

UNIVERSIDAD AUTONOMA DE NUEVO LEON
FACULTAD DE INGENIERIA MECANICA Y ELECTRICA



**ALGORITMOS QUE MINIMIZAN EL NUMERO DE ARCOS EN
REDES BIPARTITAS COMPLETAS CON FLUJOS
BALANCEADOS**

POR

JOSE LUIS MARTINEZ FLORES

**Como requisito parcial para obtener el Grado de DOCTOR EN
INGENIERIA con Especialidad en Ingeniería de Sistemas**

ENERO, 1996

1971

1972

1973

1974

1975

1976

1977

1978

1979

1980

1981

1982

1983

1984

ALGORITMOS QUE MINIMIZAN EL NUMERO DE ARCOS EN

REDES BIPARTITAS CON FLUJOS BALANCEADOS

JOSE LUIS MARTINEZ F.

NUMERO DE ARCOS EN

REDES BIPARTITAS CON FLUJOS BALANCEADOS

JOSE LUIS MARTINEZ F.

NUMERO DE ARCOS EN

REDES BIPARTITAS CON FLUJOS BALANCEADOS

JOSE LUIS MARTINEZ F.

NUMERO DE ARCOS EN

REDES BIPARTITAS CON FLUJOS BALANCEADOS

JOSE LUIS MARTINEZ F.

NUMERO DE ARCOS EN

REDES BIPARTITAS CON FLUJOS BALANCEADOS



1020112150

UNIVERSIDAD AUTONOMA DE NUEVO LEON
FACULTAD DE INGENIERIA MECANICA Y ELECTRICA



ALGORITMOS QUE MINIMIZAN EL NUMERO DE ARCOS EN
REDES BIPARTITAS COMPLETAS CON FLUJOS
BALANCEADOS

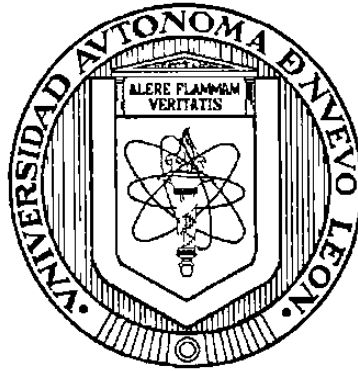
POR

JOSE LUIS MARTINEZ FLORES

Como requisito parcial para obtener el Grado de DOCTOR EN
INGENIERIA con Especialidad en Ingeniería de Sistemas

ENERO, 1996

UNIVERSIDAD AUTONOMA DE NUEVO LEON
FACULTAD DE INGENIERIA MECANICA Y ELECTRICA



ALGORITMOS QUE MINIMIZAN EL NUMERO DE ARCOS EN
REDES BIPARTITAS COMPLETAS CON FLUJOS
BALANCEADOS

Por

JOSE LUIS MARTINEZ FLORES

Como requisito parcial para obtener el Grado de DOCTOR EN
INGENIERIA con Especialidad en Ingeniería de Sistemas

Enero, 1996



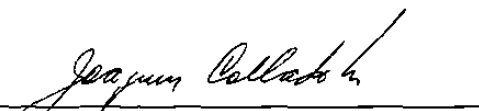
9

**ALGORITMOS QUE MINIMIZAN EL NUMERO DE ARCOS
EN REDES BIPARTITAS COMPLETAS CON FLUJOS
BALANCEADOS**

Aprobación de la Tesis:




Asesor: Dra. Ada Margarita Alvarez Socarrás



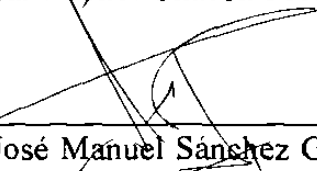
Dr. Joaquín Collado Moctezuma



Dr. Oscar Flores Rosales



Dr. Oscar Leonel Chacón Mondragón



Dr. José Manuel Sánchez García



Subdirector de Estudios de Postgrado: M.C. Marco Antonio Méndez Cavazos

¡A mi familia!...bajo mi concepto.

AGRADECIMIENTOS

En primera instancia quiero agradecer sinceramente a la Dra. Ada Margarita Alvarez Socarrás, asesora de esta tesis, por toda la paciencia, confianza y dedicación que puso en mi persona para poder concluir con esta meta. También quiero hacer un reconocimiento al Dr. Joaquín Collado Moctezuma por sus acertadas observaciones y comentarios con respecto a los resultados de esta tesis. Asimismo, al Dr. Oscar Flores Rosales, coordinador académico del programa doctoral en Ingeniería de Sistemas, por su apoyo incondicional durante toda mi estancia en el programa. Finalmente, a los Dres. Oscar Leonel Chacón Mondragón y José Manuel Sánchez García quiero agradecerles su colaboración para que los resultados de esta tesis fueran lo más claro posibles.

RESUMEN

Las tareas de investigación consideradas en esta tesis tienen como propósito la formulación de los modelos matemáticos y la búsqueda de la solución general analítica a problemas determinísticos de diseño de redes donde se desea minimizar: el máximo valor del flujo sobre los arcos y el número de arcos utilizados. Particularmente se analizan los problemas de diseño de redes, en los cuales las redes son bipartitas completas con flujos uniproducción y con flujos multiproducción, donde se considera: *i)* los flujos son enteros no-negativos, *ii)* los arcos son de costo unitario y *iii)* los costos por unidad de flujo sobre los arcos son despreciables.

En el presente trabajo se formula el modelo matemático para redes arbitrarias con flujos uniproducción, se determina un algoritmo de solución para el caso de redes bipartitas completas y se demuestra matemáticamente su optimalidad y que su complejidad es polinomial. Asimismo se formula el modelo matemático para redes arbitrarias con flujos multiproducción y se determina un algoritmo que encuentra soluciones enteras aproximadas para el caso de redes bipartitas completas.

La ventaja de estos algoritmos es que resuelven en tiempo polinomial un problema particular del problema general de diseñar una red óptima, el cual es un problema *NP-hard*. Además, los algoritmos diseñados en este trabajo de tesis son aplicables a problemas de distribución de productos y a problemas de redes de comunicación por computadora.

NOTACION

:	Tal que.
\triangle	Igualdad por definición.
\mathbb{R}	Conjunto de números reales.
$\lceil x \rceil$	Menor entero mayor o igual que x , $x \in \mathbb{R}$.
$\lfloor x \rfloor$	Mayor entero menor o igual que x , $x \in \mathbb{R}$.
$ I $	Número de elementos del conjunto I .
$ M ^+$	Número de celdas con componentes positivas de la matriz M .
f^+	Número de componentes positivas del flujo f , sobre la red.
X'	Transpuesta de X , donde X es una matriz.
$X \leq Y$	si y solo si $x_{ij} \leq y_{ij}$, para cada i, j . Donde X y Y son matrices.
\mathbb{R}^+	Conjunto de números reales no negativos.
\mathbb{R}_e^+	$\{\mathbb{R}^+ \cup \infty\}$.
\mathbb{R}^n	$\{(x_1, x_2, \dots, x_n) : x_i \in \mathbb{R}\}$.
\mathbb{R}^{+n}	$\{(x_1, x_2, \dots, x_n) : x_i \in \mathbb{R}^+\}$.
\mathbb{Z}	Conjunto de números enteros.
\mathbb{Z}_+	Conjunto de números enteros no negativos.
$\mathbb{Z}_+^{m \times n}$	Conjunto de matrices de dimensión $m \times n$ con componentes en \mathbb{Z}_+ .
N	Conjunto de nodos.
S	Conjunto de fuentes.
X^+	Conjunto de fuentes en una red bipartita.
T	Conjunto de sumideros.
X^-	Conjunto de sumideros en una red bipartita.
R	Conjunto de intermedios.
$a(x)$	Oferta en el nodo x .
$b(x)$	Demanda en el nodo x .
A	Conjunto de arcos.
$A(x)$	$\{y \in N : (x, y) \in A\}$.
$B(x)$	$\{y \in N : (y, x) \in A\}$.
g	Grafo.
$g(N, A)$	Grafo (dirigido) con nodos en N y arcos en A .
(N, A)	Red con nodos en N y arcos en A .
(x, y)	El arco del nodo inicial x al nodo final y .
$c(x, y)$	Capacidad del arco (x, y) .
$f(x, y)$ f_{xy}	Flujo (uniprodueto) sobre el arco (x, y) .
$f^\tau(x, y)$ f^τ_{xy}	Flujo del producto τ sobre el arco (x, y) .
$f(x, y)$	Flujo multiproducto sobre el arco (x, y) .

TABLA DE CONTENIDO

Capítulo	Página
1. INTRODUCCION	1
2. ANTECEDENTES	6
2.1 Introducción.	6
2.2 Conceptos Básicos de Grafos.	6
2.3 Redes y Flujos	8
2.4 Clasificación de los Problemas de Flujos en Redes	11
2.5 Flujos Multiproducto en Problemas de Análisis de Redes	12
2.5.1 Problemas Lineales de Flujos Multiproducto.	14
2.5.2 Problemas No Lineales de Flujos Multiproducto.	16
2.6 Flujos Multiproducto en Problemas de Diseño de Redes	17
2.6.1 Un Caso Especial del Problema de Diseño de Red Optima	18
2.7 Definición del Problema de Investigación .	21
3. UN ALGORITMO POLINOMIAL QUE MINIMIZA EL NUMERO DE ARCOS EN UNA RED BIPARTITA COMPLETA CON FLUJOS UNIPRODUCTO BALANCEADOS.	26
3.1 Introducción.	26
3.2 Formulación del Problema.	27
3.3 Solución al Problema Uniproducto	30
3.3.1 Definiciones Básicas	30
3.3.2 Determinación de la Solución al Problema Uniproducto	32
3.3.2.1 Búsqueda de δ -ciclos	35
3.3.2.2 Algoritmo General de Diseño (AGD)	40
3.4 Optimalidad del Algoritmo AGD	41
3.5 Optimalidad Consistente del Algoritmo AGD .	44
3.6 Resumen	44

Capítulo	Página
4. UN ALGORITMO POLINOMIAL QUE MINIMIZA EL NUMERO DE ARCOS EN UNA RED BIPARTITA COMPLETA CON FUJOS MULTIPRODUCTO BALANCEADOS	46
4.1 Introducción.	46
4.2 Formulación del Problema.	46
4.3 Solución al Problema Multiproducto	51
4.3.1 Determinación de Cotas Inferiores.	52
4.3.2 Solución para $T=2$	54
4.3.2.1 Búsqueda de δ -ciclos no-disminuidores de celdas no-operables	59
4.3.2.2 Algoritmo General de Diseño Biproducto (AGDB)	64
4.3.3 Solución Entera Aproximada para el Caso General Multiproducto	66
4.4 Resumen	67
5. APLICACION A UN PROBLEMA DE DISTRIBUCION DE PRODUCTOS.	68
6. CONCLUSIONES Y RECOMENDACIONES	81
BIBLIOGRAFIA.	84
APENDICES.	88
APENDICE A.- UN EJEMPLO DONDE EL PROBLEMA DEL TRANSPORTE CON CAPACIDADES NO RESUELVE EL PROBLEMA UNIPRODUCTO.	89
APENDICE B.- DETERMINACION DEL VALOR F_0	92
APENDICE C.- UN EJEMPLO DONDE EXISTEN DOS SOLUCIONES A UN PROBLEMA UNIPRODUCTO.	98

LISTA DE FIGURAS

Figura	Página
1. Representación gráfica de un ciclo en una red y en una matriz	31
2. δ -ciclo de redistribución, donde $\delta=2$ y $F_0=6$	32
3. δ -tabla asociada a la matriz de distribución de la Figura 2, donde $\delta=2$	36
4. Distribución de flujo donde se encuentra la matriz T -flujos M , para $T=2$	52
5. Distribución de flujos uniproducto donde se encuentra la matriz transformable M_1	56
6. Distribución de flujos uniproducto donde se encuentra la matriz fija M_2	57
7. Ofertas y demandas del producto PARA1.	68
8. Ofertas y demandas del producto PARA2.	68
9. Distribución inicial para el cálculo de F_0 del producto PARA1.	70
10. Solución inicial para el producto PARA1	72
11. Solución inicial para el producto PARA2	72
12. 1-ciclo disminuidor de celdas ocupadas en la celda (4,9).	73
13. 1-ciclo disminuidor de celdas ocupadas en la celda (3,8).	74
14. Solución óptima, número de celdas ocupadas es la cota inferior 13	74
15. 2-ciclo disminuidor de celdas ocupadas en la celda (4,9).	75
16. 2-ciclo disminuidor de celdas ocupadas en la celda (3,8).	75

Figura	Página
17. Solución óptima, número de celdas ocupadas es la cota inferior 13	75
18. Distribución inicial donde se encuentra la solución inicial M	76
19. 2-ciclo no disminuidor de celdas no operables en la celda (3,7).	78
20. 1-ciclo no disminuidor de celdas no operables en la celda (3,7).	78
21. No existe 1-ciclo no disminuidor de celdas no operables. Empalme máximo.	78
22. 1-ciclo no disminuidor de celdas no operables en la celda (3,9).	79
23. No existe 1-ciclo no disminuidor de celdas no operables. Seleccionar nueva celda removible.	79
24. 2-ciclo no disminuidor de celdas no operables en la celda (2,8).	80
25. No existe 1-ciclo no disminuidor de celdas no operables. Empalme máximo	80
26. Solución entera aproximada al problema planteado	80
27. Matriz inicial de costos	90
28. Solución óptima del Problema de Transporte, obtenida por el Método Simplex	91
29. Solución óptima obtenida por el algoritmo AGD	91
30. Problema inicial.	99
31. Distribución inicial	99
32. Una solución al problema uniproducto P	100
33. Una solución al problema uniproducto P diferente a la de la Figura 32	100

CAPITULO 1

INTRODUCCION

Uno de los factores críticos en el área de Administración de Sistemas de Información es "La planeación y administración de redes de comunicaciones" [Niederman *et al*, 1991; Janz *et al*, 1995]. Más específicamente, este factor crítico ha aumentado su importancia pasando del décimo lugar en 1990 [Niederman *et al*, 1991], al quinto lugar en 1994 [Janz *et al*, 1995]. Una de las principales premisas para que esto haya sucedido es que las redes de comunicaciones permiten administrar más efectivamente la interdependencia organizacional de una compañía [Rockart and Short, 1991].

Una de las componentes de la planeación de una red de comunicaciones es el diseño topológico (diseño) de la red. El diseño de una red involucra, entre otras cosas, la asignación de enlaces y sus capacidades para conectar los nodos de la red [Ahuja, 1985; Schwartz, 1977], y tiene implicaciones en su desempeño y costo. Las ubicaciones de los nodos, los enlaces y sus velocidades determinan directamente el tiempo de tránsito, el costo de construcción y operación de la red. Por lo tanto, es muy importante determinar algoritmos de diseño de redes que

permitan seleccionar óptimamente los enlaces y sus capacidades dentro de diferentes restricciones, tales como conectividad, retardos de transmisión de mensajes, costo y tráfico de la red.

Por otra parte, uno de siete factores críticos de éxito en la industria manufacturera es la "Entrega Competitiva" [Martin, 1987] (referenciado en [Chase and Aquilano, 1989]), lo cual involucra una entrega que satisfaga las demandas de los clientes. En otras palabras, se debe diseñar una distribución que satisfaga las demandas de los clientes.

La distribución genera costos, los cuales son llamados *costos de distribución*. Estos costos están directamente relacionados al envío de los productos a los clientes. El objetivo de la Administración de la Distribución [Chase and Aquilano, 1989; Adam and Ebert, 1991] es satisfacer las demandas de envío al cliente a un mínimo costo. Por lo tanto, es muy importante determinar algoritmos que permitan seleccionar una distribución óptima, la cual esté sujeta a diferentes restricciones, tales como tiempo de entrega, demanda completa del cliente, etc.

Haciendo una analogía entre los problemas mencionados anteriormente (de redes de comunicación por computadora y de distribución) podemos hacer una abstracción de ellos y llevarlos a un problema general existente en la Teoría de Flujos en Redes [Ford and Fulckerson, 1962; Bazaraa, Jarvis and Sherali, 1990; Ahuja, Magnanti and Orlin, 1993] llamado "Diseño de Red Óptima".

Los problemas en diseño de redes óptimas se caracterizan por la búsqueda de una configuración de red que satisfaga un conjunto dado de requerimientos. Hay una gran variedad de trabajos que existen en la literatura donde se involucran problemas de diseño de redes: en sistemas de computación distribuida [Shen and Tsai, 1985; Ma *et al*, 1982; Naor and Roth, 1995], en redes de computadoras [Shwartz, 1977; Ahuja, 1985; Zwass and Veroy, 1988] en sistemas eléctricos de potencia [Aoki *et al*, 1990; Chiang and Jean-Jumeau, 1990], en síntesis de redes [Gomory and Hu, 1962, 1964; Minoux, 1987, 1989; Itai and Pradhan, 1984; Sridhar and Chandrasekaran, 1992], en redes de transporte [Guardado, 1982; Alvarez y Guardado, 1992; Alvarez, 1985, 1993], y en otros tipos de problemas [Bienstock and Gonlok, 1995; Goemans *et al*, 1994; Lucertini and Paletta, 1986].

Uno de los problemas de síntesis en redes bipartitas completas, en redes del tipo de transporte, es el problema de encontrar el mínimo de los máximos flujos sobre cualquier arista (arco) de la red [Guardado, 1982], esto es, encontrar la mínima capacidad sobre cualquier arco de la red. Esto permite balancear lo mejor posible los flujos sobre los arcos, es decir, dado cualquier otro valor de la capacidad sobre todos los arcos, obviamente mayor que la mínima encontrada, generaría una diferencia mayor entre los valores de los flujos sobre los arcos de la red.

Una conjetura es encontrar una red que sujeta a la capacidad mínima, tenga un número mínimo de aristas (arcos) [Guardado, 1982]. Este problema es

considerado como un problema de diseño óptimo. Esta conjetura fue resuelta parcialmente por Alvarez y Guardado [1992], diseñando un algoritmo de complejidad polinomial que determina el número mínimo de arcos, pero no demuestran matemáticamente la optimalidad del algoritmo. Por otra parte, una extensión natural de estos resultados es la generalización de ellos a problemas de flujos multiproducto en redes arbitrarias.

Dado lo anterior, la presente tesis tiene los siguientes objetivos:

I. Para el caso de redes con flujos uniproducto:

I.1) Formular un modelo matemático que exprese el problema de diseño de redes arbitrarias en el cual se desee: *i)* minimizar la capacidad de los arcos de la red, *ii)* minimizar el número de arcos de la red sujetos a *i)*.

I.2) Reformular el algoritmo de solución para el caso de redes bipartitas completas, expuesto por Alvarez y Guardado [1992], demostrar matemáticamente la optimalidad y que, en efecto, es de complejidad polinomial.

II. Para el caso de redes con flujos multiproducto:

II.1) Formular el modelo matemático que exprese el problema de diseño de redes arbitrarias en el cual se desee: *i)* minimizar la capacidad de todos los arcos de la red; *ii)* minimizar el número de arcos de la red sujetos a *i)*.

II.2) Diseñar un algoritmo de complejidad polinomial que resuelva el problema formulado en II.1), para el caso de redes bipartitas completas.

La resolución de los objetivos de la tesis permite resolver el problema de diseño de redes de comunicaciones donde se desee encontrar la mínima capacidad que deben de tener todos los enlaces de la red a un costo mínimo, suponiendo que el costo de construcción del enlace es el mismo. En el Capítulo 5 se da un ejemplo aplicado a redes de distribución.

CAPITULO 2

ANTECEDENTES

2.1 Introducción

En este capítulo se hace una recolección de algunos de los conceptos y resultados de la Teoría de Flujos en Redes encontrados en la literatura, que son necesarios para fundamentar las soluciones de los problemas de investigación planteados en la presente tesis.

2.2 Conceptos Básicos de Grafos

Un *grafo dirigido*, o simplemente *grafo*, está formado por un conjunto finito N cuyos elementos son llamados *nodos*, y un conjunto finito A de parejas ordenadas de nodos, cuyos elementos son llamados *arcos dirigidos*, o simplemente *arcos*. Denotaremos el grafo anterior como $\mathcal{G}(N,A)$, los nodos como x, y , y el arco que va del nodo x al nodo y como (x,y) . Los nodos x, y son llamados extremos del arco, o bien, *nodo inicial* y *nodo terminal* respectivamente.

Algunos problemas en Teoría de Grafos requieren solamente del conocimiento de los nodos x, y en cada arco (x,y) . En casos como este, la *dirección* no necesita

especificarse y el arco *no dirigido* es llamado *arista*. Comúnmente se denota por E el conjunto de aristas. Un grafo cuyas direcciones de arco no se especifican se llama grafo *no dirigido* y es denotado por $\mathfrak{G}(N,E)$.

Un arco que tiene por extremos un mismo nodo se llama *lazo*. Un nodo y un arco se dice que son *incidentes* si el nodo es un extremo del arco. Dos arcos se dice que son *incidentes* si ambos tienen como incidente un mismo nodo. Dos nodos se dice que son *adyacentes* si existe un arco que los una.

Sea un nodo $x \in N$, denotamos $A(x)$ como el conjunto de todos los nodos $y \in N$ tal que $(x,y) \in A$, esto es,

$$A(x) = \{y \in N : (x,y) \in A\}.$$

En otras palabras, $A(x)$ es el conjunto de todos los nodos terminales cuyos arcos inician en x .

Similarmente, denotamos $B(x)$ como el conjunto de todos los $y \in N$ tal que $(y,x) \in A$, esto es,

$$B(x) = \{y \in N : (y,x) \in A\}.$$

En otras palabras, $B(x)$ es el conjunto de todos los nodos iniciales cuyos arcos terminan en x .

Una *trayectoria de x_1 a x_n* es una secuencia de arcos $(x_1, x_2); \dots; (x_p, x_{p+1}); \dots; (x_n, x_n)$ de tal forma que el nodo terminal de cada arco es el mismo que el nodo inicial del arco que le sucede. Los nodos x_1, x_n son llamados *nodo inicial* y *nodo final*

de la trayectoria, respectivamente. La *longitud* de una trayectoria es igual al número de arcos que hay en ella. Si $x_1 = x_n$ y $n > 2$, entonces la trayectoria se llama *circuito*. Podemos observar en la definición anterior que si $n = 2$, entonces la trayectoria es un lazo.

Una *cadena* de x_1 a x_n es una secuencia de arcos $\alpha_1, \alpha_2, \dots, \alpha_{n-1}$ tal que los extremos del arco α_i son x_i, x_{i+1} para $i = 1, \dots, n-1$. De esta manera, $\alpha_i = (x_i, x_{i+1})$ o $\alpha_i = (x_{i+1}, x_i)$. Longitud, nodo inicial, y nodo terminal de una cadena pueden ser definidos similarmente a la trayectoria. Si $x_1 = x_n$ entonces la cadena se llama *ciclo*. Una trayectoria, circuito, cadena, o ciclo se dice que es *simple* si ningún nodo incide a más de dos de sus arcos o aristas según corresponda, (*i.e.*, si la trayectoria, circuito, cadena, o ciclo no contiene circuitos o ciclos propios).

2.3 Redes y Flujos

En los años de 1950's, el campo de Teoría de Grafos tomó esencialmente dos direcciones: *i) teoría algebraica de grafos*, *ii) optimización en grafos*, [Minieka, 78]. En la optimización en grafos toma una gran importancia asignarle, de alguna manera, uno o más números reales a los arcos del grafo. Aquí es donde surge el concepto de red.

Una *red* es un grafo en el cual se asocian uno o más números reales no negativos con cada arco y la denotaremos por (N, A) . Por lo tanto, al hablar de una red implícitamente relacionamos, de alguna manera, un número real no negativo

con cada arco de la red.

Sea (N,A) una red, una de las asociaciones fundamentales entre A y \mathbb{R}_e^+ es la llamada *capacidad*. Una función $c:A \rightarrow \mathbb{R}_e^+$ tal que a cada arco $(x,y) \in A$ le asocia el número real no negativo $c(x,y)$ es llamada *función capacidad*. De esta forma, $c(x,y)$ es la capacidad del arco $(x,y) \in A$, la cual puede ser pensada intuitivamente como la cantidad máxima de algún producto que puede ser enviada de x a y , a través del arco (x,y) . Es importante hacer notar que algunas veces será conveniente permitir capacidades de arco infinitas, lo cual significa que no hay restricciones sobre la capacidad del arco.

Asimismo, una asociación entre A y \mathbb{R}^+ , es el concepto de *flujo*, el cual es definido como sigue. Sean s y t dos nodos de N , un *flujo uniprodueto con capacidades* (o simplemente *flujo con capacidades*) de valor F de s a t [Ford and Fulckerson, 1962] en (N,A) es una función $f: A \rightarrow \mathbb{R}^+$ que satisface:

$$\sum_{y \in A(x)} f(x,y) - \sum_{y \in B(x)} f(y,x) = \begin{cases} F, & \text{si } x = s \\ 0, & \text{si } x \neq s, t \\ -F, & \text{si } x = t \end{cases} \quad (2.1)$$

$$0 \leq f(x,y) \text{ para cada } (x,y) \in A, \quad (2.2)$$

$$f(x,y) \leq c(x,y) \text{ para cada } (x,y) \in A, \quad (2.3)$$

donde s es llamado *fuerate*, t *sumidero*, y los restantes nodos son llamados *intermedios*.

La ecuación (2.1) es llamada de *conservación de flujo*, la desigualdad (2.2) es llamada *restricción de no-negatividad* y la (2.3) es llamada *restricción de capacidades*. Si $(x,y) \in A$, el valor $f(x,y) = f_{xy}$ puede ser interpretado como la cantidad de algún producto que es enviada de x a y , a través del arco (x,y) . Si las capacidades de los arcos son infinitas, esto es, si eliminamos la restricción (2.3) en la definición de flujo, se dice que el flujo es *sin capacidades*.

La definición anterior se puede generalizar cuando existen en la red múltiples fuentes y sumideros [Ford and Fulckerson, 1962]. Sea (N,A) una red arbitraria, supongamos que N es particionado en tres subconjuntos: S (fuentes), R (nodos intermedios), y T (sumideros). Un *flujo con capacidades de S a T* es una función real no negativa f definida sobre A que satisface

$$\sum_{y \in A(x)} f(x,y) - \sum_{y \in B(x)} f(y,x) = 0 \quad \text{para } x \in R,$$

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

con valor de flujo

$$F = \sum_{(x,y) \in (S,A(x))} f(x,y) - \sum_{(y,x) \in (B(x),S)} f(y,x),$$

donde $(S,A(x)) \triangleq \{(x,y): x \in S, y \in A(x)\}$ y $(B(x),S) \triangleq \{(y,x): x \in S, y \in B(x)\}$.

No obstante esta generalización, se puede transformar cualquier problema de flujos sobre una red conteniendo dos conjuntos S , T de fuentes y sumideros, respectivamente, en un problema de flujos donde se tenga solamente una fuente

y un sumidero, lo cual se ilustra a continuación.

Si extendemos la red (N,A) a una red (N_e,A_e) adjuntando dos nodos s_e, t_e y todos los arcos $(s_e, S), (T, t_e)$ y extendiendo la función capacidad c definida sobre A a c_e definida sobre A_e por

$$\begin{aligned} c_e(s_e, x) &= \infty, & \text{si } x \in S, \\ c_e(x, t_e) &= \infty, & \text{si } x \in T, \\ c_e(x, y) &= c(x, y), & \text{si } (x, y) \in A. \end{aligned}$$

De esta manera, la restricción f de un flujo f_e de s_e a t_e en (N_e,A_e) es un flujo de S a T en (N,A) . Recíprocamente, un flujo f de S a T en (N,A) puede ser extendido a un flujo f_e de s_e a t_e en (N_e,A_e) definiendo

$$\begin{aligned} f_e(s_e, x) &= \sum_{y \in A(x)} f(x, y) - \sum_{y \in B(x)} f(y, x) & \text{si } x \in S, \\ f_e(x, t_e) &= \sum_{y \in B(x)} f(y, x) - \sum_{y \in A(x)} f(x, y) & \text{si } x \in T, \\ f_e(x, y) &= f(x, y) & \text{si } x \in R. \end{aligned}$$

En consecuencia, cualquier problema de flujos sobre una red conteniendo dos conjuntos S, T de fuentes y sumideros, respectivamente, lo podemos transformar en un problema de flujos donde se tenga solamente una fuente y un sumidero.

2.4 Clasificación de los Problemas de Flujos en Redes

Generalmente se distinguen dos clases generales de problemas de flujos en

redes [Assad, 1978]:

i).- Problemas en análisis de redes. En este tipo de problemas se analiza una configuración de red específica para encontrar el patrón de flujo óptimo con respecto a alguna función objetivo. Por ejemplo, podríamos querer maximizar el flujo total de un producto en un problema de flujo máximo, o minimizar los costos de satisfacer un conjunto dado de requerimientos en problemas de costo mínimo.

ii).- Problemas en diseño de redes. En este tipo de problemas buscamos una configuración de red óptima que satisfaga un conjunto dado de requerimientos. Podríamos, por ejemplo, satisfacer estos requerimientos instalando capacidades sobre la red, y buscar minimizar el costo total de instalación.

2.5 Flujos Multiproducto en Problemas de Análisis de Redes

Si ahora suponemos que podemos asociar a A con el conjunto \mathbb{R}^T , el conjunto de vectores de dimensión T con componentes reales no negativas, el concepto de flujo se modificará al concepto de *flujo multiproducto (flujo T -producto)*, como se verá a continuación.

Sea (N,A) una red con $N = \{x_1, x_2, \dots, x_n\}$ y $A = \{1, 2, \dots, m\}$. Supongamos que en la red se especifican T parejas de nodos, denotados por (s^τ, t^τ) , como parejas fuente-sumidero asociadas con cada uno de los $\tau=1, \dots, T$ productos. Para cada τ denotemos

$f^\tau(x_p, x_j) = f^\tau_{xx}$ el flujo del producto τ sobre el arco (x_p, x_j) ,

$F^\tau =$ el valor del flujo a ser enviado de s^τ a t^τ ,

Un *flujo multiproducto* sobre el arco (x_p, x_j) es un vector de flujos

$f(x_p, x_j) = (f^1(x_p, x_j), f^2(x_p, x_j), \dots, f^T(x_p, x_j))$, tal que para cada $\tau=1, \dots, T$, $f^\tau(x_p, x_j)$ satisface

(2.1)-(2.2) [Assad, 1978]. Esto es,

$$\sum_{x_j \in A(x)} f^\tau(x_p, x_j) - \sum_{x \in B(x)} f^\tau(x, x_p) = \begin{cases} F^\tau, & \text{si } x = s^\tau \\ 0, & \text{si } x \neq s^\tau, t^\tau \\ -F^\tau, & \text{si } x = t^\tau \end{cases} \quad (2.4)$$

$$0 \leq f^\tau(x_p, x_j); \text{ para cada } (x_p, x_j) \in A. \quad (2.5)$$

Definamos, para cada τ , el vector columna $f^\tau = (f^\tau_1, f^\tau_2, \dots, f^\tau_m)'$ tal que para cada $k=1, \dots, m$; f^τ_k es el flujo del producto τ sobre el arco $k \in A$. Denotemos la *matriz de incidencia nodo-arco* de (N, A) como la matriz $E = (e_{ij})$ de dimensión $n \times m$, cada renglón asociado con un nodo y cada columna asociada con un arco, definida por

$$e_{ij} = \begin{cases} +1, & \text{si el nodo asociado al renglón } i \text{ es el extremo inicial del arco asociado con la columna } j. \\ -1, & \text{si el nodo asociado al renglón } i \text{ es el extremo final del arco asociado con la columna } j. \\ 0 & \text{si el nodo asociado al renglón } i \text{ no incide en el arco asociado con la columna } j. \end{cases}$$

Denotemos por g^τ al vector columna de dimensión n con entradas 1 y -1 en los lugares correspondientes a s^τ y t^τ respectivamente y ceros en los restantes lugares y sea $b^\tau = F^\tau g^\tau$. Con esta notación tenemos que las ecuaciones (2.4) y (2.5) se pueden escribir para cada producto τ de la forma:

$$E f^T = F^T g^T \quad b^T$$

$$\mathbf{0} \leq f^T,$$

donde $\mathbf{0}$ es el vector columna nulo en el espacio \mathbb{R}^m .

El problema general de flujos multiproducto puede establecerse [Assad, 1978] como:

$$\text{Min } Z(f^1, f^2, \dots, f^T) \quad (2.6)$$

s. a.

$$E f^T = b^T, \quad (2.7)$$

$$\mathbf{0} \leq f^T, \quad (2.8)$$

$$\phi(f^1, f^2, \dots, f^T) \leq \mathbf{0}, \quad (2.9)$$

donde Z es una función objetivo general (posiblemente no lineal) y ϕ es un vector de restricciones adicionales impuestas en los flujos.

2.5.1 Problemas Lineales de Flujos Multiproducto

Los problemas lineales de flujos multiproducto se distinguen por asignar un costo lineal al flujo de cada producto sobre cada arco. Mediante la siguiente asignación se obtiene el conocido *problema multiproducto de costo mínimo con capacidades*,

$$Z = \sum_{\tau=1, \dots, T} \kappa^\tau f^\tau, \quad \phi(f^1, f^2, \dots, f^T) = \sum_{\tau=1, \dots, T} f^\tau - c \leq \mathbf{0},$$

donde $c = (c_1, c_2, \dots, c_m)'$ con c_k como la capacidad del arco k y $\kappa^\tau = (\kappa_1^\tau, \kappa_2^\tau, \dots, \kappa_m^\tau)$ con

κ_k^τ como el costo por unidad de flujo del producto τ en el arco k .

En otras palabras, el problema multiproducto de costo mínimo con capacidades puede ser escrito como:

$$\text{Min } Z = \sum_{\tau=1, \dots, T} \kappa^\tau \cdot f^\tau \quad (2.10)$$

s. a.

$$E f^\tau = b^\tau$$

$$0 \leq f^\tau$$

$$\sum_{\tau=1, \dots, T} f^\tau \leq c.$$

El problema (2.10) ha sido muy abordado en la literatura, siendo imposible mencionar todos los trabajos que tratan este problema en la Teoría de Flujos en Redes. Sin embargo no podemos dejar de mencionar libros tales como el de Ahuja, Magnanti and Orlin [1993], Bazaraa, Jarvis and Sheraldi [1990] y Kennington and Helgason [1980], los cuales proporcionan un amplio análisis del problema. Asimismo, artículos de recopilación de información tales como el de Assad [1978] y el de Kennington [1978]. Recientemente algunos investigadores han empezado a utilizar el enfoque de puntos interiores para resolver el problema de flujos multiproducto de costo mínimo, uno de los trabajos más recientes desarrollados con este enfoque es el artículo de Kamath y Palmon [1995].

2.5.2 Problemas No Lineales de Flujos Multiproducto

El problema no lineal de flujos multiproducto se distingue por una función objetivo no lineal sobre los flujos. Más específicamente, denotemos por

$$f = \sum_{\tau=1, \dots, T} f^{\tau}$$

el vector *flujo total*. Z es usualmente una función del flujo total f (de tal forma que podemos escribir $Z(f)$ en vez de $Z(f^1, f^2, \dots, f^T)$). Algunas veces $Z(f)$ se separa como:

$$Z(f) = \sum_{k=1, \dots, m} Z_k(f_k)$$

donde f_k es el flujo total sobre el arco k y $Z_k(\bullet)$ la correspondiente función de costo no lineal, la cual es una función solamente del flujo total sobre ese arco.

El problema de flujo multiproducto no lineal en su caso general, puede ser establecido como

$$\text{Min } Z(f) \quad (2.11)$$

s. a.

$$f = \sum_{\tau=1, \dots, T} f^{\tau} \quad (2.12)$$

$$Ef^{\tau} = b^{\tau}, \tau=1, \dots, T \quad (2.13)$$

$$0 \leq f^{\tau} \leq c^{\tau}, \tau=1, \dots, T \quad (2.14)$$

Uno de los problemas de flujos multiproducto no lineales es el definido por la siguiente función

$$Z(f) = \sum_{k=1, \dots, m} Z_k(f_k) \quad (2.15)$$

donde $Z_k(f_k) = 0$ si $f_k=0$, y $Z_k(f_k) = \alpha_k + \beta_k f_k$ si $f_k > 0$, y las componentes de f son enteras, donde α_k representa el costo fijo (carga fija) de construcción del arco k , y β_k representa el costo por unidad de flujo sobre el arco k . Este problema es llamado *el problema de la carga fija* y ha sido resuelto para $\tau=1$ por algoritmos de tipo *branch and bound* [Reagsdale and Mckeown, 1990, 1991; Jensen and Barnes, 1980; Hu, 1969] los cuales son de complejidad exponencial [Garey and Johnson, 1979].

2.6 Flujos Multiproducto en Problemas de Diseño de Redes

Supongamos que tenemos un conjunto de nodos $N = \{x_1, x_2, \dots, x_n\}$. En los problemas de diseño de redes se tiene que el conjunto A son *todos los arcos posibles* que pueden existir entre los nodos de N y se busca encontrar los arcos de la red que minimicen un criterio de costo total. Cuando el criterio de costo involucra a las capacidades de los arcos como variables independientes el problema de diseño es llamado *problema de síntesis*.

El *problema del diseño de red óptima con capacidades* se formula [Minoux, 1989; Ahuja, Magnanti and Orlin, 1993] como un problema de carga fija, esto es,

$$\text{Min } Z(f) \quad (2.11)$$

s. a.

$$f = \sum_{\tau=1, \dots, T} f^\tau \quad (2.12)$$

$$Ef^\tau = b^\tau; \tau=1, \dots, T \quad (2.13)$$

$$0 \leq f^\tau \leq c^\tau; \tau=1, \dots, T \quad (2.14)$$

donde $Z(f) = \sum_{k=1, \dots, m} Z_k(f_k)$, $Z_k(f_k)=0$ si $f_k=0$ y $Z_k(f_k) = \alpha_k + \beta_k f_k$ si $f_k > 0$, las componentes

f son enteras, y c^τ es un vector columna de dimensión m , el cual representa la capacidad sobre la red, para el producto τ .

El problema pertenece a la clase de problemas "NP-hard" [Johnson *et al*, 1978]. Entre los algoritmos que lo resuelven existe uno que da una solución exacta para el caso sin capacidades [Minoux, 1989], basado en un método *branch and bound*, y también un algoritmo de aproximación de complejidad polinomial basado en un método de aproximación primal-dual [Goemans *et al*, 1994], para $\tau=1$.

2.6.1 Un Caso Especial del Problema de Diseño de Red Optima

Sea (N, A) una red de tal forma que $N = \{x_1, x_2, \dots, x_m, x_{m+1}, \dots, x_n\}$ y $A = \{(x_i, x_j) : i=1, \dots, m; j=m+1, \dots, n\}$, entonces (N, A) es llamada *red bipartita completa*. La red (N, A) es bipartita en el sentido que N está dividido en los subconjuntos $S = \{x_1, x_2, \dots, x_m\}$ de fuentes y $T = \{x_{m+1}, x_{m+2}, \dots, x_n\}$ de sumideros (note que $N = S \cup T$ y $R = \emptyset$, el conjunto de nodos intermedios). La red (N, A) es completa en el sentido de que todos los arcos posibles están presentes.

Sea (N,A) una red bipartita completa con fuentes S y sumideros T , con flujos enteros no negativos. Asociemos con cada $x \in S$ un número entero no negativo $a(x)$, que significa la *oferta* de un producto en x , y con cada $x \in T$ un número entero no negativo $b(x)$, que significa la *demanda* del mismo producto en x . Por lo tanto, el valor del flujo que sale de S es, a lo más,

$$\sum_{x \in S} a(x),$$

y el valor del flujo que llega a T es, al menos,

$$\sum_{x \in T} b(x).$$

Esto es,

$$\sum_{x \in T} b(x) \leq \sum_{(x,y) \in (S,T)} f(x,y) \leq \sum_{x \in S} a(x).$$

Diremos que la red es *balanceada* si cumple que las demandas del producto sean iguales a las ofertas, esto es, si satisface la condición:

$$\sum_{x \in T} b(x) = \sum_{x \in S} a(x).$$

Por otra parte, la ecuación de conservación (2.1) y la no-negatividad del flujo (2.2) nos lleva a las siguientes tres condiciones sobre las redes bipartitas completas con ofertas-demandas.

$$\sum_{x \in T} f(x,y) = a(y); \text{ para cada } y \in S, \quad (2.15)$$

$$\sum_{x \in S} f(x,y) = b(y); \text{ para cada } y \in T, \quad (2.16)$$

$$0 \leq f(x_p, x_j); \text{ para cada } (x_p, x_j) \in A. \quad (2.17)$$

De esta manera, el flujo f de S a T en (N, A) puede ser extendido a un flujo f_e de s_e a t_e en una red extendida (N_e, A_e) definiendo

$$f_e(s_e, x_i) = a(x_i); \text{ para cada } x_i \in S,$$

$$f_e(x_j, t_e) = b(x_j); \text{ para cada } x_j \in T,$$

$$f_e(x_i, x_j) = f(x_p, x_j); \text{ para cada } (x_p, x_j) \in (S, T).$$

En consecuencia podemos considerar cualquier problema de red bipartita completa balanceada con ofertas-demandas, como si tuviera una sola pareja fuente-sumidero.

En la literatura revisada se encuentra un algoritmo polinomial que resuelve el siguiente problema de diseño de redes, para el caso en que los flujos son enteros [Alvarez y Guardado, 1992]: "Dada una red bipartita completa balanceada (N, A) , se quiere encontrar el número mínimo de aristas, así como las magnitudes del flujo que deben asociarse a ellas, con la condición adicional de que dichos flujos no excedan cierto valor entero F_0 , determinado por un criterio minimax".

Considero de gran importancia el mencionar que algunos de los *vocablos* utilizados en la Teoría de Flujos en Redes, crean ambigüedad; por ejemplo, algunos autores se refieren a la palabra *arco* como sinónima a las palabras *arista*, *enlace*. Por lo cual, en nuestro contexto, el resultado de Alvarez y Guardado [1992] mencionado anteriormente es válido para *arcos*, en lugar de *aristas*.

El trabajo de Alvarez y Guardado [1992] se basa en un resultado previo de

Guardado [1982] en el cual se resuelve el siguiente problema de síntesis: calcular la mínima capacidad entera F_0 sobre los arcos de una red con flujos enteros. El valor F_0 ofrece uniformidad a los flujos sobre los arcos de la red, esto es, ofrece la capacidad mínima de tal forma que los flujos sobre los arcos de la red no difieran mucho. El tener flujos uniformes o balanceados sobre la red, asegura que la satisfacción de la demanda de un sumidero no depende mucho de una sola fuente.

Ahora bien, Alvarez y Guardado [1992] no demuestran la optimalidad del algoritmo que proponen para resolver el problema planteado. Por otra parte, no formulan un problema más general, como considerar flujos multiproducto o una red arbitraria, no necesariamente bipartita. En consecuencia este problema puede ser visto desde un punto de vista más general.

2.7 Definición del Problema de Investigación

Sea (N,A) una red uniproducto cualquiera con flujos enteros no negativos, donde $N=\{x_1, x_2, \dots, x_n\}$, $A=\{1,2, \dots, m\}$, fuente s y sumidero t . El problema de encontrar el número mínimo de arcos, así como las magnitudes del flujo que deben asociarse a ellos satisfaciendo que la oferta de la fuente s es igual a la demanda del sumidero t , con la condición adicional de que dichos flujos no excedan cierto valor entero F_0 en cada uno de los arcos, puede formularse de la siguiente manera.

Primero, obsérvese que para minimizar el número de arcos, debemos hacer que el flujo sea cero en el mayor número posible de arcos en la red, o equivalentemente, debemos permitir que el flujo sea positivo en el menor número posible de arcos.

Por otra parte, podemos observar que para cada $(x_r, x_j) \in A$ se tiene lo siguiente: si $f(x_r, x_j) > 0$ entonces $\lceil f(x_r, x_j)/F_0 \rceil = 1$, y si $f(x_r, x_j) = 0$ entonces $\lceil f(x_r, x_j)/F_0 \rceil = 0$, donde $\lceil x \rceil$ representa el menor entero mayor o igual que x . Por lo tanto, el **problema uniproducto P** de encontrar el número mínimo de arcos en una red lo podemos representar matemáticamente como:

$$\text{Min } \sum_{(x_r, x_j) \in A} \lceil f(x_r, x_j)/F_0 \rceil$$

s. a.

$$\sum_{x_j \in A(x_r)} f(x_r, x_j) - \sum_{x_r \in B(x_j)} f(x_r, x_j) = \begin{cases} b, & \text{si } x_r = s \\ 0, & \text{si } x_r \neq s, t \\ -b, & \text{si } x_r = t \end{cases} \quad (\text{P1})$$

$$0 \leq f(x_r, x_j) \text{ para cada } (x_r, x_j) \in A, \quad (\text{P2})$$

$$f(x_r, x_j) \leq F_0 \text{ para cada } (x_r, x_j) \in A, \quad (\text{P3})$$

donde b es la oferta (demanda) de la fuente (del sumidero) s (t).

Observemos que el problema uniproducto **P** pudiera ser considerado como un caso particular del problema del diseño óptimo de redes (2.11), seleccionando $\alpha_k = 1$, $\beta_k = 0$, cota F_0 y $\tau = 1$. Por lo tanto, el problema **P** no puede ser resuelto

exactamente por los algoritmos anteriormente mencionados, ya que el de Minoux [1989] es para el caso sin cota y el de Goemans [1994] es de aproximación. Esto es lo que favorece al trabajo de Alvarez y Guardado [1992].

Ahora bien, el problema **P** puede ser generalizado cuando definimos en la red (N,A) un flujo multiproducto de la siguiente forma. Supongamos que tenemos un flujo de T productos sobre la red con parejas fuente-sumidero (s^τ, t^τ) para $\tau=1, \dots, T$; asociadas con los T productos. Observemos que para cada $(x_p, x_j) \in A$ se tiene lo siguiente:

si $\sum_{\tau=1, \dots, T} f^\tau(x_p, x_j) > 0$ entonces

$$\left[\sum_{\tau=1, \dots, T} f^\tau(x_p, x_j) / \sum_{\tau=1, \dots, T} F_0^\tau \right] = 1$$

y si $\sum_{\tau=1, \dots, T} f^\tau(x_p, x_j) = 0$ entonces

$$\left[\sum_{\tau=1, \dots, T} f^\tau(x_p, x_j) / \sum_{\tau=1, \dots, T} F_0^\tau \right] = 0$$

Por lo tanto, el **problema multiproducto PP** de encontrar el número mínimo de arcos en una red lo podemos representar matemáticamente como:

$$\text{Min } \sum_{(x_p, x_j) \in A} \left[\sum_{\tau=1, \dots, T} f^\tau(x_p, x_j) / \sum_{\tau=1, \dots, T} F_0^\tau \right]$$

s. a.

$$\sum_{x_j \in A(x_p)} f^\tau(x_p, x_j) - \sum_{x \in B(x_p)} f^\tau(x, x_p) = \begin{cases} b^\tau, & \text{si } x = s^\tau \\ 0, & \text{si } x \neq s^\tau, t^\tau \\ -b^\tau, & \text{si } x = t^\tau \end{cases}$$

$$0 \leq f^\tau(x_p, x_j); \text{ para cada } (x_p, x_j) \in A$$

$$f^\tau(x_p, x_j) \leq F_0^\tau; \text{ para cada } (x_p, x_j) \in A,$$

donde b^τ es la oferta (o demanda) de la fuente (o sumidero) s^τ (o t^τ), para cada $\tau=1, \dots, T$.

En forma compacta podemos escribir el problema multiproducto **PP** de la siguiente forma:

$$\text{Min } \sum_{k=1, \dots, m} \left[\sum_{\tau=1, \dots, T} f_k^\tau / \sum_{\tau=1, \dots, T} F_0^\tau \right]$$

s. a.

$$E f^\tau = b^\tau \quad (\text{PP1})$$

$$0 \leq f^\tau \quad (\text{PP2})$$

$$f^\tau \leq F_0^\tau, \quad (\text{PP3})$$

donde E es la matriz de incidencia nodo-arco de (N, A) de dimensión $n \times m$, b^τ es el vector columna de dimensión m de componentes con valores b^τ , $-b^\tau$ en los lugares correspondientes a s^τ y t^τ respectivamente y cero en los restantes lugares; $F_0^\tau = (F_0^\tau, F_0^\tau, \dots, F_0^\tau)'$ de dimensión m y $f^\tau = (f_1^\tau, f_2^\tau, \dots, f_m^\tau)'$.

Los problemas uniproducto **P** y multiproducto **PP** no han sido abordados en la literatura en su forma general, por lo que es conveniente empezar a investigarlos. Se dispone de un algoritmo que resuelve el problema uniproducto **P** para redes bipartitas completas [Alvarez y Guardado, 1992], aunque en su forma original no se formula el problema de esta manera ni se demuestra la optimalidad.

El objetivo de este trabajo de tesis es *i*).- demostrar la optimalidad del algoritmo que resuelve el problema uniproducción **P** para el caso de redes bipartitas completas [Alvarez y Guardado, 1992], *ii*).- resolver el problema multiproducción **PP** para redes bipartitas completas, dejando la línea de investigación abierta para abordar los problemas más generales.

CAPITULO 3

UN ALGORITMO POLINOMIAL QUE MINIMIZA EL NUMERO DE ARCOS EN UNA RED BIPARTITA COMPLETA CON FLUJOS UNIPRODUCTO BALANCEADOS

3.1 Introducción

En el presente capítulo se analiza el siguiente problema de flujos uniproducto en redes bipartitas completas dirigidas: Determinar el número mínimo de arcos, así como las magnitudes del flujo que debe asociarse a ellos, para que se satisfagan las demandas de los sumideros, pero con la característica adicional de que dichos flujos no excedan cierta capacidad F_0 , determinada por un criterio minimax. El criterio minimax es el cálculo del mínimo entre todos los máximos flujos posibles que se les puede asignar a cada uno de los arcos.

La cota F_0 le imprime características especiales al problema, garantizando uniformidad en los flujos asociados a los arcos. Si bien dicha cota pudiera considerarse una capacidad de paso, los algoritmos clásicos de transporte con capacidades no son utilizables aquí, ya que su objetivo es minimizar costos y no el número de arcos. En el Apéndice A, se da un ejemplo donde la solución dada

por el algoritmo clásico de transporte a capacidades no contiene el número mínimo de arcos.

El problema que se propone es de gran aplicación práctica en problemas de síntesis de redes de comunicación y de transporte, por lo cual se considera importante desarrollar un algoritmo que encuentre su solución.

3.2 Formulación del Problema

Sea (N, A) una red bipartita completa dirigida donde N representa el conjunto de nodos y A el conjunto de arcos que enlazan dichos nodos. Sean $N = X^+ \cup X^-$, $|N| = n$, $|X^+| = m$, $X^+ \cap X^- = \phi$ y $A = X^+ \times X^-$; donde $|I|$ representa el número de elementos del conjunto I . Definamos en X^+ una función entera no negativa $a(x_i) = a_i$, que representa la magnitud de la oferta de cierto producto en la fuente x_i . Similarmente, definamos en X^- una función entera no negativa $b(x_j) = b_j$, que representa la magnitud de la demanda de dicho producto en el sumidero x_j . Supongamos que se satisface la ecuación de balance, esto es,

$$\sum_{x_i \in X^+} a(x_i) = \sum_{x_j \in X^-} b(x_j).$$

Definamos sobre A una función entera no negativa $f(x_i, x_j)$ (f_{ij} en el caso de saturación de subíndices) que representa el flujo de dicho producto desde el nodo x_i hasta el x_j de tal forma que se cumplan las ecuaciones de conservación de flujo. Esto es, en el contexto del presente trabajo, un flujo que satisface las ecuaciones

de conservación de flujo es uno que satisface las siguientes restricciones:

$$\sum_{x_j \in X^+} f(x_p, x_j) = a(x); \text{ para cada } x_p \in X^+ \quad (3.1)$$

$$\sum_{x_i \in X^-} f(x_i, x_j) = b(x); \text{ para cada } x_j \in X^- \quad (3.2)$$

$$0 \leq f(x_i, x_j) \text{ para cada } x_i \in X^+ \text{ y cada } x_j \in X^-. \quad (3.3)$$

En la literatura revisada [Guardado, 1982] está resuelto el problema de encontrar el valor mínimo F_0 de cierta función F de valores enteros, tal que para cada red (N, A_k) , $A_k \subseteq A$, que satisface (3.1)-(3.3), determina el máximo valor de flujo entre todos los arcos, es decir,

$$F_0 = \underset{\{(N, A_k)\}}{\text{Mínimo}} \{F(A_k)\}, \text{ donde } F(A_k) = \underset{(x_i, x_j) \in A_k}{\text{Máximo}} f(x_i, x_j), A_k \subseteq A.$$

El algoritmo que se utiliza para obtener el valor de F_0 puede verse en el Apéndice B y un ejemplo de su cálculo en el Capítulo 5.

Sean E la matriz de incidencia nodo-arco asociada a (N, A) , f el vector columna de flujos-arco de dimensión $|A|$ definido por $f \triangleq (f_{1,(m+1)}, f_{1,(m+2)}, \dots, f_{1,n}, f_{2,(m+1)}, f_{2,(m+2)}, \dots, f_{2,n}, \dots, f_{m,(m+1)}, f_{m,(m+2)}, \dots, f_{m,n})'$, b el vector columna de ofertas-demandas de dimensión n definido por $b \triangleq (a_1, a_2, \dots, a_m, -b_{m+1}, -b_{m+2}, \dots, -b_n)$. De esta forma las restricciones (3.1)-(3.3) las podemos escribir como

$$Ef = b \quad (3.4)$$

$$0 \leq f \quad (3.5)$$

o en forma compacta, podemos definir

$$\mathcal{F} = \{f : Ef = b, \mathbf{0} \leq f\}. \quad (3.6)$$

Así, el valor F_0 , lo podemos representar por

$$F_0 = \text{Mínimo} \{ \text{Máximo } f_y \}_{f \in \mathcal{F}}.$$

Observemos que de esta manera, $F: \mathbb{R}^{+(mn)} \rightarrow \mathbb{R}^+$ tal que para cada $f \in \mathcal{F}$,

$$F(f) = \text{Máximo } f(x_r, x_j) \text{ y } F_0 = \text{Mínimo } \{F(f)\}.$$

Ahora bien, sea

$$\mathcal{F}_0 = \{f : Ef = b, \mathbf{0} \leq f \leq F_0\}, \quad (3.7)$$

donde F_0 es el vector columna de dimensión $m(n-m)$ cuyas componentes son todas iguales al valor F_0 . Es fácil comprobar que $\mathcal{F}_0 \neq \emptyset$ (se verá después cómo la solución dada por el Algoritmo 1 es un elemento de \mathcal{F}_0). Se desea encontrar $f_0 \in \mathcal{F}_0$ tal que,

$$|f_0|^+ = \text{Mínimo} \{ |f|^+ \}_{f \in \mathcal{F}_0}$$

donde $|x|^+$ indica la cantidad de componentes positivas del vector x .

Podemos observar que a cada $f_k \in \mathcal{F}_0$ le podemos asociar una subred $(N, A_k) \subseteq (N, A)$ de tal forma que $(x_r, x_j) \in A_k$ si y solo si $f_k(x_r, x_j) > 0$, y viceversa. De esta manera, si encontramos el vector solución $f_0 \in \mathcal{F}_0$, encontramos la red (N, A_0) asociada a él que utiliza la menor cantidad de arcos, así como sus flujos correspondientes. La anterior red será denotada por $(N, A_0; F_0)$. La siguiente

proposición nos ofrece un contador del número de arcos con flujos positivos en una red.

PROPOSICION 3.1 Para cada $f \in \mathcal{F}_0$, la cantidad de flujos-arco $f(x_p, x_j) > 0$ está

dada por la suma $\sum_{f(x_p, x_j) \in f} \lceil f(x_p, x_j) / F_0 \rceil$.

DEM: Para cada $f(x_p, x_j) \in f$ se tiene que $f(x_p, x_j) \leq F_0$, esto es, $f(x_p, x_j) / F_0 \leq 1$. Por lo

tanto, para cada $f(x_p, x_j) \in f$ se tiene lo siguiente: si $f(x_p, x_j) > 0$, $\lceil f(x_p, x_j) / F_0 \rceil = 1$ y si

$f(x_p, x_j) = 0$, $\lceil f(x_p, x_j) / F_0 \rceil = 0$. ■

En consecuencia, nuestro problema consiste en encontrar $f_0 \in \mathcal{F}_0$ donde

$$|f_0|^+ = \min \left\{ \sum_{f(x_p, x_j) \in f} \lceil f(x_p, x_j) / F_0 \rceil \right\}. \quad (3.8)$$

En otras palabras, el problema consiste en minimizar la función de valores enteros (3.8) y que por comodidad llamaremos **problema uniproducto P**.

3.3 Solución al Problema Uniproducto

3.3.1 Definiciones Básicas.

Primeramente formalizaremos un grupo de definiciones que serán utilizadas frecuentemente en el resto del trabajo.

Una *solución factible* para el problema **P** es cualquier red que satisfaga las

condiciones del problema, sin ser necesariamente la óptima. Esto es, cualquier $f \in \mathcal{F}_0$ puede considerarse una solución factible del problema P. Llamemos $(N, A_f; F_0)$ la red asociada a f .

Una *matriz de flujos* M , correspondiente al problema P y a un flujo f , es la matriz cuya celda (i, j) es f_{ij} . Diremos que una celda (i, j) de M está *ocupada* si $f_{ij} \neq 0$ y que está *libre* si $f_{ij} = 0$. A la cantidad de celdas ocupadas de M la llamaremos *cardinalidad* y la denotaremos por $|M|^+$. Por consiguiente, si M es la matriz de flujos correspondiente a una solución factible f de P, se tiene que $|M|^+ = |f|^+$.

Un *ciclo* en M es cualquier sucesión de celdas de la forma $(i_1, j_1); (i_2, j_1); (i_2, j_2); \dots; (i_p, j_p); (i_1, j_p)$, y es la representación sobre la matriz M del concepto de ciclo de una red (ver Figura 1). Cada ciclo deberá contener una cantidad par de celdas mayor o igual que cuatro. Una *celda impar* de un ciclo es una celda que ocupa una posición impar, y una *par* es una que ocupa una posición par, en la sucesión que define el ciclo.

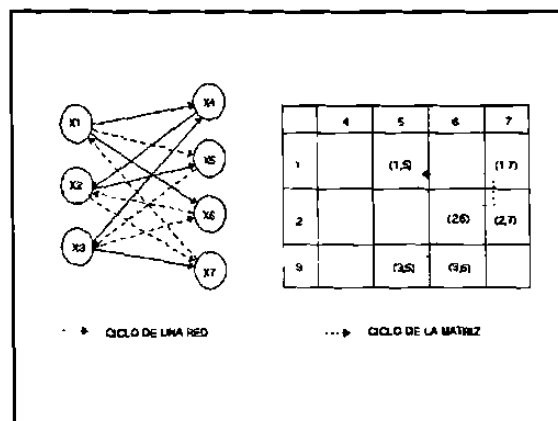


Figura 1. Representación gráfica de un ciclo en una red y en una matriz.

Sea δ un entero tal que $0 < \delta \leq F_0$, definimos un δ -ciclo de redistribución como un ciclo sobre el que se realiza una redistribución de cierta magnitud δ de flujo, restando δ al flujo de las celdas impares y sumando δ al flujo de las celdas pares. Nótese que en la fila y columna donde haya celdas que están en un δ -ciclo, deben existir igual cantidad de celdas pares que celdas impares para que se satisfaga (3.6). Para una ilustración de ésta definición ver la Figura 2.

		DEMANDAS				
		21	15	11	10	8
O F E R T A S	27	6 ⁽⁻²⁾	6	6	6	3 ⁽⁺²⁾
	19	6	6	5	2	0
	14	6	3	0	2	3
	5	3 ⁽⁺²⁾	0	0	0	2 ⁽⁻²⁾

Figura 2. δ -ciclo de redistribución, donde $\delta=2$ y $F_0=6$

3.3.2 Determinación de la Solución al Problema Uniproducto.

No es posible conocer a *priori* el valor mínimo de (3.8), sin embargo, como se dispone de una cota inferior [Guardado, 1982]

$$K_0 = \max \left\{ \sum_{x_i \in X^*} \lceil a(x_i) / F_0 \rceil, \sum_{x_j \in X^*} \lceil b(x_j) / F_0 \rceil \right\},$$

el proceso de búsqueda de la solución de **P** se hará en forma iterativa. Sin pérdida de generalidad podemos suponer que

$$K_0 = \sum_{x_i \in X^*} \lceil a(x_i) / F_0 \rceil,$$

de no ser así, intercambiamos las ofertas por las demandas y después de resolver el problema volvemos a la situación inicial y cambiamos la orientación de los arcos encontrados.

La solución inicial de la cual partiremos puede obtenerse a partir del siguiente algoritmo [Alvarez y Guardado, 1992], el cual llena por filas una matriz M_1 que es asociada a una solución inicial $f^* \in \mathcal{F}_0$. El algoritmo es finito y de complejidad polinomial $O(mn)$.

ALGORITMO 1.

Consideremos que $a(x_i)$, $b(x_j)$, están ordenados en forma no-creciente para cada $i=1, \dots, m$; $j=m+1, \dots, n$.

PASO 1. Hacer

$$a^{m+1}(x_i) = a(x_i), \text{ para cada } i=1, \dots, m.$$

$$b^1(x_j) = b(x_j), \text{ para cada } j=m+1, \dots, n.$$

PASO 2. Calcular

$$f_y = \min \{ F_0 ; a^i(x_i) ; b^j(x_j) \}, \quad i=1, \dots, m ; j=m+1, \dots, n.$$

$$a^{i+1}(x_i) = a^i(x_i) - f_y,$$

$$b^{j+1}(x_j) = b^j(x_j) - f_y,$$

donde $b^j(x_j)$ denota la demanda que queda en el sumidero x_j ($j=m+1, \dots, n$)

después de la construcción de la fila $(i-1)$ de M_I para $i=2,\dots,m$ y $\alpha(x_i)$ la oferta que queda en la fuente x_i ($i=1,\dots,m$) después de hallar el elemento $(j-1)$ de la fila i con $j=m+2,\dots,n$.



Observe que se tienen dos únicos casos: 1) Si $|M_I|^+ = K_0$ entonces tomamos $f_0 = f^*$ y el problema está resuelto. 2) En caso de que $|M_I|^+ > K_0$ la estrategia a seguir será la disminución, de ser posible, del número de celdas ocupadas de M_I mediante δ -ciclos de redistribución que garanticen que el valor del flujo de cualquier celda del ciclo no exceda el valor de F_0 . El criterio de parada se dará por la no existencia de tales ciclos.

Sea M_{Ik} la matriz de flujos obtenida después de componer k δ -ciclos de redistribución sobre M_I . Los elementos de la matriz M_{Ik} deberán satisfacer las condiciones de $f \in \mathcal{F}_0$ y $|M_{Ik}|^+ \leq |M_I|^+$. Para garantizar lo anterior los δ -ciclos de redistribución deberán tener las siguientes características antes de realizar la distribución del flujo δ :

- A1) En las celdas pares, sumarle δ al flujo existente no debe exceder la magnitud F_0 .
- A2) En las celdas impares, restarle δ al flujo existente no debe ser negativo.
- A3) La cantidad de celdas pares con valor de flujo igual a cero debe ser menor o igual que la cantidad de celdas impares con valor de flujo igual a δ .

En lo que sigue, quedará implícito que un δ -ciclo satisface A1, A2 y A3. Si en A3 se cumple la desigualdad estricta el δ -ciclo recibe el nombre de *δ -ciclo disminuidor de celdas ocupadas* (δ CDCO), si se cumple la igualdad se denominará *δ -ciclo parcial* (δ CP). De lo anterior se sigue que la realización de un δ CP mantiene el número de celdas ocupadas en la matriz M_1 , mientras que un δ CDCO lo disminuye.

3.3.2.1 Búsqueda de δ -ciclos.

En dependencia del δ seleccionado y del valor de F_0 las celdas de M_1 pueden clasificarse y marcarse unívocamente de la siguiente forma:

celdas "+" : $\{(i,j) : f_y + \delta \leq F_0 \text{ y } f_y - \delta < 0\}$.

celdas "-" : $\{(i,j) : f_y + \delta > F_0 \text{ y } f_y - \delta \geq 0\}$.

celdas "±" : $\{(i,j) : f_y + \delta \leq F_0 \text{ y } f_y - \delta \geq 0\}$.

celdas "@" : $\{(i,j) : f_y + \delta > F_0 \text{ y } f_y - \delta < 0\}$.

Distinguiremos con la marca adicional "&" a las celdas "+" tales que $f_y = 0$ y con "*" a las celdas "-" o "±" tales que $f_y - \delta = 0$. De lo anterior se tiene que, para un valor δ dado, las celdas "+" solamente podrán ser celdas pares en los δ -ciclos; las celdas "-", celdas impares; las celdas "±" pueden ser ambas cosas y las celdas "@" no pueden pertenecer a ningún δ -ciclo.

Llamaremos *δ -tabla* a la tabla que se obtiene a partir de M_1 al colocar en cada celda $(i,j) \in M_1$, la marca correspondiente (ver Figura 3). Teniendo en cuenta las características de los δ -ciclos resulta conveniente adoptar, en cada δ -tabla, las

siguientes reglas de simplificación:

- 1).- Si alguna fila (columna) de la tabla contiene exclusivamente celdas "+" y "@" o celdas "-" y "@", entonces puede ser extraída de la tabla, pues en cualquiera de los casos, ninguna celda puede formar parte de un δ -ciclo.
- 2).- Si alguna fila (columna) de la tabla contiene exclusivamente celdas "-" (sin considerar las celdas "@") y una única celda "±" en cuya correspondiente columna (fila) solamente existen celdas "+", dicha fila y columna podrán ser extraídas de la tabla.
- 3).- Cuando una fila (columna) sea extraída de la tabla no se alterará el número de orden de las restantes y será reincorporada a la tabla al variar el valor de δ .

		DEMANDAS				
		21	15	11	10	8
O F E R T A S	27	-	-	-	-	+ -
	19	-	-	-	*	&
	14	-	+ -	&	*	+ -
	5	+ -	&	&	&	*

Figura 3. δ -tabla asociada a la matriz de distribución de la Figura 2, donde $\delta=2$.

Estas reglas se aplicarán sucesivamente mientras sea posible y como resultado obtendremos una δ -tabla transformada que denotaremos por $D(\delta)$ con la cual trabajaremos en adelante. Observe que si en $D(\delta)$ no existe alguna celda marcada

con "*" podemos asegurar que para ese valor δ no existe δ CDCO.

Supongamos entonces que en $D(\delta)$ existe al menos una celda con la marca adicional "*". Pongamos en correspondencia con $D(\delta)$ un grafo dirigido y marcado $b=[\mathcal{X}^+, \mathcal{X}, A]$ de la siguiente forma:

- El conjunto \mathcal{X}^+ lo formaremos con los nodos fuentes asociados a las filas de $D(\delta)$ y el conjunto \mathcal{X} con los nodos sumideros asociados a las columnas.
- A cada celda $(i,j) \in D(\delta)$ con marca "+" le corresponderá el arco (x_p, x_j) ; $x_i \in \mathcal{X}^+$, $x_j \in \mathcal{X}$.
- A cada celda $(i,j) \in D(\delta)$ con marca "-" le corresponderá el arco (x_p, x_i) ; $x_j \in \mathcal{X}$, $x_i \in \mathcal{X}^+$.
- A cada celda $(i,j) \in D(\delta)$ con marca "±" le corresponderán los arcos (x_p, x_j) y (x_p, x_i) ; $x_i \in \mathcal{X}^+$, $x_j \in \mathcal{X}$.
- A las celdas "@" de $D(\delta)$ no se les asociará algún arco de b .
- Si la celda $(i,j) \in D(\delta)$ posee la marca "&" o "*" dicha marca será trasladada al arco correspondiente.

No es difícil verificar que hay una correspondencia biunívoca entre b y $D(\delta)$.

Dada la forma en que fue definida la dirección del arco (x_p, x_j) en b en correspondencia con la marca de la celda (i,j) , no es difícil notar que a cualquier δ -ciclo de la matriz M_I le corresponde en el grafo b un circuito con una cantidad par de arcos no menor que cuatro.

De acuerdo a como fueron asignadas las marcas adicionales "&" y "*" y tomando en cuenta su significado, a un δ CDCO le corresponderá en b un circuito en el cual, la cantidad de arcos con marca "*" es mayor que la cantidad de arcos con marca "&". Este será llamado *circuito disminuidor*. A un δ CP le corresponderá en b un circuito en el cual el número de celdas con marca "*" es igual al número de celdas con marca "&". Este será llamado *circuito parcial*.

De esta forma, el problema de búsqueda de los δ -ciclos puede realizarse a través de la búsqueda de tales circuitos en b . Esta se realizará sobre la base del procedimiento Búsqueda en Profundidad en grafos dirigidos [Szwarcfiter, 1984]. Por lo tanto, la búsqueda de los δ -ciclos que conduzcan, si es posible, a disminuir la cardinalidad de una matriz de flujos puede resumirse en el siguiente procedimiento llamado *búsqueda de δ -ciclos* y que denotamos por B- δ C.

PROCEDIMIENTO: BUSQUEDA DE δ -CICLOS (B- δ C).

Dados una matriz M , un determinado valor de δ y el índice k de la fila de M que se analiza.

PASO 1. Construir la δ -tabla. Formar la lista L_k de las celdas con marca "*" en la fila k . Hallar $D(\delta)$. Si la fila k fue eliminada ir al PASO 3. En caso contrario construir b .

PASO 2. Seleccionar una celda (k,l) con marca "*" no analizada. Si no existe ir al PASO 3. En caso afirmativo, buscar una trayectoria de x_k a x_l que no contenga el arco (x_k, x_l) , mediante una búsqueda en profundidad con raíz x_k . Si tal trayectoria

no existe reiniciar el PASO 2. Si existe trayectoria, verificar si el circuito formado por dicha trayectoria y el arco (x_i, x_k) constituye un circuito disminuidor. En caso afirmativo el procedimiento concluye redistribuyendo δ unidades de flujo a través del δ CDCO hallado. En caso de que el circuito no constituya un circuito disminuidor se busca otra trayectoria de x_k a x_i y se repite el análisis. De no existir trayectoria reiniciar el PASO 2.

PASO 3. Construir $D(1)$. Si la fila k no se encuentra en la tabla entonces no existe un 1-CP y el procedimiento concluye. En caso afirmativo, construir el grafo asociado a la tabla $D(1)$, b^1 , e ir al PASO 4.

PASO 4. Seleccionar una celda $(k,s) \in L_k$ y buscar una trayectoria de x_k a x_s , mediante una búsqueda en profundidad, que junto al arco (x_s, x_k) forme un circuito parcial. Si no existe un circuito parcial, eliminar esa celda de L_k . Si $L_k = \emptyset$, no existe 1-CP y el procedimiento concluye. Si $L_k \neq \emptyset$ reiniciar el PASO 4. En caso de existir circuito parcial se realiza la redistribución de θ unidades de flujo a través del mismo, donde

$$\theta = \min \left\{ F_0 - \max_{(i,j) \text{ par}} \{f_{ij}\}, \min_{(i,j) \text{ impar}} \{f_{ij}\} \right\}.$$

Hacer $\delta = \delta - \theta$ y retornar al PASO 1.



La complejidad de este procedimiento es $O(mn)$ la cual es determinada por la complejidad de la búsqueda en profundidad [Szwarcfiter, 1984]. La finitud del

mismo se garantiza por el hecho de que en una fila la cantidad de celdas marcadas con "*" es finita al igual que el número de circuitos simples en un grafo. Este procedimiento se utilizará reiteradamente y constituye la base del algoritmo general de diseño que expondremos a continuación.

3.3.2.2 Algoritmo General de Diseño (AGD).

ALGORITMO

Dados el valor F_0 y la matriz de flujos M_i asociada a un $f^* \in \mathcal{F}_0$ obtenida mediante el criterio de distribución del Algoritmo 1.

PASO i ($i = m, \dots, 1$):

NIVEL 1. Sea $P_i = \{f_y > 0, j = m+1, \dots, n\}$. Si $|P_i| = \lceil a_i / F_0 \rceil$ reiniciar el paso para el siguiente valor de i . De otra forma hacer $\delta = (\min\{f_y \in P_i\} \neq F_0)$ y pasar al NIVEL 2.

NIVEL 2. Si $\delta \in P_i$ ejecutar procedimiento B- δ C. Si $\delta \notin P_i$, hacer $\delta = \delta + 1$, si $\delta < F_0$ recomenzar este nivel, de lo contrario reiniciar el paso para el siguiente valor de i .

NIVEL 3. Si existe δ CDCO reiniciar el paso con el mismo valor de i . En otro caso hacer $\delta = \delta + 1$. Si $\delta < F_0$ formar el conjunto P_i y recomenzar el NIVEL 2. De lo contrario reiniciar el paso para el siguiente valor de i .



La finitud del algoritmo AGD se garantiza porque ejecuta un procedimiento

finito (B- δ C) a lo sumo $(n-m)$ veces para valores de δ menores a F_θ . Para el análisis de la complejidad veamos la siguiente proposición.

PROPOSICION 3.2 El algoritmo AGD es de complejidad polinomial $O(m^2n^2)$.

DEM: En cada fila $i=m, \dots, 1$, el algoritmo AGD ejecuta el procedimiento B- δ C a lo sumo $(n-m)$ veces para valores de δ menores a F_θ . Además, el procedimiento B- δ C es de complejidad $O(mn)$. Por lo tanto, la complejidad de AGD es del orden $O(m(n-m)mn)$, esto es, del orden $O(m^2n^2)$.



3.4 Optimalidad del Algoritmo AGD

Primero observemos que cualquier δ -ciclo puede "descomponerse" como una secuencia de δ 1-ciclos, ya que si encontramos un δ -ciclo, pudimos haber encontrado δ veces un 1-ciclo. Luego, para ver que efectivamente el algoritmo AGD resuelve el problema P, demostraremos el siguiente teorema.

TEOREMA 3.3 Sean $A, B \in \mathbb{Z}_+^{m \times n}$, donde \mathbb{Z}_+ es el conjunto de enteros no negativos, tales que

$$(1) \quad 0 \leq a_{ij}, b_{ij} \leq K \quad \text{para cada } i=1, \dots, m, \text{ y cada } j=1, \dots, n;$$

$$(2) \quad \sum_{j=1, \dots, n} a_{ij} = \sum_{j=1, \dots, n} b_{ij} \quad \text{para cada } i=1, \dots, m;$$

$$(3) \quad \sum_{i=1, \dots, m} a_{ij} = \sum_{i=1, \dots, m} b_{ij} \quad \text{para cada } j=1, \dots, n,$$

donde $K \in \mathbb{Z}_+$. Entonces existe un conjunto finito de 1-ciclos de tal forma que A puede ser transformada en B .

DEM: Para transformar A en B procedamos como sigue.

1).- Si $A = B$ la demostración termina. Si $A \neq B$, fijemos las celdas (p,q) de A tales que $a_{pq} = b_{pq}$.

2).- Sea (k,l) una celda de A tal que $b_{kl} > a_{kl} \geq 0$.

3).- Fijemos la celda (k,l) .

Mientras $b_{kl} > a_{kl} \geq 0$ efectuemos el siguiente procedimiento de búsqueda de 1-ciclos:

Como $b_{kl} > a_{kl} \geq 0$, entonces existe un $1 \leq l_1 \leq n$, $l_1 \neq l$ tal que

$$0 \leq b_{kl_1} < a_{kl_1} \leq K,$$

esto es, $b_{kl_1} < K$. Por lo tanto, restemos 1 a b_{kl} y sumemos 1 a b_{kl_1} .

Luego, existe al menos un valor $1 \leq k_1 \leq m$, $k_1 \neq k$ tal que

$$0 \leq a_{k_1l_1} < b_{k_1l_1} \leq K,$$

esto es $0 < b_{k_1l_1}$. Por lo tanto, restemos 1 a $b_{k_1l_1}$.

Luego, existe un $1 \leq l_2 \leq n$, $l_2 \neq l_1$ tal que

$$0 \leq b_{kl_2} < a_{kl_2} \leq K,$$

esto es, $b_{kl_2} < K$. Por lo tanto, sumemos 1 a b_{kl_2} .

De esta forma eventualmente se alcanza la columna l , pues en esta columna se cumple que existe un $k_r \neq k$ tal que $0 \leq b_{k_r l} < a_{k_r l} \leq K$, esto es, $b_{k_r l} < K$.

Por lo tanto, sumemos 1 a $b_{k_r l}$. Con esto encontramos el 1-ciclo buscado y finaliza el procedimiento.

4).- Como $b_{kl} = a_{kl}$, nuevamente, vayamos a 1).

Esto termina la demostración.



TEOREMA 3.4 (Teorema de la Optimalidad de AGD).

El algoritmo AGD resuelve el problema P.

DEM: Supongamos que existe un $f_l \in \mathcal{F}_0$ tal que $|f_l|^+ < |f_0|^+$. Sean M_0 y M_l las matrices asociadas a f_0 y a f_l , respectivamente, las cuales satisfacen las condiciones del teorema 3. Por lo tanto, existe un conjunto de 1-ciclos tal que M_0 puede ser transformada en M_l , contradiciendo la no existencia de δ -ciclos al aplicar el algoritmo AGD. Por lo tanto, $|f_l|^+ \geq |f_0|^+$, para cada $f_l \in \mathcal{F}_0$.



En resumen, la obtención de un vector $f_0 \in \mathcal{F}_0$ solución del problema P se realizará según la siguiente secuencia:

ETAPA 1. Determinación de la matriz M_l correspondiente al vector inicial $f^* \in \mathcal{F}_0$ obtenido mediante el criterio de distribución dado por el Algoritmo 1.

ETAPA 2. Aplicación de AGD.

El resultado de la misma es un f_0 óptimo que constituye la solución del problema P. En consecuencia encontramos una red óptima $(N, A_0; F_0)$ que contiene flujos balanceados y con número mínimo de arcos.

3.5 Optimalidad Consistente del Algoritmo AGD

La solución óptima encontrada por el algoritmo AGD no necesariamente es única, ya que depende del f^* inicial y de la búsqueda en profundidad. En el Apéndice 3 podemos ver dos soluciones a un mismo problema. En particular, AGD toma como solución inicial la que encuentra el Algoritmo 1, pero pudiera tomarse una encontrada por algún otro método. El siguiente teorema asegura que se puede encontrar un conjunto de 1-ciclos que transformen una solución en otra.

TEOREMA 3.5 Sea M_θ la matriz asociada a la solución $f_\theta = (f_\theta^0)$ dada por el algoritmo AGD. Sea $f_1 = (f_1^l)$ cualquier otra solución con M_1 como matriz asociada, entonces existe una secuencia de 1-ciclos de tal forma que M_1 puede ser transformada en M_θ .

DEM: Aplicar el Teorema 3.3 a M_1 y M_θ .



Puede observarse que en la aplicación del Teorema 3.3 para demostrar el teorema 3.5, si M_p es una matriz intermedia en la transformación de M_θ a M_1 no necesariamente se tiene que $|M_\theta|^+ = |M_1|^+ \neq M_p^+$

3.6 Resumen

Debido a que los algoritmos existentes en la literatura no eran utilizables en el problema formulado en este trabajo, fue necesario diseñar para el mismo, algoritmos propios de solución. Dado que no es posible determinar *a priori* el

número de arcos que tendrá la red óptima, y que se conoce una cota inferior para este número, la estrategia que se siguió fue proceder de forma iterativa a partir de una solución inicial al problema. Se fue disminuyendo el número de arcos utilizados, siempre que fuera posible, obteniendo de esta forma soluciones cada vez mejores.

Para lograr lo anterior se diseñó el procedimiento B- δ C, el cual realiza la búsqueda de ciclos que anulen el flujo de algún arco sin que se violen las restricciones del problema. Este procedimiento descansa sobre la base de la Búsqueda en Profundidad en grafos dirigidos. El algoritmo AGD, también diseñado aquí, se encarga de decidir cuándo debe ser utilizado dicho procedimiento y su entrada es la matriz de flujos de la solución inicial de que se dispone, asociada al vector inicial f^* , así como el valor F_0 que acota los flujos. Todos los algoritmos y procedimientos creados son finitos y de complejidad polinomial.

Por otra parte, se demostró un teorema que afirma que una matriz de flujos correspondiente a un problema **P**, puede transformarse en otra matriz de flujos correspondiente al mismo problema. Con esto se garantiza la consistencia del algoritmo AGD, en el sentido de que dada cualquier solución del problema **P** existe un conjunto de 1-ciclos que pueden transformarla en la solución obtenida por el algoritmo AGD.

CAPITULO 4

UN ALGORITMO POLINOMIAL QUE MINIMIZA EL NUMERO DE ARCOS EN UNA RED BIPARTITA COMPLETA CON FLUJOS MULTIPRODUCTO BALANCEADOS

4.1 Introducción

En este capítulo se investiga y documenta un problema de flujos multiproducto sobre redes bipartitas completas. El objetivo es determinar el número mínimo de arcos, así como las magnitudes de flujo de cada producto que deben asociarse a ellos, para que se satisfagan las demandas de los sumideros. Adicionalmente, en cada arco el valor del flujo para cada producto t está acotado por un valor F_0^t determinado por un criterio mini-max. La cota F_0^t garantiza uniformidad en los flujos correspondientes al producto t .

4.2 Formulación del Problema

Sea (N,A) una red bipartita completa dirigida donde N es el conjunto de nodos y A el conjunto de arcos que enlazan dichos nodos. Sean $N^+ = \{x_1, x_2, \dots, x_m\}$ y $N^- = \{x_{m+1}, \dots, x_n\}$, $X^+ = \{x_1, x_2, \dots, x_m\}$, $X^- = \{x_{m+1}, \dots, x_n\}$, luego $N = X^+ \cup X^-$, $N^+ = m$, $N^- = n$.

$X^+ \cap X^- = \emptyset$ y $A = X^+ \times X^-$. Asociemos con cada $x_i \in X^+$ un vector fila de dimensión T , denotado por $\mathbf{a}(x_i) \triangleq \mathbf{a}_i = (a_i^1, \dots, a_i^\tau, \dots, a_i^T)$, cuya τ -ésima componente entera no-negativa representa la magnitud de la oferta del producto τ en la fuente x_i , para cada $\tau = 1, \dots, T$. Similarmente, asociemos con cada $x_j \in X^-$ un vector fila de dimensión T , denotado por $\mathbf{b}(x_j) \triangleq \mathbf{b}_j = (b_j^1, \dots, b_j^\tau, \dots, b_j^T)$, cuya τ -ésima componente entera no-negativa, representa la magnitud de la demanda del producto τ en el sumidero x_j , para cada $\tau = 1, \dots, T$.

Supongamos que la red está balanceada, esto es, que se satisface la siguiente ecuación, llamada *ecuación de balance*,

$$\sum_{x_i \in X^+} \mathbf{a}(x_i) = \sum_{x_j \in X^-} \mathbf{b}(x_j),$$

y denotemos $\mathbf{a}^\tau = (a_{i_1}^\tau, \dots, a_{i_m}^\tau)'$ y $\mathbf{b}^\tau = (b_{m+1}^\tau, \dots, b_n^\tau)'$, donde $a_{i_p}^\tau, b_{j_q}^\tau$ representan las ofertas y demandas del producto τ en los nodos x_{i_p}, x_{j_q} .

Definamos sobre A un flujo multiproducto $f: A \rightarrow \mathbb{R}^{+T}$, tal que a cada $(x_p, x_q) \in A$ le asignamos el vector fila de flujos de dimensión T ,

$$f(x_p, x_q) \triangleq \mathbf{f}_{pq} = (f_{pq}^1(x_p, x_q), \dots, f_{pq}^\tau(x_p, x_q), \dots, f_{pq}^T(x_p, x_q)) = (f_{pq}^1, \dots, f_{pq}^\tau, \dots, f_{pq}^T)$$

cuya τ -ésima componente entera no-negativa f_{pq}^τ representa el flujo del producto τ desde el nodo fuente x_p hasta el nodo sumidero x_q , luego $\mathbf{f} = (\mathbf{f}_{1,(m+1)}, \mathbf{f}_{1,(m+2)}, \dots, \mathbf{f}_{1,n}, \mathbf{f}_{2,(m+1)}, \mathbf{f}_{2,(m+2)}, \dots, \mathbf{f}_{2,n}, \dots, \mathbf{f}_{m,(m+1)}, \mathbf{f}_{m,(m+2)}, \dots, \mathbf{f}_{m,n})'$. Supongamos que se cumplan las ecuaciones de conservación de flujo. En nuestro contexto, esto significa que el flujo satisface:

$$\sum_{x_j \in X^-} f(x_i, x_j) = a(x_i); \text{ para cada } x_i \in X^+ \quad (4.1)$$

$$\sum_{x_i \in X^+} f(x_i, x_j) = b(x_j); \text{ para cada } x_j \in X^- \quad (4.2)$$

$$\mathbf{0} \leq f(x_i, x_j); \text{ para cada } x_i \in X^+ \text{ y cada } x_j \in X^-. \quad (4.3)$$

donde $\mathbf{0}^T$ es el vector fila nulo de \mathbb{R}^T .

Sea f^τ el vector columna de flujos del producto τ asociados a los arcos de la red, esto es, $f^\tau \triangleq (f_{1,(m+1)}^\tau, f_{1,(m+2)}^\tau, \dots, f_{1,n}^\tau, f_{2,(m+1)}^\tau, f_{2,(m+2)}^\tau, \dots, f_{2,n}^\tau, \dots, f_{m,(m+1)}^\tau, f_{m,(m+2)}^\tau, \dots, f_{m,n}^\tau)'$, luego $f = (f^1, f^2, \dots, f^T)$. Sea E la matriz de incidencia nodo-arco asociada a (N, A) , la cual es de dimensión $n \times m(n-m)$. Definamos el vector d^τ como el vector columna oferta-demanda de dimensión n por $d^\tau \triangleq (a_1^\tau, a_2^\tau, \dots, a_m^\tau, -b_{m+1}^\tau, -b_{m+2}^\tau, \dots, -b_n^\tau)'$. De esta forma las restricciones (4.1)-(4.3), las podemos escribir como

$$Ef^\tau = d^\tau, \text{ para cada } \tau, \quad (4.4)$$

$$\mathbf{0} \leq f^\tau, \text{ para cada } \tau, \quad (4.5)$$

donde $\mathbf{0}$ es el vector columna nulo de $\mathbb{R}^{m(n-m)}$. En forma más compacta las restricciones (4.4) y (4.5) pueden escribirse como,

$$Ef = d, \quad (4.6)$$

$$\mathbf{0} \leq f, \quad (4.7)$$

donde $f = (f^1, f^2, \dots, f^T)$, $d = (d^1, d^2, \dots, d^T)$ y $\mathbf{0}$ es la matriz nula de dimensión $m(n-m) \times T$. Más aún, denotemos particularmente el siguiente conjunto

$$\mathcal{F} = \{f : Ef = d, \mathbf{0} \leq f\}. \quad (4.8)$$

Como vimos en el capítulo anterior, para cada producto τ podemos encontrar el valor mínimo F^τ_θ que determina el máximo valor de flujo del producto τ entre todos los arcos. Definamos F^τ_θ el vector columna de dimensión $m(n-m)$ cuyas componentes todas son iguales al valor F^τ_θ . Por lo tanto, dada una red T -producto podemos encontrar un T -vector $F_\theta = (F^1_\theta, F^2_\theta, \dots, F^T_\theta)$.

Ahora bien, sea

$$\mathcal{F}_\theta = \{f : Ef = d, \mathbf{0} \leq f \leq F_\theta\}. \quad (4.9)$$

Nuestro primer objetivo es demostrar que \mathcal{F}_θ es no-vacío. Esto lo demuestra la siguiente proposición.

PROPOSICION 4.1 \mathcal{F}_θ es no-vacío.

Dem: Aplicando el algoritmo AGD del capítulo anterior podemos encontrar un f^τ_θ para cada τ , que satisface

$$Ef^\tau_\theta = d^\tau,$$

$$\mathbf{0} \leq f^\tau_\theta \leq F^\tau_\theta.$$

Tomemos $f^* = (f^1_\theta, \dots, f^\tau_\theta, \dots, f^T_\theta)$, obviamente $f^* \in \mathcal{F}_\theta$



Nuestro problema consiste en encontrar $f_\theta \in \mathcal{F}_\theta$ tal que,

$$f_\theta^* = \underset{f \in \mathcal{F}_\theta}{\text{Mínimo}} \{ f^* \}$$

donde, $f = (f_{1,(m+1)}, f_{1,(m+2)}, \dots, f_{1,n}, f_{2,(m+1)}, f_{2,(m+2)}, \dots, f_{2,n}, \dots, f_{m,(m+1)}, f_{m,(m+2)}, \dots, f_{m,n})'$,

$f_\theta = (f^{(\theta)}_{1,(m+1)}, f^{(\theta)}_{1,(m+2)}, \dots, f^{(\theta)}_{1,n}, f^{(\theta)}_{2,(m+1)}, f^{(\theta)}_{2,(m+2)}, \dots, f^{(\theta)}_{2,n}, \dots, f^{(\theta)}_{m,(m+1)}, f^{(\theta)}_{m,(m+2)})'$

$\dots, f^{(0)}_{m,n})'$ y x^+ indica la cantidad de componentes positivas del vector x . Nótese que en este caso las componentes son vectores. Equivalentemente, lo que estamos buscando es la red asociada a f_θ , la cual tiene el menor número posible de arcos con flujo positivo, o bien el mayor número posible de arcos con flujo cero.

Podemos observar que a cada $f_k \in \mathcal{F}_\theta$ le podemos asociar una subred $(N, A_k) \subseteq (N, A)$ de tal forma que $(x_r, x_j) \in A_k$ si y solo si $f^{(k)}_{ij} > 0$, y viceversa. De esta manera, si encontramos el vector solución $f_\theta \in \mathcal{F}_\theta$, encontramos la red (N, A_θ) asociada a él que utiliza la menor cantidad de arcos, así como sus flujos correspondientes. La anterior red será denotada por $(N, A_\theta; F_\theta)$. La siguiente proposición nos ofrece un contador del número de arcos con vectores de flujo positivos en una red.

PROPOSICION 4.2 Para cada $f \in \mathcal{F}_\theta$ la cantidad de componentes flujo-arco $f_{ij} > 0$

está dado por la suma

$$\sum_{f_\theta \in f} \left[\left\{ \sum_{\tau=1, \dots, T} f_{ij}^\tau \right\} / \left\{ \sum_{\tau=1, \dots, T} F_{ij}^\tau \right\} \right].$$

DEM: Para cada i, j y τ se tiene que $f_{ij}^\tau \leq F_{ij}^\tau$, de donde se tiene que

$$\sum_{\tau=1, \dots, T} f_{ij}^\tau \leq \sum_{\tau=1, \dots, T} F_{ij}^\tau$$

entonces

$$\left\{ \left(\sum_{\tau=1, \dots, T} f_{ij}^\tau \right) / \left(\sum_{\tau=1, \dots, T} F_{ij}^\tau \right) \right\} \leq 1$$

por lo tanto, para cada $f_y \in f$ se tiene lo siguiente.

Si $f_y^\tau > 0$ para algún τ , entonces

$$\left[\left\{ \frac{\sum_{\tau=1, \dots, T} f_y^\tau}{\sum_{\tau=1, \dots, T} F_o^\tau} \right\} \right] = 1$$

Si $f_y^\tau = 0$ para cada τ , entonces

$$\left[\left\{ \frac{\sum_{\tau=1, \dots, T} f_y^\tau}{\sum_{\tau=1, \dots, T} F_o^\tau} \right\} \right] = 0$$

con lo cual se demuestra la proposición. ■

Por consiguiente, nuestro problema consiste en encontrar $f_o \in \mathcal{F}_o$ donde

$$| f_o^+ = \text{Min}_{f \in \mathcal{F}_o} \left\{ \sum_{f_i \in f} \left[\left\{ \frac{\sum_{\tau=1, \dots, T} f_y^\tau}{\sum_{\tau=1, \dots, T} F_o^\tau} \right\} \right] \right\}. \quad (4.10)$$

En otras palabras, el problema consiste en minimizar el funcional entero (4.10) y que por comodidad llamaremos **problema multiproducto PM**.

4.3 Solución al Problema Multiproducto

Primero que todo se presenta una definición necesaria. Una *matriz T-flujos M*, correspondiente al problema multiproducto **PM** y a una solución factible f , es la matriz cuya celda (i, j) es el vector fila f_{ij}^t de dimensión T (ver Figura 4). Diremos que una celda (i, j) está *libre* si $f_{ij}^t = 0$ para cada t , y diremos que está *ocupada* si $f_{ij}^t \neq 0$ para algún t . Nuevamente, a la cantidad de celdas ocupadas la llamaremos *cardinalidad* de M y la denotaremos por M^+ . Por consiguiente, si M es la matriz

de T -flujos correspondiente a una solución factible f del problema multiproducto

PM, se tiene que $|M|^+ = |f|^+$.

	15	21	10	15	8	11	6	10	2	8
17	4	4	4	6	3	6	4	6	2	5
10	4	4	4	2	5	0	4	0	0	0
9	4	6	2	5	3	0	0	0	0	3
5	3	5	0	0	0	2	0	0	0	0

Figura 4. Distribución de flujo donde se encuentra la matriz T -flujos M , para $T=2$.

4.3.1 Determinación de cotas inferiores.

Recordemos que para el caso uniproducto, se dispone de una cota inferior para el número de componentes positivas de f , esto es, del número de celdas ocupadas de M , o bien, el número mínimo de arcos que se necesitan para distribuir el producto, dada por Guardado [1982]

$$K_0 = \text{Max} \left\{ \sum_{x_i \in X^r} \lceil a(x_i)/F_0 \rceil, \sum_{x_j \in X^c} \lceil b(x_j)/F_0 \rceil \right\}.$$

Por lo tanto, para el caso de una red T -producto podemos encontrar un T -vector $K_0 = (K_0^1, \dots, K_0^r, \dots, K_0^T)$ donde K_0^r representa la cota inferior para el número mínimo de arcos que se necesitan para distribuir el producto τ . Un resultado es la siguiente proposición, la cual nos ofrece una cota para el número de arcos en una

red T -producto.

PROPOSICION 4.3 Una cota inferior para el problema multiproducto PM es

$$K_{inf} = \text{Max}_{\tau=1,\dots,T} \{K_o^\tau\}$$

DEM: Como K_o^τ representa una cota inferior para el número mínimo de arcos que se necesitan para distribuir el producto τ , entonces para cada $f \in \mathcal{F}_0$, se tiene que

$$K_o^\tau \leq |f^\tau|^+ \leq |f|^+; \text{ para cada } \tau=1,\dots,T.$$

Por lo tanto,

$$K_{inf} = \text{Max}_{\tau=1,\dots,T} \{K_o^\tau\} \leq |f|^+$$



Esta cota se puede refinar, como lo demuestra la siguiente proposición.

PROPOSICION 4.4 Una cota inferior para el problema multiproducto PM mayor que K_{inf} es

$$K = \text{Max}_{\tau=1,\dots,T} \{ |(f_o)^\tau|^+ \},$$

donde $(f_o)^\tau$ es la componente τ del vector f_o .

DEM: La solución óptima $(f_o)^\tau$ al problema uniproducto tiene número de arcos dado por $|(f_o)^\tau|^+$, para cada τ . Por lo tanto, para cada $f \in \mathcal{F}_0$, se tiene que

$$K_o^\tau \leq |(f_o)^\tau|^+ \leq |f^\tau|^+ \leq |f|^+; \text{ para cada } \tau=1,\dots,T.$$

Esto es, para cada $f \in \mathcal{F}_0$, se tiene que

$$K_{af} \leq \text{Max}_{\tau=1, \dots, T} \{ |f_{\tau}|^+ \} \leq |f|^+$$



En el capítulo 5 se encuentra un ejemplo donde este valor de K no se alcanza.

4.3.2 Solución para $T=2$.

Sean $f^1_{\theta}, f^2_{\theta}$ soluciones óptimas, dadas por el algoritmo AGD del capítulo 3, para los productos 1 y 2 respectivamente. Como vimos anteriormente, $f^* = (f^1_{\theta}, f^2_{\theta}) \in \mathcal{F}_0$. Sean M_1, M_2 las matrices de flujos de dimensión $m \times (n-m)$, asociadas a f^1_{θ} y f^2_{θ} , respectivamente. Luego, la matriz 2-flujos M asociada a f^* es aquella en que cada celda (i,j) está formada por el vector (f^1_{ij}, f^2_{ij}) . Observemos que si $|M|^+ = K$, entonces la solución al **problema biproducto PB**, esto es $T = 2$, es f^* , de otra forma, procedamos a encontrarla.

Para resolver el problema biproducto **PB** vamos a transformar la matriz M de tal manera que M transformada tenga el número mínimo de celdas ocupadas, o bien, un número máximo de celdas libres. El procedimiento de transformación de M involucra la transformación de M_1 y M_2 .

Sean C_1 y C_2 los conjuntos de celdas libres de M_1 y M_2 , respectivamente, podemos observar que si $(C_1 \subseteq C_2) \vee (C_2 \subseteq C_1)$ entonces $|M|^+ = K$, y la solución al problema es f^* . El problema general se da cuando $(C_1 \not\subseteq C_2) \wedge (C_2 \not\subseteq C_1)$. Por

lo tanto, necesitamos un procedimiento de búsqueda de la solución al problema. Dada la matriz M_1 , la matriz transformada de M_1 será denotada por $tr(M_1)$, y su conjunto de celdas libres será denotado por $tr(C_1)$. El procedimiento que se propone seguir es el siguiente.

PROCEDIMIENTO DE ACOPLAMIENTO

1).- Transformar M_1 hasta que $tr(M_1)$ sea de tal forma que $tr(C_1) \subseteq C_2$. Si esto es posible, la matriz transformada de M_1 , $tr(M_1)$, tiene en cada celda (i,j) por componente el vector $(f_{(0)1y}^1, f_{(0)1y}^2)$, donde $f_{(0)1y}^1, f_{(0)1y}^2$ son las componentes (i,j) de las matrices $tr(M_1)$ y M_2 , respectivamente. De esta manera, la solución buscada es el f_θ asociado a $tr(M_1)$. Si no es posible, llamemos $(f_\theta)^1$ al asociado a $tr(M_1)$ y entonces pasemos al siguiente inciso.

2).- Transformar M_2 hasta que $tr(C_2) \subseteq C_1$. Si esto es posible, $tr(M_2)$ tiene en cada celda (i,j) por componente el vector $(f_{(0)2y}^1, f_{(0)2y}^2)$, donde $f_{(0)2y}^1, f_{(0)2y}^2$ son las componentes (i,j) de las matrices M_1 y $tr(M_2)$, respectivamente. De esta manera, la solución buscada es el f_θ asociado a $tr(M_2)$. Si no es posible, llamemos $(f_\theta)^2$ al asociado a $tr(M_2)$ y entonces pasemos al siguiente inciso.

3).- La solución al problema biproducto PB es $f_\theta = (f_\theta)^t$ donde t se selecciona de acuerdo a

$$(f_\theta)^t|^+ = \text{Min} \{ |(f_\theta)^1|^+, |(f_\theta)^2|^+ \}.$$



Sin pérdida de generalidad podemos suponer que queremos transformar M_1 , ya que para transformar M_2 se procedería en forma similar. En el proceso de transformación de M_1 posiblemente aumente el número de celdas ocupadas en M_1 (si $|M_1| < |M_2|$), esto no importa porque el objetivo es aumentar el conjunto intersección de celdas libres de M_1 y M_2 .

La matriz M_1 será llamada *matriz transformable* y M_2 será llamada *matriz fija*.

Cada una de las celdas $(i,j) \in M_1$ se clasifican, conforme a las de M_2 , en una única de las siguientes definiciones:

celda no-operable: la celda (i,j) tal que $f^1_y = f^2_y = 0$.

celda operable: la celda (i,j) tal que $f^2_y \neq 0$.

celda removible: la celda (i,j) tal que $f^1_y \neq 0 \wedge f^2_y = 0$.

Para una ilustración de estas definiciones considere el ejemplo de la Figura 4, en el cual la matriz transformable es M_1 , ilustrada en la Figura 5, y la matriz fija es M_2 , ilustrada en la Figura 6.

$S_i \backslash T_j$	15	10	8	6	2
17	4	4	3	4	2
10	4	4	2	0	0
9	4	2	3	0	0
5	3	0	0	2	0

Figura 5. Distribución de flujos uniprodueto donde se encuentra la matriz transformable M_1 .

$S_i \backslash T_i$	21	15	11	10	8
27	4	6	6	6	5
19	6	4	5	4	0
14	6	5	0	0	3
5	5	0	0	0	0

Figura 6. Distribución de flujos uniproducto donde se encuentra la matriz fija M_7

Para este caso particular las celdas de M_1 se clasifican, conforme las de M_2 , de la siguiente forma:

celdas no-operables: $\{(2,9), (3,8), (4,6), (4,7), (4,9)\}$.

celdas operables: $\{(1,5), (1,6), (1,7), (1,8), (1,9), (2,5), (2,6), (2,7), (2,8), (3,5), (3,6), (3,9), (4,5)\}$.

celdas removibles: $\{(3,7), (4,8)\}$.

De esta forma, el procedimiento a seguir será redistribuir, de ser posible, el flujo de las celdas removibles hacia las celdas operables de M_1 , mediante δ -ciclos de redistribución que garanticen que el valor del flujo de cualquier celda del ciclo:

1) no exceda a F^i_0 y 2) no sea inferior a cero.

En otras palabras, los δ -ciclos de redistribución deberán tener las siguientes características antes de realizar la distribución del flujo δ :

R1) En las celdas pares, al sumarle δ al flujo existente, éste no debe exceder la

magnitud F^i_o .

R2) En las celdas impares, al restarle δ al flujo existente, éste no debe convertirse en flujo negativo.

Nótese que los δ -ciclos con estas características son similares a los utilizados en el algoritmo AGD. En lo que sigue, quedará implícito que un δ -ciclo satisface R1 y R2. El δ -ciclo de distribución puede cumplir la siguiente característica:

R3) La cantidad de celdas pares no-operables es menor o igual que la cantidad de celdas removibles impares con valor de flujo igual a δ .

Si el δ -ciclo de distribución cumple R3 se puede garantizar que no disminuya el número de celdas no-operables.

En consecuencia, el procedimiento a seguir es equivalente a aumentar, de ser posible, el número de celdas no operables en M_j . Nótese que las celdas operables pueden estar libres, por lo que es posible que en un momento dado el número de celdas ocupadas en M_j aumente, pero como anteriormente comentamos, esto no tiene importancia. El criterio de parada vendrá dado por la no existencia de δ -ciclos de distribución en las celdas removibles.

Si el δ -ciclo de distribución cumple con R3, éste recibe el nombre de *δ -ciclo no-disminuidor de celdas no-operables (δ CNDCNO)*. De lo anterior se sigue que la realización de un δ CNDCNO al menos mantiene el número de celdas no-

operables en la matriz M_I . Este tipo de ciclos son los que nos interesa encontrar.

4.3.2.1 Búsqueda de δ -ciclos no-disminuidores de celdas no-operables

(δ CNDCNO).

En dependencia del δ seleccionado y del valor de F'_o , las celdas de M_I pueden marcarse unívocamente de la siguiente forma:

celdas "&" : $\{(i,j) \text{ no-operable}\}$.

celdas "+" : $\{(i,j) \text{ operable o removible} : f'_{ij} + \delta \leq F'_o \text{ y } f'_{ij} - \delta < 0\}$.

celdas "-" : $\{(i,j) \text{ operable o removible} : f'_{ij} + \delta > F'_o \text{ y } f'_{ij} - \delta \geq 0\}$.

celdas "±" : $\{(i,j) \text{ operable o removible} : f'_{ij} + \delta \leq F'_o \text{ y } f'_{ij} - \delta \geq 0\}$.

celdas "@" : $\{(i,j) \text{ operable o removible} : f'_{ij} + \delta > F'_o \text{ y } f'_{ij} - \delta < 0\}$.

Distinguiremos con la marca adicional "*" a las celdas removibles "-" o "±" tales que $f'_{ij} - \delta = 0$. De lo anterior se tiene que, para un valor δ dado, las celdas "&" y las "+" solamente podrán ser celdas pares en los δ -ciclos; las celdas "-", celdas impares; las celdas "±" pueden ser ambas cosas y las celdas "@" no pueden pertenecer a ningún δ -ciclo.

Llamaremos δ -tabla a la tabla que se obtiene a partir de M_I al colocar en cada celda $(i,j) \in M_I$, la marca correspondiente. Teniendo en cuenta las características de los δ -ciclos resulta conveniente adoptar, en cada δ -tabla, las siguientes reglas de simplificación:

1).- Si alguna fila (columna) de la tabla contiene exclusivamente celdas dentro del

conjunto {"&", "+", "@"} ó dentro del conjunto {"-", "@"}, entonces puede ser extraída de la tabla, ya que en cualquiera de los casos ninguna celda puede formar parte de un δ -ciclo.

2).- Si alguna fila (columna) de la tabla contiene exclusivamente celdas "-" (sin considerar las celdas "@") y una única celda "±" en cuya correspondiente columna (fila) solamente existen celdas "+" o "&", dicha fila y columna podrán ser extraídas de la tabla.

3).- Cuando una fila (columna) sea extraída de la tabla no se alterará el número de orden de las restantes y será reincorporada a la tabla al variar el valor de δ .

Estas reglas se aplicarán sucesivamente mientras sea posible y como resultado obtendremos una δ -tabla transformada que denotaremos por $D(\delta)$ con la cual trabajaremos en adelante. Observe que si en $D(\delta)$ no existe alguna celda marcada con "*" podemos asegurar que para ese valor δ no existe δ CNDCNO.

Supongamos entonces que en $D(\delta)$ existe al menos una celda con la marca adicional "*". Pongamos en correspondencia con $D(\delta)$ un grafo orientado y marcado $\mathcal{G} = [\mathcal{X}^+, \mathcal{X}^-, \mathcal{U}]$ de la siguiente forma:

- El conjunto \mathcal{X}^+ lo formaremos con los nodos fuentes asociados a las filas de $D(\delta)$ y el conjunto \mathcal{X}^- con los nodos sumideros asociados a las columnas.
- A cada celda $(i,j) \in D(\delta)$ con marca "+" o "&" le corresponderá el arco (x_i, x_j) ; $x_i \in \mathcal{X}^+, x_j \in \mathcal{X}^-$.
- A cada celda $(i,j) \in D(\delta)$ con marca "-" le corresponderá el arco (x_i, x_j) ; $x_i \in \mathcal{X}^-, x_j \in \mathcal{X}^+$.

$x_i \in \mathcal{X}^+$.

- A cada celda $(i,j) \in D(\delta)$ con marca " \pm " le corresponderán los arcos (x_p, x_j) y (x_p, x_i) ; $x_i \in \mathcal{X}^+$, $x_j \in \mathcal{X}^-$.
- A las celdas "@" de $D(\delta)$ no se les asociará algún arco de \mathcal{h} .
- Si la celda $(i,j) \in D(\delta)$ posee la marca "&" o "*" dicha marca será trasladada al arco correspondiente.

No es difícil verificar que hay una correspondencia biunívoca entre \mathcal{h} y $D(\delta)$. Esto es, cada \mathcal{h} determina una única tabla $D(\delta)$ y viceversa, cada tabla $D(\delta)$ determina un único grafo orientado \mathcal{h} .

Dada la forma en que fue definida la dirección del arco (x_p, x_j) en \mathcal{h} en correspondencia con la marca de la celda (i,j) , resulta evidente que a cualquier δ -ciclo de distribución de la matriz M_i le corresponde en el grafo \mathcal{h} un circuito con una cantidad par de arcos no menor que cuatro.

De acuerdo a como fueron asignadas las marcas "&" y "*" y tomando en cuenta su significado, a un δ CNDCNO le corresponderá en \mathcal{h} un circuito en el cual, la cantidad de arcos con marca "*" es mayor o igual que la cantidad de arcos con marca "&". Este será llamado *circuito no-disminuidor*. Así, el problema de búsqueda de los δ -ciclos puede realizarse a través de la búsqueda de tales circuitos en \mathcal{h} , la cual se realizará sobre la base del procedimiento Búsqueda en Profundidad en grafos dirigidos [Szwarcfiter, 1984].

Por lo tanto, la búsqueda de los δ -ciclos que conduzcan, si es posible, al incremento del número de celdas no-operables puede resumirse en el siguiente procedimiento llamado *búsqueda de δ -ciclos no disminuidores de celdas no-operables* y que denotamos por B- δ CNDCNO.

PROCEDIMIENTO: Búsqueda de δ -ciclos no disminuidores de celdas no-operables (B- δ CNDCNO):

Dados un determinado valor de δ , una matriz M y la celda $(k,l) \in M$ que se analiza:

PASO 1. Hallar $D(\delta)$. Si la celda (k,l) fue eliminada quiere decir que para ese valor de δ y esa celda no existe un δ -ciclo de distribución e ir al PASO 3. En caso contrario, la celda (k,l) tiene marca "*", construir \mathcal{h} e ir al PASO 2.

PASO 2. Buscar una trayectoria no analizada en \mathcal{h} , mediante una búsqueda en profundidad con raíz x_k de x_k a x_l que no contenga el arco (x_p, x_k) . Si tal trayectoria no existe ir al PASO 3. Si la trayectoria existe, verificar que el circuito formado por dicha trayectoria y el arco (x_p, x_k) constituya un circuito no-disminuidor. Si lo es, entonces el procedimiento concluye redistribuyendo δ unidades de flujo a través del δ -ciclo hallado. Si no es un circuito no-disminuidor, iniciar nuevamente el PASO 2.

PASO 3. Construir $D(1)$. Si la celda $(k,l) \notin D(1)$ no existe 1-ciclo, lo que significa que el flujo de esa celda no se puede remover, y el procedimiento concluye. En caso contrario, construir el grafo \mathcal{h}' asociado a la tabla $D(1)$ e ir a PASO 4.

PASO 4. Buscar una trayectoria no analizada, mediante una búsqueda en profundidad con raíz x_k , de x_k a x_l que no contenga el arco (x_p, x_k) . Si la trayectoria no existe, no existe un 1-ciclo y el procedimiento concluye. Si existe ir al PASO 5.

PASO 5. En caso de existir trayectoria, verificar que el circuito formado por dicha trayectoria y el arco (x_p, x_k) constituya un circuito no-disminuidor. Si no lo es, ir al PASO 4. Si lo es, se realiza la redistribución de θ unidades de flujo a través del mismo, donde

$$\theta = \text{Min} \left\{ F^l_0 - \text{Max}_{(i,j) \text{ par}} \{ f^l_{ij} \}, \text{Min}_{(i,j) \text{ impar}} \{ f^l_{ij} \} \right\},$$

hacer $\delta = \delta - \theta$ y retornar al PASO 1.



La complejidad de este procedimiento es $O(mn)$ determinada por la complejidad de búsqueda en profundidad [Szwarcfiter, 1984]. La finitud del mismo se garantiza por el hecho de que en una fila la cantidad de celdas marcadas con "*" es finita al igual que el número de circuitos simples en un grafo. Este procedimiento se utilizará reiteradamente y constituye la base del algoritmo general de diseño biproducto que expondremos posteriormente.

Puede observarse en los Apéndices A y C que las matrices de flujos M_1 , M_2 asociadas a f_0^1 y f_0^2 respectivamente, acumulan las celdas libres, generalmente, en las últimas filas. Por consiguiente, las transformaciones de M_1 serán en orden

descendiente, con respecto al índice de filas, empezando por la última.

4.3.2.2 Algoritmo General de Diseño Biproducto (AGDB).

ALGORITMO

Dados M_1 la matriz de flujos asociada a f_0^1 , el valor F_0^1 y las celdas libres de M_2 .

Paso i ($i = m, \dots, 1$):

NIVEL 1. Sea $P_i = \{f_{ij}^1 : (i,j) \text{ es removible}, j=m+1, \dots, n\}$. Si $P_i = \phi$, reiniciar el paso para el siguiente valor de i . En caso contrario hacer

$$\delta^* = f_{ij}^1 = \text{Min}\{f_{ij}^1 \in P_i\}$$

y pasar al Nivel 2.

NIVEL 2. Ejecutar procedimiento B- δ CNDCNO con entradas δ^* , M_1 , (i,J) .

NIVEL 3. Eliminar la celda (i,J) de P_i e ir al NIVEL 1.



PROPOSICION 4.5 El algoritmo AGDB es finito y de complejidad $O(m^2n^2)$.

DEM: En cada fila $i=1, \dots, m$, el algoritmo AGDB ejecuta el procedimiento B- δ CNDCNO, el cual es finito, a lo sumo $(n-m)$ veces y este tiene una complejidad de $O(mn)$. Esto es, a lo más se realizan $m(n-m)$ ejecuciones del procedimiento B- δ CNDCNO. Por lo tanto, el algoritmo AGDB es finito y su complejidad es del orden $O(m^2n^2)$.



Nótese que en el algoritmo AGDB iniciamos, prácticamente, de dos soluciones

óptimas uniproducto, para los productos 1 y 2, las cuales en conjunto constituyen una solución factible para el problema biproducto. Como disponemos de una cota inferior del número mínimo de celdas ocupadas para este problema, solo existen dos posibilidades para esta solución factible del problema biproducto: *i)* El número de celdas ocupadas coincide con la cota inferior, *ii)* El número de celdas ocupadas no coincide con la cota inferior. En el primer caso esa solución factible sería la solución óptima del problema biproducto, en el segundo caso se intenta mejorar esa solución.

Podemos observar que el algoritmo encuentra una mejor solución que la solución inicial, pero solo podemos asegurar que ésta sea la óptima global cuando el número de celdas ocupadas sea igual a la cota inferior, como sucedió en la mayoría de los ejemplos de prueba realizados. Cuando el número de celdas ocupadas de la solución dada por el algoritmo AGDB no coincida con la cota inferior, no podemos asegurar que sea la solución óptima global, pero el conocimiento de las cotas inferiores nos permite evaluar "qué tan buena" es la solución obtenida. Por lo anteriormente mencionado, la solución encontrada por el algoritmo AGDB es una *solución entera aproximada* para el problema biproducto **PB**.

En resumen, la obtención de una solución entera aproximada $f_0 \in \mathcal{F}_0$ del problema biproducto **PB** se realizará según la siguiente secuencia:

ETAPA 1. Encontrar las soluciones f^1_0, f^2_0 soluciones óptimas uniproducto,

dadas por el algoritmo AGD del capítulo anterior, para los productos 1 y 2 respectivamente, y sean M_1, M_2 las matrices de flujos asociadas a ellas.

ETAPA 2. Aplicar el procedimiento de acoplamiento a M_1 y M_2 .

4.3.3 Solución Entera Aproximada para el Caso General Multiproducto

La obtención de un vector $f_0 \in \mathcal{F}_0$ que sea una solución entera aproximada al problema multiproducto PM, con T productos, se obtiene fácilmente basándonos en los resultados obtenidos para el caso biproducto. Solo se necesita aplicar el siguiente algoritmo.

ALGORITMO GENERAL DE DISEÑO MULTIPRODUCTO (AGDM)

Sean $f^1_{\theta}, f^2_{\theta}, \dots, f^T_{\theta}$ las soluciones óptimas uniproducto, dadas por el algoritmo AGD del capítulo 3, para los productos 1, 2, ... y T respectivamente, y sean M_1, M_2, \dots, M_T las matrices de flujos asociadas a ellas.

PASO 1. Aplicar el procedimiento de acoplamiento a M_1, M_2 . El acoplamiento se obtuvo transformando M_1 ó M_2 . Sean N^1, O^1 las matrices transformada y no transformada, respectivamente.

PASO 2. Aplicar el procedimiento de acoplamiento a N^1, M_3 . Sean N^2, O^2 las matrices transformada y no transformada, respectivamente.

•
•
•

PASO T-1. Aplicar el procedimiento de acoplamiento a N^{T-1}, M_T . Sean N^T, O^T las matrices transformada y no transformada, respectivamente.

Luego la matriz T -flujos M tal que cada celda (i,j) tiene por componente el vector $(O^1_{ij}, O^2_{ij}, \dots, O^{T-1}_{ij}, N^T_{ij})$ es una solución entera aproximada del problema multiproducto PM. No es muy difícil demostrar que el algoritmo AGDM es finito, de complejidad $O(Tm^2n^2)$.

4.4 Resumen

Debido a que los algoritmos existentes en la literatura revisada no eran utilizables en el problema formulado en este capítulo fue necesario diseñar para el mismo, algoritmos propios de solución. La estrategia que se siguió fue proceder de forma iterativa a partir de una solución inicial al problema, la cual fue obtenida por la aplicación del algoritmo AGD a cada uno de los productos. En seguida, se fue disminuyendo el número de arcos utilizados, siempre que fuera posible, obteniendo de esta forma soluciones cada vez mejores. Para lograr esto se diseñó el *procedimiento de acoplamiento* el cual transforma la solución inicial por medio del procedimiento B- δ CNDCNO, el cual realiza la búsqueda de ciclos que anulen el flujo de algún arco sin que se violen las restricciones del problema. El algoritmo AGDB, también diseñado aquí, se encarga de implementar el procedimiento de acoplamiento. Finalmente, se diseñó el algoritmo AGDM el cual se encarga de encontrar una solución entera aproximada al caso general de flujos multiproducto. Todos los algoritmos y procedimientos creados son finitos y de complejidad polinomial.

CAPITULO 5

APLICACION A UN PROBLEMA DE DISTRIBUCION DE PRODUCTOS

PROBLEMA: La empresa X es una empresa manufacturera de parabrisas para automóviles y la empresa Y es una empresa armadora de automóviles. La empresa X tiene un contrato con la compañía Y para distribuir dos tipos de parabrisas, el PARA1 y el PARA2. La empresa X tiene cuatro fábricas desde donde pueden enviarse los parabrisas requeridos por la empresa Y. La empresa Y tiene cinco armadoras donde se necesitan los parabrisas. Las Figuras 7 y 8 muestran las ofertas y demandas de los parabrisas PARA1 y PARA2, respectivamente.

		Y ₁	Y ₂	Y ₃	Y ₄	Y ₅
		15	10	8	6	2
X ₁	17					
X ₂	10					
X ₃	9					
X ₄	5					

Figura 7. Ofertas y demandas del producto PARA1.

		Y ₁	Y ₂	Y ₃	Y ₄	Y ₅
		21	15	11	10	8
X ₁	27					
X ₂	19					
X ₃	14					
X ₄	5					

Figura 8. Ofertas y demandas del producto PARA2.

Supongamos que el costo de envío es pequeño comparado con el costo de que no lleguen a su destino, el costo de perder al cliente, y que es el mismo desde cualquier fábrica de la empresa X a las armadoras de la empresa Y. La empresa X tiene interés de:

- i).*- Encontrar la distribución adecuada de parabrisas de tal forma que si por alguna causa externa a su distribución los envíos no llegaran, esto afectara lo menos posible a la empresa Y.
- ii).*- Encontrar la distribución que, bajo la condición *i)*, sea la que tenga menos rutas, lo cual hace que el costo de distribución sea mínimo.

SOLUCION:

I. Cálculo del valor F_{σ}

Para utilizar nuestros algoritmos vamos a hacer la siguiente asignación $x_{i+4} = y_{5-i}$ para $i=1, \dots, 5$.

a).- Para el producto PARA1.

PASO 1. Aplicar el algoritmo de distribución inicial para encontrar la matriz M .

La primera fila de M está dada por los valores:

$$f(x_1, x_5) = \text{Min} \{ 2, \lfloor 17/5 \rfloor \} = \text{Min} \{ 2, 3 \} = 2,$$

$$f(x_1, x_4) = \text{Min} \{ 6, \lfloor 15/4 \rfloor \} = \text{Min} \{ 6, 3 \} = 3,$$

$$f(x_1, x_3) = \text{Min} \{ 8, \lfloor 12/3 \rfloor \} = \text{Min} \{ 8, 4 \} = 4,$$

$$f(x_1, x_2) = \text{Min} \{ 10, \lfloor 8/2 \rfloor \} = \text{Min} \{ 10, 4 \} = 4,$$

$$f(x_1, x_4) = \text{Min} \{ 15, \lfloor 4/1 \rfloor \} = \text{Min} \{ 15, 4 \} = 4.$$

La segunda fila de M está dada por los valores:

$$f(x_2, x_5) = \text{Min} \{ 0, \lfloor 10/5 \rfloor \} = \text{Min} \{ 0, 2 \} = 0,$$

$$f(x_2, x_6) = \text{Min} \{ 3, \lfloor 10/4 \rfloor \} = \text{Min} \{ 3, 2 \} = 2,$$

$$f(x_2, x_7) = \text{Min} \{ 4, \lfloor 8/3 \rfloor \} = \text{Min} \{ 4, 2 \} = 2,$$

$$f(x_2, x_8) = \text{Min} \{ 6, \lfloor 6/2 \rfloor \} = \text{Min} \{ 6, 3 \} = 3,$$

$$f(x_2, x_9) = \text{Min} \{ 11, \lfloor 3/1 \rfloor \} = \text{Min} \{ 11, 3 \} = 3.$$

Así sucesivamente, llegamos a que la distribución inicial tiene las entradas que se muestran en la Figura 9.

	2	6	8	10	15
17	2	3	4	4	4
10	0	2	2	3	3
9	0	1	2	3	3
5	0	0	0	0	5

Figura 9. Distribución inicial para el cálculo de F_0 del producto PARA1.

La matriz M es la siguiente:

$$\begin{array}{cccc|c} 2 & 3 & 4 & 4 & 4 & | \\ 0 & 2 & 2 & 3 & 3 & \\ 0 & 1 & 2 & 3 & 3 & \\ 0 & 0 & 0 & 0 & 5 & \end{array}$$

PASO 2. Como el mayor valor de la matriz de flujos $M \neq f(x_1, x_2)$ entonces continuamos con el PASO 3.

PASO 3. Descomponemos M en 4 submatrices, donde M_{11} es la matriz

$$\begin{array}{|cccc|} \hline 2 & 3 & 4 & 4 \\ \hline 0 & 2 & 2 & 3 \\ \hline 0 & 1 & 2 & 3 \\ \hline 0 & 0 & 0 & 0 \\ \hline \end{array}$$

La matriz M_{12} es la matriz

$$\begin{array}{|c|} \hline 4 \\ \hline 3 \\ \hline 3 \\ \hline 5 \\ \hline \end{array}$$

y las matrices M_{21} y M_{22} son vacías. Por lo tanto,

$$F_0 = \lfloor (4+3+3+5)/4 \rfloor = \lfloor 15/4 \rfloor = 4.$$

b).- Para el producto PARA2 se puede hacer un análisis similar y encontrar que

$$F_0 = 6.$$

II. Cálculo de la cota inferior K_0

a).- Para el producto PARA1.

$$K_0 = \max\left\{\left(\lfloor 17/4 \rfloor + \lfloor 10/4 \rfloor + \lfloor 9/4 \rfloor + \lfloor 5/4 \rfloor\right), \left(\lfloor 2/4 \rfloor + \lfloor 6/4 \rfloor + \lfloor 8/4 \rfloor + \lfloor 10/4 \rfloor + \lfloor 15/4 \rfloor\right)\right\} = \max\{(5+3+3+2), (1+2+2+3+4)\}$$

$$= \max\{13, 12\} = 13.$$

b).- Para el producto PARA2 se puede hacer un análisis similar y encontrar que

$$K_0 = 13.$$

III. Cálculo de la solución inicial (Algoritmo 1).

Aplicando el Algoritmo 1 se obtienen las soluciones iniciales, para los productos PARA1 y PARA2, que se muestran en las Figuras 10 y 11. Observemos que se han reordenado los sumideros.

$S_i \backslash T_j$	15	10	8	6	2
17	4	4	4	4	1
10	4	4	2	0	0
9	4	2	2	1	0
5	3	0	0	1	1

Figura 10. Solución inicial para el producto PARA1.

$S_i \backslash T_j$	21	15	11	10	8
27	6	6	6	6	3
19	6	6	5	2	0
14	6	3	0	2	3
5	3	0	0	0	2

Figura 11. Solución inicial para el producto PARA2.

IV. Cálculo de la solución óptima uniproducto (Algoritmo AGD).

a).- Para el producto PARA1.

Mostraremos cómo se obtiene la solución óptima mediante una secuencia de figuras, resaltando los δ -ciclos de redistribución.

PASO $i=4$

NIVEL 1. $P_i = \{f_{45}=3, f_{48}=1, f_{49}=1\}$, $|P_i|=3$ y $\lfloor 5/4 \rfloor = 2$ entonces

$\delta = (\min\{3, 1, 1\} \neq 4) = 1$.

NIVEL 2. Como $1 \in \{3, 1, 1\}$. Ejecutar procedimiento B- δ C con valor $\delta=1$. En la Figura 12 se muestra que se encuentra un 1-ciclo disminuidor de celdas ocupadas.

$S_i \backslash T_j$	15	10	8	6	2
17	4	4	4	4	1
10	4	4	2	0	0
9	4	2	2	1	0
5	3	0	0	1	1

Figura 12. 1-ciclo disminuidor de celdas ocupadas en la celda (4,9).

Haciendo un análisis similar podemos llegar a la solución óptima. Las Figuras 13 y 14 muestran cómo se llega a la solución óptima para el producto PARA1.

$S_i \backslash T_j$	15	10	8	6	2
17	4	4	4	3	2
10	4	4	2	0	0
9	4	2	2	1	0
5	3	0	0	2	0

Figura 13. 1-ciclo disminuidor de celdas ocupadas en la celda (3,8).

$S_i \backslash T_j$	15	10	8	6	2
17	4	4	3	4	2
10	4	4	2	0	0
9	4	2	3	0	0
5	3	0	0	2	0

Figura 14. Solución óptima, número de celdas ocupadas es la cota inferior 13.

b).- Para el producto PARA2.

Haciendo un análisis similar encontramos la solución óptima para el producto PARA2 (Figuras 15, 16 y 17).

$S_i \backslash T_j$	21	15	11	10	8
27	6	6	6	6	3
19	6	6	5	2	0
14	6	3	0	2	3
5	3	0	0	0	2

Figura 15. 2-ciclo disminuidor de celdas ocupadas en la celda (4,9).

$S_i \backslash T_j$	21	15	11	10	8
27	4	6	6	6	5
19	6	6	5	2	0
14	6	3	0	2	3
5	5	0	0	0	0

Figura 16. 2-ciclo disminuidor de celdas ocupadas en la celda (3,8).

$S_i \backslash T_j$	21	15	11	10	8
27	4	6	6	6	5
19	6	4	5	4	0
14	6	5	0	0	3
5	5	0	0	0	0

Figura 17. Solución óptima, número de celdas ocupadas es la cota inferior

V. Cálculo de una solución entera aproximada biproducto (AGDB).

La solución inicial es la matriz dos-flujos M , mostrada en la Figura 18, donde las matrices que la componen son las matrices M_1 y M_2 que se muestran en las distribuciones de las figuras 14 y 17, respectivamente.

	15	21	10	15	8	11	6	10	2	8
17	27	4	4	4	6	3	6	4	6	2
10	19	4	6	4	4	2	5	0	4	0
9	14	4	6	2	5	3	0	0	0	3
5	5	3	5	0	0	0	2	0	0	0

Figura 18. Distribución inicial donde se encuentra la solución inicial M .

Observemos que la cota inferior, sobre el número mínimo de celdas que pueden ser ocupadas en la matriz dos-flujos M es $K = \max\{13, 13\} = 13$. Podemos observar también que $(C_1 \not\subseteq C_2) \wedge (C_2 \not\subseteq C_1)$, por lo tanto, aplicamos el procedimiento de acoplamiento.

PROCEDIMIENTO DE ACOPLAMIENTO.

1).- Primeramente se transforma la matriz de distribución que se muestra en la Figura 14, de tal forma que se maximice el número de celdas libres que se empalmen con las celdas libres de la matriz de distribución que se muestra en la

Figura 17. Mostraremos cómo se obtiene el empalme de celdas mediante una secuencia de figuras, resaltando los δ -ciclos de redistribución. Las celdas marcadas "x", son las celdas no-operables y las marcadas por "*" son las removibles.

PASO $i=4$

NIVEL 1. $P_4 = \{f'_{48}=2\}$ entonces $\delta^* = 2$.

NIVEL 2. Ejecutar procedimiento B- δ CNDCNO con valor $\delta^* = 2$, M_1 , (4,8). ¡No se encontró 1-ciclo!

NIVEL 3. Eliminar (4,8) e ir al NIVEL 1.

NIVEL 1. $P_4 = \phi$. Reiniciar el paso para el siguiente valor de i .

PASO $i=3$

NIVEL 1. $P_3 = \{f'_{37}=3\}$ entonces $\delta^* = 3$.

NIVEL 2. Ejecutar procedimiento B- δ CNDCNO con valor $\delta^* = 3$, M_1 , (3,7). ¡No se encontró un 3-ciclo no disminuidor de celdas no operables! ¡Se encontró un 2-ciclo no disminuidor de celdas no operables!, Figura 19. ¡Se encontró un 1-ciclo no disminuidor de celdas no operables!, Figura 20.

NIVEL 3 Eliminar (3,7) e ir al NIVEL 1.

NIVEL 1. $P_3 = \phi$. Reiniciar el paso para el siguiente valor de i .

PASO $i=2$ y PASO $i=1$, se tiene que P_2 y P_1 son vacíos.

SOLUCION FINAL. Empalme máximo encontrado, Figura 21.



<i>Si</i> \ <i>Ti</i>	15	10	8	6	2
17	4	4	3	4	2
10	4	4	2	0	0
9	4	2	3	0	0
5	3	0	0	2	0

Figura 19. 2-ciclo no disminuidor de celdas no operables en la celda (3,7).

<i>Si</i> \ <i>Ti</i>	15	10	8	6	2
17	4	4	3	4	2
10	4	2	4	0	0
9	4	4	1	0	0
5	3	0	0	2	0

Figura 20. 1-ciclo no disminuidor de celdas no operables en la celda (3,7).

<i>Si</i> \ <i>Ti</i>	15	10	8	6	2
17	4	4	4	4	1
10	4	2	4	0	0
9	4	4	0	0	1
5	3	0	0	2	0

*Figura 21. No existe 1-ciclo no disminuidor de celdas no operables.
Empalme máximo.*

b).- Como no conseguimos empalmar todas las celdas libres (comparar con la distribución de la Figura 17), lo que haremos ahora es transformar la matriz de distribución que se muestra en la Figura 17, de tal forma que se maximice el número de celdas libres que se empalmen con las celdas libres de la matriz de distribución que se muestra en la Figura 14.

Haciendo un análisis similar al inciso a) sólo mostraremos cómo se obtiene el empalme de celdas mediante una secuencia de Figuras, resaltando los δ -ciclos de redistribución.

$S_i \backslash T_j$	21	15	11	10	8
27	4	6	6	6	5
19	6	4	5	4	0
14	6	5	0	0	3
5	5	0	0	0	0

Figura 22. 1-ciclo no disminuidor de celdas no operables en la celda (3,9).

$S_i \backslash T_j$	21	15	11	10	8
27	4	5	6	6	6
19	6	4	5	4	0
14	6	6	0	0	2
5	5	0	0	0	0

Figura 23. No existe 1-ciclo no disminuidor de celdas no operables. Seleccionar nueva celda removible.

$S_i \backslash T_j$	21	15	11	10	8
27	4	5	6	6	6
19	6	4	5	4	0
14	6	6	0	0	2
5	5	0	0	0	0

Figura 24. 2-ciclo no disminuidor de celdas no operables en la celda (2,8).

$S_i \backslash T_j$	21	15	11	10	8
27	6	3	6	6	6
19	6	6	5	2	0
14	6	6	0	0	2
5	3	0	0	2	0

Figura 25. No existe 1-ciclo no disminuidor de celdas no operables. Empalme máximo.

Por lo tanto, una solución entera aproximada al problema planteado, es la siguiente matriz de distribución 2-flujos. Observemos que el número de celdas ocupadas es $14 > K (-13)$, mientras que en la solución inicial eran 15 (Figura 18).

	Y_1	Y_2	Y_3	Y_4	Y_5					
X_1	15 17	21 27	10 4	15 4	8 4	11 6	6 4	10 6	2 1	8 5
X_2	10 19	4 6	2 6	4 4	4 5	0 0	0 4	0 0	0 1	0 3
X_3	9 5	4 5	6 3	4 5	0 0	0 0	0 2	0 0	0 0	0 0

Figura 26. Solución entera aproximada al problema planteado.

CAPITULO 6

CONCLUSIONES Y RECOMENDACIONES

En este trabajo se construye el modelo matemático del problema de diseñar una red uniproducto con un número mínimo de arcos, sujeta a condiciones de uniformidad o balanceo en los flujos. Se reformula el algoritmo AGD, propuesto por Alvarez y Guardado [1992], que resuelve el problema mencionado para el caso en que la red es bipartita completa. Se demuestra matemáticamente la optimalidad del algoritmo AGD. Se demuestra la consistencia del algoritmo AGD, en el sentido de que dada cualquier solución del problema, podemos transformarla en la solución obtenida por el algoritmo AGD.

Generalizando lo anterior, se construye el modelo matemático del problema de diseñar una red multiproducto con un número mínimo de arcos, sujeta a condiciones de uniformidad o balanceo en los flujos. Se encontraron cotas inferiores para el número mínimo de arcos y se diseñó el algoritmo AGDM que encuentra una solución entera aproximada del problema mencionado, el cual es finito y de complejidad polinomial.

Con los resultados obtenidos en la presente tesis se llegó a las siguientes conclusiones.

Aún cuando el problema general de diseñar una red óptima es un problema *NP-hard*, la presente tesis exhibe un algoritmo polinomial que resuelve este problema óptimamente para el caso particular de redes bipartitas con flujos uniproducto, sobre las cuales se considera que los costos por unidad de flujo sobre los arcos son despreciables en comparación con el costo de construcción de ellos. En este caso la solución es exacta, a diferencia de los resultados encontrados en la literatura revisada [Goemans *et al*, 1994], donde se exhibe un algoritmo polinomial que es de aproximación.

La consideración de una red con flujos multiproducto complica aún más el problema de diseñar una red óptima. En la literatura abordan el problema de diseñar una red óptima con flujos multiproducto pero sin restricciones de capacidad en los arcos y se encuentran algoritmos de aproximación [Minoux, 1989]. La presente tesis exhibe un algoritmo polinomial que encuentra soluciones enteras aproximadas para el caso particular de redes bipartitas con flujos multiproducto, sobre las cuales se hace la misma consideración en relación a los costos, que la hecha para el caso uniproducto.

Aún cuando se ejemplifica la aplicación práctica de los algoritmos diseñados en este trabajo de tesis sobre un problema de distribución de productos en el cual

se aplican los algoritmos estudiados en este trabajo de tesis, éste no es el único tipo de problemas donde ellos se pueden aplicar. Los algoritmos también son aplicables, por ejemplo, a problemas de redes de comunicación por computadora. Una aplicación a este tipo de problemas pudiera ser el de encontrar la mínima capacidad de los enlaces de la red, que sujeta a condiciones de oferta-demanda sea uniforme para cada uno de los enlaces, donde además, se desee minimizar el número de enlaces.

Por lo anteriormente mencionado, se han alcanzado los objetivos propuestos en este trabajo de tesis.

Es de vital importancia mencionar que en la presente tesis no se resolvió el problema para el caso de una red general, éste se deja para investigaciones futuras. Asimismo, en este trabajo consideramos que el costo de cada uno de los enlaces es unitario. Alvarez [1993] desarrolla un algoritmo que resuelve parcialmente el problema para el caso con costos no unitarios, sobre redes bipartitas completas con flujos uniproducto, pero no se ha formulado ni resuelto el problema sobre una red arbitraria con flujos multiproducto. Esto también se deja para investigaciones futuras.

BIBLIOGRAFIA

Adam, E. E. and Ebert, R. J. *Administración de la Producción y las Operaciones. Conceptos, modelos y funcionamiento*, cuarta edición, Prentice-Hall Hispanoamericana, S. A., México, 1991.

Ahuja, R. K., Magnanti, T. L. and Orlin, J. B. *Network Flows. Theory, Algorithms and Applications*, Prentice Hall, Englewood Cliffs, N. J., 1993.

Ahuja, V. *Design and Analysis of Computer Communication Networks*, McGraw-Hill International Editions Computer Science Series, 1985.

Alvarez, A. M. "Síntesis de una red de tipo de transporte con un número mínimo de aristas dada una cota superior de flujo", *Investigación Operacional*, Vol. 2, No. 3, pp. 3-9, 1985.

Alvarez, A. M. y Guardado, M. "Síntesis de Redes Bipartidas Simples con un Número Mínimo de Aristas", *Investigación Operacional*, Vol. 13, No. 3, pp. 185-195, 1992.

Alvarez, A. M. "Algunos Problemas de Síntesis de Redes del Tipo de Transporte", Tesis Doctoral, Universidad Central de las Villas, Cuba, 1993.

Aoki, K., Nara, K., Satoh, T., Kitagawa, M. and Yamanaka, K. "New Approximate Optimization Method for Distribution System Planning", *IEEE Transactions on Power Systems*, Vol. 5, No. 1, February 1990.

Assad, A. A. "Multicommodity Network Flows -A Survey", *Networks*, Vol. 8, pp. 37-91, 1978.

Bazaraa, M. S., Jarvis, J. J. and Sherali, H. D. *Linear Programming and Network Flows*, second edition, John Wiley & Sons, 1990.

Bienstock, D. and Gonlok, O. "Computational Experience with a Difficult Mixed-Integer Multicommodity Flow Problem", *Mathematical Programming*, Vol. 68, pp. 213-237, 1995.

Chase, R. B., and Aquilano, N. J. *Production and Operations Management. A life cycle approach*, fifth edition, Irwin, inc., 1989.

Chiang, H. and Jean-Jumeau, R. "Optimal Network Reconfigurations in Distribution Systems: Part 1: A New Formulation and A Solution Methodology", *IEEE Transactions on Power Delivery*, Vol. 5, No. 4, November 1990.

Ford, L. R., and Fulckerson, *Flows in Networks*, Princeton University Press, Princeton, N. J., 1962.

Garey, M. R. and Johnson, D. S. *Computers and Intractability: A Guide to the Theory of NP-completeness*, W. H. Freeman and Company, New York, 1979.

Goemans, M. X., Tardos, E., Shmoys, D. B., Goldberg, A. V., Plotkin, S. and Williamson, D. P. "Improved Approximation Algorithms for Network Design Problems", in *Symposium on Discrete Algorithms ACM-SIAM*, 1994.

Gomory, R. E. and Hu, T. C. "An Application of Generalized Linear Programming to Network Flows", *J. SIAM*, Vol. 10, No. 2, pp. 260-283, June 1962.

Gomory, R. E. and Hu, T. C. "Synthesis of a Communication Network", *J. SIAM*, Vol. 12, No. 2, pp. 348-369, June 1964.

Guardado, M., "Método de síntesis de redes bipartidas con valor mínimo de máximo flujo por las aristas". *Viestnik Bieloruskago Univiersitieta. Serie 1. Registro 5834-82*. 1982.

Hu, T. C. *Integer Programming and Network Flows*, Addison-Wesley, 1969.

Itai, A. and Pradhan, D. K. "Synthesis of Directed Multicommodity Flow Networks", *Networks*, Vol. 14, pp. 213-224, 1984.

Janz, B., Brancheau, J. C. and Wetherbe, J. C. "Key Issues in IS Management", por aparecer, 1995.

Jensen, P. A. and Barnes, J. W. *Network Flow Programming*, John Wiley & Sons, 1980.

Johnson, D. S., Lenstra, J. K. and Rinnooy Kann, A. H. G. "The Complexity of the Network Design Problem", *Networks*, Vol. 8, pp. 279-285, 1978.

Kamath, A. and Palmon, O. "Improved Interior Point Algorithms for Exact and Approximate Solution of Multicommodity Flow Problems" in *Symposium on Discrete Algorithms* ACM-SIAM, 1995.

Kennington, J. L. "A Survey of Linear Cost Multicommodity Network Flows", *Operations Research*, No. 26, pp. 209-236, 1978.

Kennington, J. L. and Helgason, R. V. *Algorithms for Network Programming*, John Wiley & Sons, 1980.

Lucertini, M. and Paletta, G. "A Class of Network Design Problems with Multiple Demand: Model Formulation and An Algorithmic Approach", *Mathematical Programming Study*, Vol. 26, pp. 225-228, 1986.

Ma P. R., Lee, E. Y. S., and Tsuchiya, M. "A Task Allocation model for Distributed Computing Systems", *IEEE Transactions on Computers*, Vol c-31, No. 1, pp. 41-47, January 1982.

Martin, J. M. "Strategic Planning for Manufacturing", *Manufacturing Engineering*, November 1987, pp. 45-50.

Minieka, E. *Optimization Algorithms for Networks and Graphs*, Marcel Dekker, New York, 1978.

Minoux, M. "Network Synthesis and Dynamic Network Optimization", in *Surveys in Combinatorial Optimization*, Edited by S. Martello, G. LaPorte, M. Minoux and C. Ribeiro, Annals of Discrete Mathematics, North -Holland, Vol. 31, pp. 283-324, 1987.

Minoux, M. "Network Synthesis and Optimum Network Design Problems: Models, Solution Methods and Applications", *Networks*, Vol. 19, pp. 313-360, 1989.

Naor, M. and Roth, R. M. "Optimal File Sharing in Distributed Networks", *SIAM J. on Computing*, Vol. 24, No. 1, pp 158-183, February 1995.

Niederman, F., Brancheau, J. C. and Wetherbe, J. C. "Information Systems Management Issues for the 1990s", *MIS Quarterly*, Vol. 15, No. 4, pp. 475-495, December 1991.

Ragsdale, C. T. and McKeown, P. G. "A Computational Study of Using Preprocessing and Stronger Formulations to Solve Large Central General Fixed Charge Problems", *Computers Operations Research*, Vol. 17, No. 1, pp 9-16, 1990.

Ragsdale, C. T. and McKeown, P. G. "An Algorithm for Solving Fixed-Charge Problems Using Surrogate Constraints", *Computers Operations Research*, Vol. 18, No. 1, pp 87-96, 1991.

Rockart, J. F. and Short, J. E. "The Networked Organization and the Management of Interdependence", in *The Corporation of the 1990s. Information Technology and Organizational Transformation*, edited by Michael S. Scott Morton, Oxford University Press, New York, 1991.

Schwartz, M. *Computer Communication Network Design and Analysis*, Prentice Hall, Englewood Cliffs, N. J., 1977.

Shen, Chien-Chung and Tsai Wen-Hsiang. "A Graph Matching Approach to Optimal Task Assignment in Distributed Computing Systems Using a Minimax Criterion", *IEEE Transaction on Computers*, Vol c-34, No. 3, pp. 197-203, March 1985.

Sridhar, S. and Chandrasekaran, R. "Integer Solution to Synthesis of Communication Networks", *Mathematics of Operations Research*, Vol. 17, No. 3, pp. 581-585, August 1992.

Szwarcfiter, J. L., *Grafos e Algoritmos Computacionais*, Editorial Campus, Rio de Janeiro, 1984.

Zwass, V. and Veroy, B. "Capacity Expansion for Information Flow Distribution in Multi-Path Computer Communication Networks", *Journal of Management Information Systems*, Vol. 5, No. 2, pp. 57-70, Fall 1988.

APENDICES

APENDICE A

**UN EJEMPLO DONDE EL PROBLEMA DEL TRANSPORTE CON
CAPACIDADES NO RESUELVE EL PROBLEMA UNIPRODUCTO**

APENDICE A

UN EJEMPLO DONDE EL PROBLEMA DEL TRANSPORTE CON CAPACIDADES NO RESUELVE EL PROBLEMA UNIPRODUCTO

La Figura 27 nos presenta un problema de transporte a capacidades, donde el valor de la capacidad máxima es 7.

Si \ Ti	19	12	12
20	f_{11}	f_{12}	f_{13}
12	f_{21}	f_{22}	f_{23}
11	f_{31}	f_{32}	f_{33}

Figura 27. Matriz inicial de costos.

La Figura 28 nos muestra la solución a este problema utilizando la técnica del Método Simplex.

COSTO MINIMO 43			
Si \ Ti	19	12	12
20	7	6	7
12	5	2	5
11	7	4	0

Figura 28. Solución óptima del Problema de Transporte, obtenida por el método simplex.

La Figura 29 nos muestra la solución dada por el algoritmo AGD. Podemos observar que el número de celdas ocupadas por la solución óptima de transporte es mayor que el número de celdas encontradas por nuestro algoritmo.

Si \ Ti	19	12	12
20	7	7	6
12	7	5	0
11	5	0	6

Figura 29. Solución óptima obtenida por el algoritmo AGD.

APENDICE B

DETERMINACION DEL VALOR F_0

APENDICE B

DETERMINACION DEL VALOR F_0

Cota inferior para F_0

Formemos una matriz de flujos M , donde las filas corresponden a las fuentes y las columnas a los sumideros. Los valores de las fuentes (ofertas) se colocan en orden no creciente y los valores de los sumideros (demandas) en orden no-decreciente. Una cota inferior para F_0 es

$$F_{inferior} = \text{Max} \left\{ \left\lceil B/(m-k) \right\rceil, \left\lceil A/[(n-m)-r] \right\rceil \right\}$$

donde,

k : es el mayor entero en el intervalo $[1, \dots, m-1]$ tal que

$$a(x_{m-(k-1)}) < \left\lceil b(x_n)/m \right\rceil.$$

Si no existe k , entonces $k = 0$.

r : es el mayor entero en el intervalo $[1, \dots, (n-m)-1]$ tal que

$$b(x_{r+m}) < \left\lceil a(x_1)/(n-m) \right\rceil.$$

Si no existe r , entonces $r = 0$.

Finalmente,

$$B = b(x_n) - \sum_{i=(m-(k-1)), \dots, m} a(x_i) \quad A = a(x_j) - \sum_{j=1, \dots, r} b(x_{m+j})$$

Determinación de una distribución inicial.

Sin pérdida de generalidad, podemos suponer que la cota inferior

$$F_{inferior} = \lceil A/[(n-m)-r] \rceil,$$

ya que en caso de que sea

$$F_{inferior} = \lceil B/(m-k) \rceil,$$

es suficiente intercambiar las ofertas por las demandas y después de resolver el problema volver a la situación inicial y cambiar la orientación de los arcos encontrados.

El procedimiento será llenar la matriz M por las filas. Antes de iniciar el algoritmo se definirán algunos términos básicos.

k : contador de filas, esto es, $k=1, \dots, m$.

j : contador de columnas, esto es, $j=(m+1), \dots, n$.

Si $k=1$, $b^1(x_j) = b(x_j)$. En general, $b^{k+1}(x_j) = b^k(x_j) - f(x_k, x_j)$, para cada $k=1, \dots, (m-1)$.

En otras palabras, $b^{k+1}(x_j)$ es la demanda que queda en el sumidero x_j , después de haber llenado la fila k de la matriz M , para $k=1, \dots, (m-1)$.

Si $j=m+1$, $a^{m+1}(x_j) = a(x_j)$. En general, $a^{j+1}(x_j) = a(x_j) - f(x_k, x_j)$, para cada $j=(m+1) \dots, (n-1)$. En otras palabras, $a^{j+1}(x_j)$ es la oferta que queda en la fuente

x_k después de haber llenado la celda (j,k) de la matriz M , para $j=(m+1),\dots,(n-1)$.

Algoritmo de distribución inicial.

PASO 1. Para cada $k=1,\dots,m-1$ hacer lo siguiente:

Para cada $j=m+1,\dots,n$.

$$f(x_k, x_j) = \text{Min} \left\{ b^k(x_k), \left\lfloor a^j(x_j) / (n-j+1) \right\rfloor \right\}.$$

donde $\lfloor x \rfloor$ representa el mayor entero menor o igual que x .

PASO 2. Para $k=m$.

$$f(x_k, x_j) = b^k(x_k), \text{ para cada } j=m+1,\dots,n.$$

Cálculo del valor F_0

Algoritmo.

PASO 1. Aplicar el algoritmo de distribución inicial para encontrar la matriz M .

PASO 2. Si el mayor valor de la matriz de flujos M es $f(x_1, x_n)$ entonces

$$F_0 = f(x_1, x_n).$$

Si no es así, ir al PASO 3.

PASO 3. Sea X el conjunto de celdas de M que tengan el mayor valor del flujo en la matriz, y definamos $(r,s) \in X$ de tal manera que $r = \text{Min}\{i\}$, para cada $(i,j) \in X$ y $s = \text{Max}\{j\}$, para cada $(r,j) \in X$. Dividamos M en cuatro submatrices de la siguiente forma:

$$M = \begin{array}{c|c} M_{11} & M_{12} \\ \hline M_{21} & M_{22} \end{array}$$

donde M_{11} es la matriz cuyas componentes son $f(x_p, x_j)$ para $i=1, \dots, r; j=m+1, \dots, s-1$. M_{12} es la matriz cuyas componentes son $f(x_p, x_j)$ para $i=1, \dots, r; j=s, \dots, n$. M_{21} es la matriz cuyas componentes son $f(x_p, x_j)$ para $i=r+1, \dots, m; j=m+1, \dots, s-1$. M_{22} es la matriz cuyas componentes son $f(x_p, x_j)$ para $i=r+1, \dots, m; j=s, \dots, n$.

Si M_{21} es la matriz nula, o no existe, entonces

$$F_0 = \text{Max} \left\{ \left[\left(\sum_{i=1, \dots, r} \sum_{j=s, \dots, n} f(x_p, x_j) \right) / (r(n-s+1)) \right], F_{inferior} \right\}$$

Si lo anterior no se cumple ir al PASO 4.

PASO 4.

1.- Hagamos decrecer el valor de s hasta encontrar un valor $m+1 \leq s^* < s$ de tal forma que la matriz M_{21} sea nula o vacía. (De esta manera, M se divide en cuatro submatrices de la siguiente forma:

$$M = \begin{array}{|c|c|} \hline M_{11}^* & M_{12}^* \\ \hline M_{21}^* & M_{22}^* \\ \hline \end{array}$$

donde M_{11}^* es la matriz cuyas componentes son $f(x_p, x_j)$ para $i=1, \dots, r; j=m+1, \dots, s^*-1$. M_{12}^* es la matriz cuyas componentes son $f(x_p, x_j)$ para $i=1, \dots, r; j=s^*, \dots, n$. M_{21}^* es la matriz cuyas componentes son $f(x_p, x_j)=0$ para $i=r+1, \dots, m; j=m+1, \dots, s^*-1$. M_{22}^* es la matriz cuyas componentes son $f(x_p, x_j)$ para $i=r+1, \dots, m; j=s^*, \dots, n$.)

Luego, calculemos

$$F_0^* = \left[\left(\sum_{i=1, \dots, r} \sum_{j=s^*, \dots, n} f(x_p, x_j) \right) / (r(n-s^*+1)) \right]$$

2.- Aumentemos el valor de r hasta encontrar un valor $r < r^* \leq m$ de tal forma que la matriz M_{21} sea nula o vacía. (De esta manera, M se divide en cuatro

submatrices de la siguiente forma:

$$M = \begin{vmatrix} M^{**}_{11} & M^{**}_{12} \\ M^{**}_{21} & M^{**}_{22} \end{vmatrix}$$

donde M^{**}_{11} es la matriz cuyas componentes son $f(x_p, x_j)$ para $i=1, \dots, r^*$; $j=m+1, \dots,$

$s-1$. M^{**}_{12} es la matriz cuyas componentes son $f(x_p, x_j)$ para $i=1, \dots, r^*$; $j=s, \dots, n$.

M^{**}_{21} es la matriz cuyas componentes son $f(x_p, x_j)=0$ para $i=r^*+1, \dots, m$; $j=m+1, \dots,$

$s-1$. M^{**}_{22} es la matriz cuyas componentes son $f(x_p, x_j)$ para $i=r^*+1, \dots, m$; $j=s, \dots, n$.)

Luego, calculemos

$$F^{**}_0 = \left[\left(\sum_{i=1, \dots, r^*} \sum_{j=s, \dots, n} f(x_p, x_j) \right) / (r^*(n-s+1)) \right].$$

$$3.- F_0 = \text{Max} \{ F^*_0, F^{**}_0, F_{inferior} \}.$$

APENDICE C

**UN EJEMPLO DONDE EXISTEN DOS SOLUCIONES A UN
PROBLEMA UNIPRODUCTO**

APENDICE C

UN EJEMPLO DONDE EXISTEN DOS SOLUCIONES A UN PROBLEMA UNIPRODUCTO

Considere el problema uniprodueto P con los datos de entrada de la Figura 30.

Si \ Ti	15	10	8	6	2
17	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
10	f_{21}	f_{22}	f_{23}	f_{24}	f_{25}
9	f_{31}	f_{32}	f_{33}	f_{34}	f_{35}
5	f_{41}	f_{42}	f_{43}	f_{44}	f_{45}

Figura 30. Problema inicial.

La Figura 31 muestra la distribución inicial dada por el Algoritmo 1.

Si \ Ti	15	10	8	6	2
17	4	4	4	4	1
10	4	4	2	0	0
9	4	2	2	1	0
5	3	0	0	1	1

Figura 31. Distribución inicial.

Con la distribución inicial anterior podemos encontrar dos soluciones diferentes al problema uniproducción P, dependiendo del circuito encontrado por el procedimiento B- δ C. Esto se muestra en las Figuras 32 y 33.

Si \ Tj	15	10	8	6	2
17	4	4	3	4	2
10	4	4	2	0	0
9	4	2	3	0	0
5	3	0	0	2	0

Figura 32. Una solución al problema uniproducción P.

Si \ Tj	15	10	8	6	2
17	4	4	4	4	1
10	4	4	0	2	0
9	4	2	3	0	0
5	4	0	0	0	1

Figura 33. Una solución al problema uniproducción P diferente a la de la Figura 32.

RESUMEN AUTOBIOGRAFICO

José Luis Martínez Flores

Candidato para el Grado de Doctor en Ingeniería de Sistemas
con Especialidad en Investigación de Operaciones

**Tesis: ALGORITMOS QUE MINIMIZAN EL NUMERO DE ARCOS EN
REDES BIPARTITAS COMPLETAS CON FLUJOS BALANCEADOS**

Campo de Estudio: Ingeniería

Biografía:

Datos personales: Nacido en el poblado de El Coyote, Municipio de Matamoros, Coahuila el 23 de septiembre de 1962, hijo de Ernesto Martínez Molina y Raquel Flores Santoyo.

Educación:

Egresado de la Universidad Autónoma de Nuevo León, grados obtenidos Licenciado en Matemáticas en 1983 y Maestro en Ciencias de la Administración con Especialidad en Sistemas en 1994.

Experiencia Profesional: Profesor de Tiempo Completo en la Escuela de Ciencias Físico-Matemáticas de la Universidad Autónoma de Puebla de 1984 a 1991, Profesor de Medio Tiempo en el Departamento de Economía y Administración de la Universidad Madero de Puebla de 1990 a 1991, Profesor por Horas y de Postgrado en la Facultad de Ingeniería Mecánica y Eléctrica de la Universidad Autónoma de Nuevo León de 1993 a Julio de 1995.

