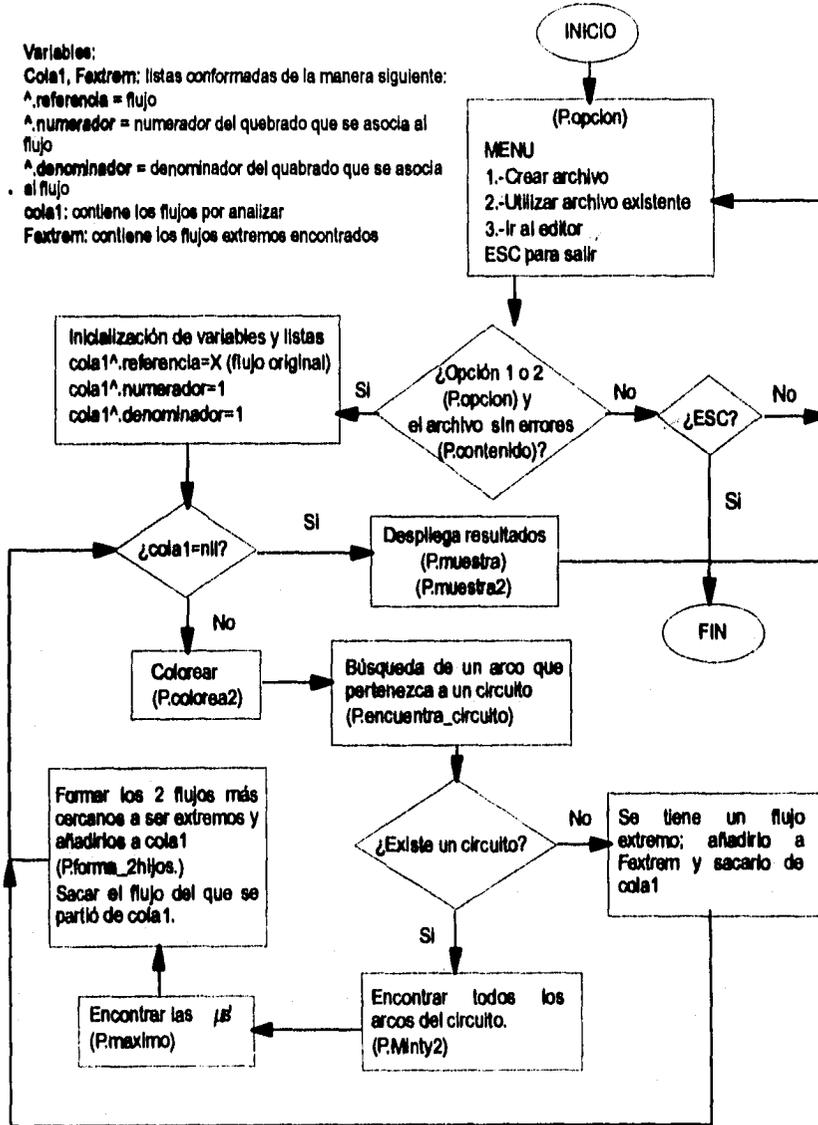
^referencia = flujo

^numerador = numerador del quebrado que se asocia al flujo

^denominador = denominador del quebrado que se asocia al flujo

cola1: contiene los flujos por analizar

Fextrem: contiene los flujos extremos encontrados



### SECCION 4.3

#### PRINCIPALES CONSTANTES

#### A LO LARGO DEL PAQUETE DE COMPUTO

Las constantes en su mayoría utilizan los valores extremos permitidos por el lenguaje utilizado para el tipo de variable correspondiente; definir las como constantes permite principalmente hacerlas más localizables a la persona que observa el código del programa para modificarlas fácilmente de así desearse y hacer más sencillas las rutinas de validación de datos. Las constantes son las siguientes :

**maxarcos = 32767** Número máximo de arcos debido a que se miden con variables enteras cuyo máximo en la máquina es el dado.

**maxvalnodo = 255** Número máximo de nodos ya que se guardan en variables de tipo conjunto. El máximo número de elementos aceptado en pascal para la versión 5.5 es de 255 ( Esto puede cambiar con otra versión )

**minvalnodo = 0** Pues se estableció el criterio de dar etiquetas de números naturales en adición con el cero.

**maxflujo = 9.0E36** El flujo y las cotas del intervalo de capacidad son variables de tipo real y éste es su máximo valor.

**minflujo = -9.0E36** Mínimo valor para variables de tipo real como lo son el flujo y las cotas inferior y superior de los intervalos de capacidad.

Estas constantes aparecen al principio del programa para que sean fáciles de localizar y modificar.

## **SECCION 4.4**

### **MANUAL DEL USUARIID**

En esta sección se presenta una guía para desenvolverse sin problemas a lo largo de él. Para instalar el programa basta con copiar todos los archivos del disquete en un mismo directorio del sistema operativo.

Este paquete de cómputo aplica a las redes que el usuario proporcione los siguientes algoritmos :

- Algoritmo de Realización
- Algoritmo para la Representación Extrema de Flujos

#### **SECCION 4.4.1**

##### **COMO COMENZAR**

##### **UN VIAJE A TRAVES DE LOS MENUS**

¿Cómo entrar al programa?, simplemente teclee **FLUJOS** desde el directorio donde tiene instalados los archivos de este paquete y aparecerá la pantalla de presentación que menciona los algoritmos que resuelve y el nombre del programador, oprima enter para seguir adelante como indica en la parte inferior de la pantalla y se encontrará frente al siguiente menú :

**MENU PRINCIPAL**

- 1.- Algoritmo de Realización
- 2.- Representación Extrema de Flujos

**ESC : salir**

Cualquiera que sea la opción que se escoja el paquete lo llevará a un menú similar al siguiente donde, en caso de haber escogido el segundo algoritmo, aparecerá el encabezado correspondiente en lugar del encabezado ALGORITMO DE REALIZACION.

**ALGORITMO DE REALIZACION**

**MENU**

- 1.- Crear archivo
- 2.- Utilizar archivo existente
- 3.- Ir al editor

**ESC : salir**

La primera opción lo irá guiando para crear al archivo de datos que representa a la red, le preguntará el número de arcos que forma a la red y posteriormente las características del arco. Al terminar de formar su archivo de datos volverá al mismo menú dándole la opción de crear nuevas archivos de datos o aplicar el algoritmo a algún archivo ya creado.

La segunda opción se utiliza cuando el archivo de datos que representa a la red ya ha sido creado, le preguntará la trayectoria en la que se encuentra el archivo y procederá a leerlo para la validación de los datos; si los datos están correctos procederá inmediatamente a aplicar el algoritmo previamente elegido, de lo contrario le hará saber al menos algún error que se esté cometiendo en el archivo. Una vez que los resultados hayan aparecido en pantalla verá un mensaje de término del algoritmo y al oprimir ENTER volverá al menú del que proviene. El programa no da opción a grabar el resultado en disco.

La tercera y última opción permite modificar archivos de datos o crear alguno sin necesidad de ser guiado por la opción de crear archivos utilizando un editor de texto. Consulte la sección referente a la construcción de archivos de datos. Como siempre, al salir del editor regresará al mismo menú.

Si se desea ejecutar el otro algoritmo, lo único que tiene que hacer es oprimir la tecla ESC. Para salir del programa oprima de nuevo ESC como dice al pie de la pantalla y regresará al sistema operativo.

#### **SECCION 4.4.2**

#### **SOBRE LOS ARCHIVOS DE DATOS QUE REPRESENTAN A LAS REDES**

Cada renglón en el archivo representa a un arco, de manera que habrá tantos renglones como arcos en la red; cada dato deberá de separarse de los demás por al menos un espacio y sin ningún otro carácter de separación como lo son las comas o tabulares y el final de cada línea deberá usarse la tecla de fin de línea ENTER (↵). Siguiendo este esquema habrá dos tipos de redes según el algoritmo que se quiera utilizar; aunque si usted desea, el esquema del archivo de datos que resuelve el algoritmo para Representación Extrema de Flujos le servirá también para el algoritmo de Realización. Es decir que puede utilizar el mismo archivo de datos que utilizó con el algoritmo de Representación Extrema para resolver el de Realización; sin embargo, no podrá usar uno que fue diseñado originalmente para el algoritmo de Realización pues no se encontrarán los intervalos de capacidad necesarios para la resolución del problema de Representación Extrema de Flujos.

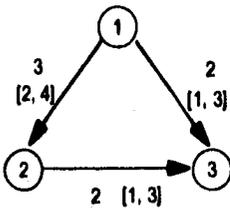
**Cómo formar un archivo de datos para redes que serán usadas en el algoritmo para Representaciones Extremas de Flujos**

Todos los registros del archivo siguen la misma estructura; en el diagrama siguiente se especifica qué datos deben estar contenidos, en qué orden y el intervalo de valores posibles de cada tipo de datos. Utilice ENTER (↵) al final del renglón.

Editor de Texto :

Nodo Inicial	Nodo Final	Flujo	Cota Inferior	Cota Superior	↓
{0,255}	{0,255}	{-9.0E36, 9.0E36}	{-9.0E36, 9.0E36}	{-9.0E36, 9.0E36}	
		Más Infinito : I+	Más Infinito : I+	Más Infinito : I+	
		Menos Infinito : I-	Menos Infinito : I-	Menos Infinito : I-	

Un ejemplo de un archivo que corresponde a la figura de la izquierda sería el siguiente :



**ARCHIVO**

```

1 2 3 2 4
2 3 2 1 3
1 3 2 1 3
  
```

**Sobre los datos de la red :**

Cuando usted cree un archivo de datos tenga cuidado de los rangos que están permitidos.

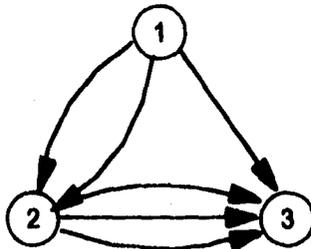
En caso de tener un flujo infinito (menos infinito) o alguna cota del intervalo de capacidad también infinita (menos infinito) bastará con teclear I+ (I-) en lugar del flujo o la cota del intervalo.

Los nodos de la red deberán ser números enteros positivos e inclusive podrá utilizar el cero.

En el caso de la numeración de los nodos, no será posible asignar un número mayor a 255, de manera que el número total de nodos no podrá exceder de 256 (porque incluye al cero).

En este paquete computacional no están permitidos los bucles, es decir, arcos cuyo nodo inicial y final coincidan (de la forma  $(i, i)$ ).

Sin embargo en el caso de multiplicidades no tendrá ningún problema, esto quiere decir que usted puede ingresar varios arcos con el mismo extremo inicial y el mismo extremo final, como por ejemplo la siguiente red:

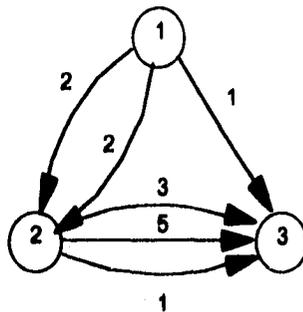


En este ejemplo encontramos 2 arcos de la forma  $(1, 2)$ , 3 arcos de la forma  $(2, 3)$  y uno de la forma  $(1, 3)$ , pero en el paquete computacional se trabaja con las etiquetas asignadas a los arcos de manera que, a pesar de haber más de un arco con la misma representación  $(i, j)$ , éstos serán diferentes al aplicar los algoritmos respetando así la estructura original de la red. Además usted podrá identificar en la pantalla de salida de qué arco se trata pues aparece del lado izquierdo de cada arco la etiqueta asignada por el paquete al arco en cuestión que simplemente se asigna por orden de aparición.

**Cómo formar un archivo de datos para redes que serán usadas en el algoritmo de Realización**

La única diferencia con el archivo anterior es que no se toman en cuenta los últimos dos datos de cada renglón; como ya habíamos dicho, los archivos utilizados en el algoritmo para Representaciones Extremas de Flujos pueden utilizarse para el algoritmo de Realización pero los del algoritmo de Realización no podrán utilizarse para el algoritmo para Representaciones Extremas de Flujos.

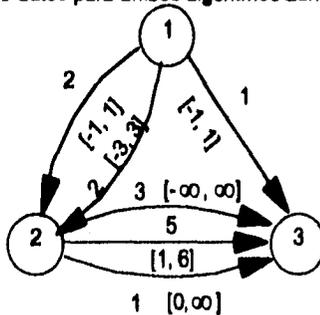
Si usted sólo va a aplicar el algoritmo de Realización bastará con poner en cada renglón el nodo inicial, nodo final y el flujo del arco. De cualquier modo puede consultar el siguiente ejemplo con su correspondiente archivo por si quedó con alguna duda :



**ARCHIVO**

1	2	2
1	2	2
1	3	1
2	3	3
2	3	5
2	3	1

Este ejemplo es una réplica de la red que utilizamos para explicar el caso de multiplicidades, se utilizará ahora junto con los intervalos de capacidad con el fin de dejar clara la estructura de los archivos de datos para ambos algoritmos aún cuando las redes tienen multiplicidades.



**ARCHIVO**

1	2	2	-1	1
1	2	2	-3	3
1	3	1	-1	1
2	3	3	-	+
2	3	5	1	6
2	3	1	0	+

### SECCION 4.4.3

#### RESULTADOS

#### EJEMPLOS

#### ALGORITMO DE REALIZACION

El algoritmo tiene como resultado la expresión de un flujo como la suma de una circulación y una anticirculación a su vez expresadas como combinaciones lineales de trayectorias elementales; así pues al desplegar los resultados se desplegarán en dos bloques: el primero el de la circulación y el segundo el de la anticirculación. Ambos bloques llevan la misma estructura.

Todo arco  $(n_i, n_f)$ , donde  $n_i$  es el nodo inicial y  $n_f$  el nodo final, lleva su correspondiente etiqueta  $J_i$  que es asignada por orden de aparición del lado izquierdo. la constante que multiplica al vector de incidencia de la trayectoria siempre aparecerá del lado izquierdo. Sólo las entradas distintas de cero del vector de incidencia son desplegadas en pantalla. El siguiente diagrama es una copia de la pantalla de resultados del algoritmo de Realización:

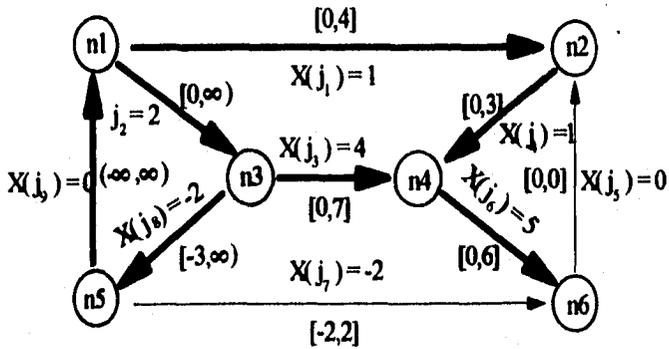
CIRCULACION				
	$J_a(n_i, n_f)$	$J_b(n_i, n_f)$	$J_c(n_i, n_f)$	$J_d(n_i, n_f)$
constante A (	$e_p(J_a)$	$e_p(J_b)$	$e_p(J_c)$	$e_p(J_d) +$
	.	.	.	.
	.	.	.	.
	.	.	.	.
ANTICIRCULACION				
	$J_a(n_i, n_f)$	$J_b(n_i, n_f)$	$J_c(n_i, n_f)$	$J_d(n_i, n_f)$
constante B (	$e_p(J_a)$	$e_p(J_b)$	$e_p(J_c)$	$e_p(J_d) +$
	.	.	.	.
	.	.	.	.
	.	.	.	.

ESC : Salir

EL FLUJO HA QUEDADO EXPRESADO COMO LA SUMA DE UNA CIRCULACION Y UNA ANTICIRCULACION.  
 ASI TAMBIEN, EL FLUJO HA QUEDADO EXPRESADO COMO COMBINACION LINEAL DE TRAYECTORIAS ELEMENTALES

ESC : Salir

En la siguiente red se aplicará el algoritmo de Realización con el paquete. Esta red es la misma que se utilizará para el caso del algoritmo para Representación Extrema de Flujos. Observe que el mismo archivo de datos es utilizado por ambos algoritmos. El despliegue de resultados se observa en seguida.



El archivo de datos correspondiente a la red, es el siguiente:

**ARCHIVO DE DATOS**

12104  
13201+  
24103  
34407  
35-2-31+  
46506  
5101-1+  
62000  
56-2-22

La pantalla final que despliega los resultados del algoritmo para Representación Extrema de Flujos es similar a la que se presenta a continuación

<b>CIRCULACION</b>				
	5(3,5)	4(3,4)	6(4,6)	8(5,6)
2.00 (	-1	1	1	-1 ) +
<b>ANTICIRCULACION</b>				
	4(3,4)	6(4,6)		
2.00 (	1	1 ) +		
	6(4,6)			
1.00 (	1 ) +			
	1(1,2)	2(1,3)	3(2,4)	
1.00 (	1	1	1 ) +	
	2(1,3)			
1.00 (	1 )			

ESC: Salir

**EL FLUJO HA QUEDADO EXPRESADO COMO LA SUMA DE  
UNA CIRCULACION Y UNA ANTICIRCULACION.  
ASI TAMBIEN, EL FLUJO HA QUEDADO EXPRESADO COMO  
COMBINACION LINEAL DE TRAYECTORIAS ELEMENTALES**

**ESC : Salir**

**Finalmente usted regresará al menú principal del algoritmo; si usted quiere aplicar ahora el otro algoritmo oprima ESC y regresará al menú principal del programa. Para salir, oprima de nuevo la tecla ESC.**

**ALGORITMO DE REALIZACION**

**MENU**

- 1.- Crear archivo**
- 2.- Utilizar archivo existente**
- 3.- Ir al editor**

**ESC : salir**

**MENU PRINCIPAL**

- 1.- Algoritmo de Realización**
- 2.- Representación Extrema de Fiujos**

**ESC : salir**

ALGORITMO PARA LA REPRESENTACION EXTREMA DE FLUJOS

En el caso del algoritmo para la Representación Extrema de Flujos una vez que se ha seleccionado el archivo de datos a la cual se aplicará el algoritmo, el paquete presenta en primera instancia a los flujos extremos (reconozca la sección por la aparición de la variable lambda), si es que la solución encontrada tiene alguno, y en segunda instancia a los intervalos de capacidad ilimitada. El diagrama siguiente describe la presentación de los flujos extremos:

X=		$j_m ( N_i, N_f )$	$j_{m-1} ( N_i, N_f )$	$j_{m-2} ( N_i, N_f )$
LAMBDA		[ $C_i, C_f$ ]	[ $C_i, C_f$ ]	[ $C_i, C_f$ ]
lambda	[	flujo( $j_m$ )	flujo( $j_{m-1}$ )	flujo( $j_{m-2}$ )
		.	.	.
		.	.	.
		.	.	.
		$j_3 ( N_i, N_f )$	$j_2 ( N_i, N_f )$	$j_1 ( N_i, N_f )$
		[ $C_i, C_f$ ]	[ $C_i, C_f$ ]	[ $C_i, C_f$ ]
		flujo( $j_3$ )	flujo( $j_2$ )	flujo( $j_1$ ) ]+

Presione ENTER para continuar . . .

Donde LAMBDA es un escalar multiplicado por el flujo extremo;  $j_k$  es la etiqueta correspondiente al orden de aparición del arco en el archivo de datos;  $j_k ( N_i, N_f )$  son nodo inicial y final del arco  $j_k$  y [  $C_i, C_f$  ] es el intervalo de capacidad asociado al mismo. El flujo asociado a  $j_k$  se encuentra en la línea inmediata inferior al intervalo de capacidad.

Los arcos son dispuestos en orden inverso al archivo de datos y todos son presentados.

El diagrama siguiente es similar al anterior y describe la presentación de los intervalos de capacidad, si es que la solución encontrada contiene alguno.

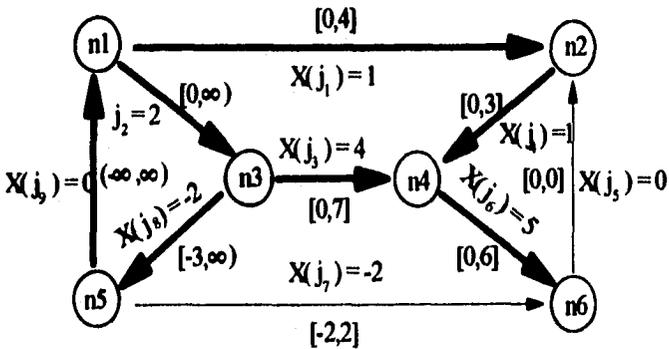
	$j_i ( n_i, n_f )$		$j_p ( n_i, n_f )$
MU	$[ C_i, C_f ]$	...	$[ C_i, C_f ]$
mu	$[ \text{flujo}(j_m),$		$\text{flujo}(j_{m+2}) ] +$
	.	.	.
	.	.	.

Presione ENTER para continuar . . .

Donde MU es la constante que multiplica al vector de incidencia del circuito de capacidad ilimitada. En pantalla únicamente aparecerán los elementos distintos de cero del vector de incidencia, por lo que los arcos aparecen en desorden y no aparecen todos los arcos de la red. Los intervalos de capacidad tampoco aparecen aquí.

Recuerde que las soluciones no son únicas ya que se pueden encontrar diferentes representaciones para el mismo flujo.

En la siguiente red se aplicará el algoritmo de Realización con el paquete; usted podrá ver como es el despliegue de resultados del algoritmo.



El archivo de datos correspondiente a esta red, es el siguiente:

**ARCHIVO DE DATOS**

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

La pantalla final que despliega los resultados del algoritmo se presenta a continuación:

X=	9( 6, 2)	8( 5, 6)	7( 5, 1)
LAMBDA	[ 0.0, 0.0]	[ -2.0, 2.0]	[ 1-, 1+]
0.66667	[ 0.00	-2.00	-1.00
	6( 4, 6)	5( 3, 5)	4( 3, 4)
	[ 0, 6]	[ -3.0, 1+]	[ 0.0, 7.0]
	5.00	-3.00	3.00 ]+
	3( 2, 4)	2( 1, 3)	1( 1, 2)
	[ 0.0, 3.0]	[ 0.0, 1+]	[ 0.0, 4.0]
	2.00	0.00	2.00 ]+

Presione ENTER para continuar...

X=	9( 6, 2)	8( 5, 6)	7( 5, 1)
LAMBDA	[ 0.0, 0.0]	[ -2.0, 2.0]	[ 1-, 1+]
0.50000	[ 0.00	-2.00	-1.00
	6( 4, 6)	5( 3, 5)	4( 3, 4)
	[ 0, 6]	[ -3.0, 1+]	[ 0.0, 7.0]
	5.00	-3.00	5.00 ]+
	3( 2, 4)	2( 1, 3)	1( 1, 2)
	[ 0.0, 3.0]	[ 0.0, 1+]	[ 0.0, 4.0]
	0.00	2.00	0.00 ]+

Presione ENTER para continuar...

MU		5( 3, 5)	2( 1, 3)	7( 5, 1)
1.00000	[	1.00	1.00	1.00 ]

Presione ENTER para continuar . . .

**EL FLUJO HA QUEDADO EXPRESADO COMO COMBINACION LINEAL  
DE FLUJOS EXTREMOS Y CIRCUITOS DE CAPACIDAD ILIMITADA**

Presione ENTER para continuar . . .

Finalmente usted regresará al menú principal del algoritmo.

**ALGORITMO PARA LA REPRESENTACION  
EXTREMA DE FLUJOS**

**MENU**

- 1.- Crear archivo
- 2.- Utilizar archivo existente
- 3.- Ir al editor

ESC : salir

Para aplicar el otro algoritmo oprima ESC y regresará al menú principal del programa. Para salir oprima de nuevo la tecla ESC.

**MENU PRINCIPAL**

- 1.- Algoritmo de Realización**
- 2.- Representación Extrema de Flujos**

**ESC : salir**

## CONCLUSIONES

El objetivo del trabajo fue logrado con éxito fomentándose un camino propio para la teoría de redes que se hace ahora más independiente de la programación lineal, recurrida frecuentemente para resolver muchos de los problemas análisis de redes, al ser considerado como una particularización de sus teorías.

La dualidad siempre presente en programación lineal fue un rasgo que continua siendo característico en la teoría de redes y que fue tratado con particular ahínco principalmente como complemento para el análisis de flujos, interés de esta tesis. Es de hacer notar que la dualidad también se halla en otros problemas de la teoría de redes como lo es el de flujo máximo - corte mínimo que no fueron incluidos por no ser de interés para el desarrollo de este trabajo.

Una hipótesis de generalización de estas teoría para problemas de optimización, aún cuando no necesariamente estén definidos sobre gráficas surge de manera naturel, sugiriendo una continuación del trabajo.

Una gran cantidad de similitudes entre conceptos, resultados y algoritmos fue encontrada entre los flujos y los diferenciales, por lo que aún cuando ciertos resultados no eran de importancia para el logro de los objetivos del trabajo, estos se mencionaron para mostrar dichas analogías, que son cristalizadas en una tabla al final del tercer capítulo.

El paquete de cómputo también tuvo resultados positivos. Hasta ahora por lo menos a nivel de estudiantes, la mayoría de los algoritmos de redes que habían sido programados, se basaban en arreglos enfocándose así únicamente a encontrar un instrumento más didáctico. En este paquete se trabaja con pilas a las que considero como una opción más viable para aplicaciones más extensas proponiendo una continuación del trabajo en el área de cómputo para optimización de algoritmos.

## APENDICE

Guía de referencia sobre los principales procedimientos y funciones con sus parámetros.

### ACTUALIZAFLUJO

---

Sintaxis :

Procedure ACTUALIZAFLUJO;

Descripción :

Modifica el flujo con base en el valor de alfa y en la trayectoria o circuito detectado durante la ejecución del algoritmo de Realización.

### ACTUALIZA\_M

---

Sintaxis :

Procedure ACTUALIZA\_M;

Descripción :

Asigna un 1 al campo M del registro correspondiente a los arcos del corte detectado durante la ejecución del algoritmo de Realización.

### ALGUNATRAYEC

---

Sintaxis :

Procedure ALGUNATRAYEC;

Descripción :

Encuentra una trayectoria compatible con la coloración aún cuando no se proporcione un conjunto  $N^*$ .

## CHECA

---

**Sintaxis :**

Procedure CHECA(var linea:string; max,min:real; var variable:real);

**Descripción :**

Revisa que la cadena de caracteres que van del principio de la cadena linea hasta el primer espacio encontrado se encuentre en el intervalo cerrado [min, max] y deposita dicha cadena en la variable llamada variable. Si previamente se asigna el valor TRUE a la variable cota este procedimiento también revisará si existen la etiqueta de I+ o la de I- que significan más y menos infinito respectivamente, en cuyo caso asigna la constante masinfinito(1.0E37) y meninfinito(-1.0E37) a la variable variable.

## CONTENIDO

---

**Sintaxis :**

Procedure CONTENIDO(var arch:text);

**Descripción :**

Procede a la lectura y validación de la información contenida en el archivo arch.

## COLOREA

---

**Sintaxis :**

Procedure COLOREA;

**Descripción :**

Asigna la coloración para flujos conformables a la lista contenida en cola1ª.referencia.

ESTA TESIS NO DEBE  
SALIR DE LA BIBLIOTECA

## COLOREA2

---

**Sintaxis :**

Procedure COLOREA2;

**Descripción :**

Asigna la siguiente coloración a la lista contenida en el apuntador cola1^referencia :

**Verde** si  $Cota\ inferior < Flujo < Cota\ superior$

**Rojo** si  $Flujo=Cota\ inferior.$

**Rojo** si  $Flujo=Cota\ superior$

## CREAR\_ARCH

---

**Sintaxis :**

Procedure CREAR\_ARCH;

**Descripción :**

Creo un archivo de datos cuyo nombre, trayectoria y datos son proporcionados por el usuario.

## DATOSERROR

---

**Sintaxis :**

Procedure DATOSERROR;

**Descripción :**

Despliega mensajes de errores detectados en el archivo de datos que contiene la información de la red.

## **ENCUENTRA\_CIRCUITO**

---

**Sintaxis :**

Procedure ENCUENTRA\_CIRCUITO(var circuito:boolean; var arco\_circuit:registro);

**Descripción :**

En caso de existir un circuito compatible con la coloración nos proporciona los datos de un arco (arco\_circuit) perteneciente a él. En este caso la variable circuito adquirirá el valor de verdadero.

## **ENCUENTRACORTE**

---

**Sintaxis :**

Procedure ENCUENTRACORTE;

**Descripción :**

Subrutina del procedimiento ENRUTAR que se encarga de detectar los cortes compatibles con la coloración.

## **ENCUENTRANODOS**

---

**Sintaxis :**

Procedure ENCUENTRANODOS;

**Descripción :**

Guarda en una lista los nodos existentes en la red.

## ENRUTAR

---

**Sintaxis :**

Procedure ENRUTAR(Nm:conj\_ya; Nme:conj\_ya; l:lista; var trayec:lista; var Qm:lista; var Qme:lista; var sitray:boolean; var sicorte:boolean);

**Descripción :**

Aplica el algoritmo de Enrutamiento al conjunto de arcos de la lista l, con Nm (N<sup>+</sup>) y Nme (N<sup>-</sup>). El enrutamiento resultante queda en la variable trayec. Si existe trayectoria sitray adquiere el valor de TRUE y si corte, sicorte tiene el valor da TRUE. En caso de que sicorte sea verdadero, los arcos del corte quedan en Qm y Qme que son la parte positiva y negativa del corte respectivamente.

## ESCRIBETRAYEC

---

**Sintaxis :**

Procedure ESCRIBETRAYEC;

**Descripción :**

Presenta los resultados finales en pantalla después de aplicar el algoritmo de Realización.

## FORMA\_2HIJOS

---

**Sintaxis :**

Procedure FORMA\_2HIJOS;

**Descripción :**

Crea dos flujos más cercanos a ser extremos a partir de uno que no lo es; en caso de tener un circuito de capacidad ilimitada lo guarda en una lista y forma solamente un flujo más cercano a ser extremo.

## LEER\_ARCH

---

**Sintaxis :**

Procedure LEER\_ARCH;

**Descripción :**

Lee el archivo de datos con la información de la red y procede a la validación y posteriormente a la asignación de información a las variables correspondientes que serán utilizadas a lo largo del programa.

## MARCO

---

**Sintaxis :**

Procedure MARCO;

**Descripción :**

Dibuja un marco alrededor de la pantalla.

## MAXIMO

---

**Sintaxis :**

Procedure MAXIMO;

**Descripción :**

Calcula las  $\mu$ 's durante la ejecución del algoritmo para Representación Extrema de Flujos.

## MINIMO

---

**Sintaxis :**

Function MINIMO(l:lista):real;

**Descripción :**

Calcula el valor de alfa durante la ejecución del algoritmo de Realización.

## **MINTY**

---

**Sintaxis :**

**Procedure MINTY(Temp:lista);**

**Descripción :**

**Aplica el algoritmo de Minty a los arcos contenidos en la lista temp.**

## **MINTY2**

---

**Sintaxis :**

**Procedure MINTY2;**

**Descripción :**

**Aplica el algoritmo de Minty para un arco verde.**

## **MUESTRA**

---

**Sintaxis :**

**Procedure MUESTRA;**

**Descripción :**

**Despliega los resultados del algoritmo para Representación extrema de Flujos.**

## **MUESTRA2**

---

**Sintaxis :**

**Procedure MUESTRA2;**

**Descripción :**

**Despliega los resultados correspondientes a circuitos de capacidad limitada.**

## **OPCION**

---

**Sintaxis :**

**Procedure OPCION;**

**Descripción :**

Genera el menú y manda a llamar las subrutinas para creación de archivo, utilización de archivo de datos previamente creado y salida al editor de texto mediante las subrutinas : CREA\_ARCH, LEE\_ARCH y la función exec para la ejecución de l.exe (editor).

## **PANTALLAFIN**

---

**Sintaxis :**

**Procedure PANTALLAFIN;**

**Descripción :**

Despliega mensaje final del algoritmo de Realización.

## **BIBLIOGRAFIA**

**Rockafellar, R. Tyrrel**  
Network flows and monotropic optimization.  
Editorial J.Wiley  
New York, 1984.

**Bertsekas**  
Linear network optimization: algorithms and codes.  
Editorial MIT y Cambirdge.  
Massachusetts, 1991.

**Christófidis**  
Graph theory : An algorithmic approach  
Editorial Academic  
New York, 1975

**Busacker, Saaty**  
Finite graphs and networks : An introduction with application.  
Editorial Mc Graw Hill  
Estados Unidos, 1965.

**Günther Ruhe**  
Algorithmic aspects of flows in networks.  
Editorial Kluwer Academic.  
Dordrecht, 1991.

**Bazaraa, Jarvis, Sherali**  
Linear programming and network flows  
Editorial WILEY, segunda edición.  
Singapore, 1990.

**Victor R. Pérez Pérez**  
Problemas de acoplamiento en gráficas  
Tesis para obtener el título de actuario  
Cd. Universitaria, 1993.

**Ma. del Carmen Hernández Ayuso**  
**Seminario de análisis combinatorio (optimización y combinatoria)**  
**Serie: notas de clase**  
**Publicaciones del Depto. de Matemáticas**  
**de la Fac. de Ciencias UNAM, segunda edición**  
**México, 1992.**

**SIAM Review**  
**Society for industrial and applied mathematics.**  
**Vol. 33, No. 2, Junio de 1991.**

**Stephen O' Brien**  
**Turbo Pascal 5.5 the complete reference**  
**Editorial Borland-Osborne/Mc Graw Hill**  
**Estados Unidos, 1989.**

**Ma. del Carmen Hernández Ayuso**  
**Análisis de redes**  
**Publicaciones del Depto. de Matemáticas**  
**de la Fac. de Ciencias UNAM, segunda edición**  
**México, 1992.**