TECNIA, Vol.12 N° 1, págs. 79 - 86, 2002
Universidad Nacional de Ingeniería
Lima - Perú

ALGORITMO DE ETIQUETAS PARA EL PROBLEMA DEL FLUJO MÁXIMO

Rósulo Pérez Cupe
Escuela Profesional de Matemática
Facultad de Ciencias
E-mail : perezcu@yahoo.com

RESUMEN

En el presente trabajo se estudia el teorema del Flujo máximo–Corte mínimo(L.Ford y D. Fulkerson) desde el punto de vista práctico, esto es, su demostración se basa en la prueba de correctitud del algoritmo de etiquetas. Además se presenta la implementación de tal algoritmo.

ABSTRACT

Presently work is studied the theorem of the flow maximum-cut minimum (L. Ford and D. Fulkerson) from the practical poin of view, this is its demonstration is based on the test of correctude of the algorithm of labels . The implementation of such an algorithm is also presented.

INTRODUCCIÓN

En esta sección se estudia el método clásico de Ford-Fulkerson para resolver el problema del Flujo máximo; se presenta el algoritmo de **etiquetas,** el cual no es un algoritmo polinomial, a diferencia de los algoritmos **capacity scaling, shortest augmenting path** y **preflow push.**

ALGORITMO DE ETIQUETAS

A continuación daremos una serie de definiciones y conceptos importantes que no sólo son útiles para este método (de etiquetas), sino para muchos otros métodos , estos conceptos son **red residual,** y **cortes**.

Notaciones y Definiciones

Definición

Dado G = (N , A) una red dirigida, definimos en general una función f sobre el conjunto de los arcos del siguiente modo:

$$f : A \rightarrow Z^+$$
$$(i,j) \rightarrow f(i,j) := f_{ij} \qquad (1)$$

Donde Z^+: = { 0,1,2,3,.....} enteros no negativos.

Generalmente la función f representa **capacidad** (u), **costo** (c) o **flujo (x).**

Definición

Sea G = (N, A) una red dirigida y consideremos ahora un subconjunto S cualquiera de N ($S \subset N$). Además sea $\overline{S}$ = (N-S) entonces definimos el conjunto:

$$(S, N-S)= (S, \overline{S} :=\{(i,j)\in A \,/\, i\in S, j\in \overline{S} \} \qquad (2)$$

El cual representa el conjunto de arcos que salen o nacen en S entran o llegan a $\overline{S} = (N - S)$

Definición

Sea G = (N, A) una red dirigida y sea f : $A \rightarrow Z^+$ cualquier función como el definido en la ecuación (5), entonces definimos:

$$f(S,N-S)=f(S, \overline{S})=\sum_{i\in S}\sum_{j\in \overline{S}} f_{ij} = \sum_{(i,j)\in(S,\overline{S})} f_{ij} \qquad (3)$$

Esta cantidad representa la suma de los valores sobre los arcos considerados en (2). Este número puede ser visto como el valor total que sale de S y entra en $\overline{S} = (N - S)$ para simplificar las operaciones posteriores adoptaremos la siguiente notación:

$$f(S) := f(S, \overline{S}) = f(S, N-S) \quad (4)$$

A modo de ilustrar las definiciones anteriores consideremos la red de la figura (1), donde N = {1,2,3,4,5,6} y la función f = u (Capacidad) definida sobre A.

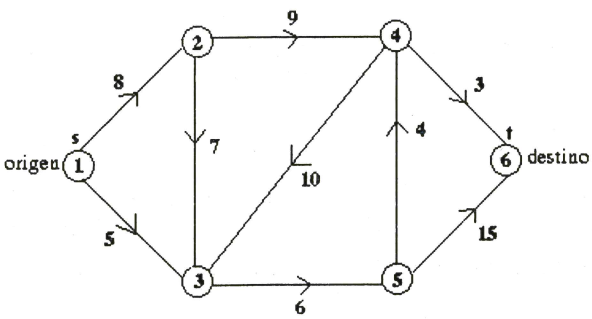

Fig. 1 Red Dirigida G=(N,A)

Ejemplo 1
Si consideramos: S={1,2,3}, entonces
$\overline{S}$ =(N-S) = {4,5,6}
$\rightarrow f(S, \overline{S}) = u(S, \overline{S}) = u(S) = u_{24} + u_{35} = 9+6=15$

Ejemplo 2
Por otra parte, si ahora consideramos: S = {1,3,4}, $\overline{S}$ =(N-S) = {2,5,6}, entonces:
$\rightarrow f(S, \overline{S}) = u(S, \overline{S}) = u(S) = u_{12} + u_{35} + u_{46} = 8+6+3=17$

Ejemplo 3
Sea S = {4}; $\overline{S}$ =(N-S)={1,2,3,5,6}, entonces :
$f(S, \overline{S}) = u(S, \overline{S}) = u(S) = u(\{4\}) = u_{43} + u_{46} = 10+3 = 13$

Ejemplo 4
Si S={4} $\overline{S}$ =(N-S)={1,2,3,5,6}, entonces:
$f(S, \overline{S}) = u(\overline{S}, S) = u(\overline{S}) = u_{24} + u_{54} = 9+4 = 13$

FORMULACIÓN DEL PROBLEMA

Sea la red dirigida G = (N, A) tomemos dos nodos arbitrarios diferentes: **nodo origen** denotado mediante $\underline{s}$; y **nodo destino** denotado mediante $\underline{t}(s \neq t)$.

Definición (FLUJO)

Dados una red G = (N, A); dos nodos s y t (origen y destino respectivamente) un **flujo** de s a t en G es una función:

$$x : A \rightarrow Z^{+} \quad (5)$$
$$(i, j) \rightarrow x(i, j) := x_{ij}$$

Tal que :

$$x(\{i\}) = x(N - \{i\}) \quad \leftrightarrow i \in N \setminus \{s,t\} \quad (6)$$

La relación (6) es un caso particular de la ecuacion (4) y es conocida como **condición de equilibrio de masa,** el cual significa que el valor que sale del nodo i (x{i}) es igual al valor total que entra al nodo i (x{N-{i}}) para todo nodo i diferente de s y t.

Definición:

Sean G = (N, A) una red; x un flujo de s a t en G, entonces el **valor del flujo** v(x) es definido mediante:

$$v(x) := x(\{s\}) = -x(N-\{s\}) = x(\{t\}) = -x(N-\{t\}) \quad (7)$$

El cual es interpretado como la cantidad de flujo que sale de s menos la cantidad de flujo que entra a s o él negativo de lo que sale de t menos lo que entra en t.

Consideremos ahora una función:

$$u : A \rightarrow Z^{+} \quad (8)$$
$$(i, j) \rightarrow u(i, j) := u_{ij}$$

la cual se denominará función **capacidad,** por otro lado u_{ij} representa la máxima capacidad que tiene el arco (i , j) esta función generalmente es dato en los diversos algoritmos y más aún en aplicaciones reales.

Definición:

Sean las funciones x (flujo) y u (Capacidad) sobre una red G = (N , A) se dice que un flujo x **respeta** u si:

$$x_{ij} \leq u_{ij} \; \forall (i, j) \in A \quad (9)$$

Definición (Flujo Máximo)

Sean G = (N , A) una red capacitada con capacidad, u, **s** y **t** dos nodos especiales (origen y destino respectivamente) y $x^0 : A \rightarrow Z^+$ un flujo de s a t en G. Se dice que x^0 es un **flujo máximo** si **respeta** u y su valor es máximo considerando todos los flujos de s a t en G que **respetan** u, es decir:

$v(x^0)=\max\{ v(x) / x$ es un flujo de s a t en G que respeta $u\}$

Ejemplo:

Consideremos el flujo x sobre la red de la Figura (1), además sean s = 1 (nodo origen) y t = 6 (nodo destino), tal como se muestra en la siguiente figura:

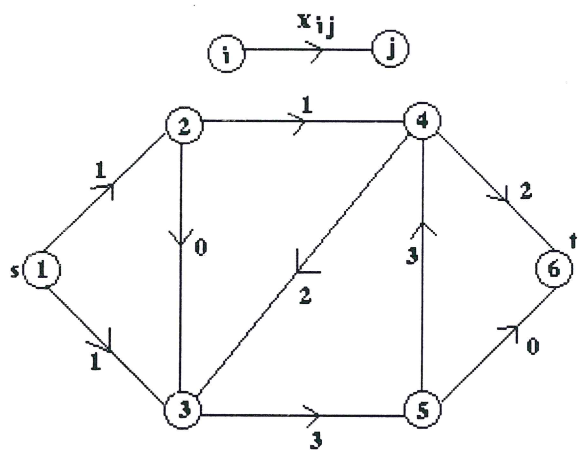

***Fig. 2** flujo de s=1 a t=6 que **respeta** u y cuyo valor es v(x) =2*

Consideramos ahora otro flujo x con s = 1 y t = 6 (origen y destino), tal como se muestra en la figura siguiente:

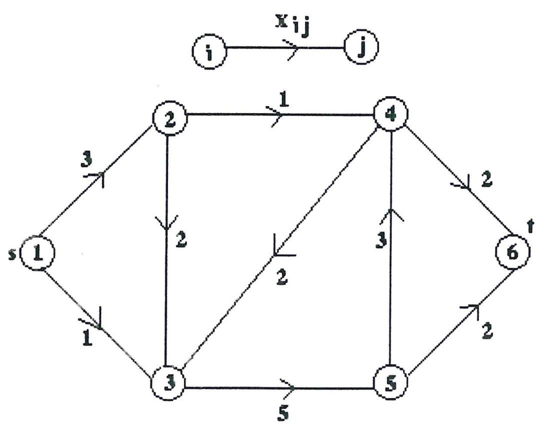

***Fig. 3** Flujo de s = 1 a t = 6 que **respeta** u y cuyo valor es v(x) =4.*

Ejemplo

x = 0 también es un flujo de s=1 a t=6, cuyo valor es v(0) = 0.

Es claro que dados una red G = (N , A) y una función capacidad $u: A \rightarrow Z^+$ existe un número finito de flujos x que **respetan** u (es decir $x_{ij} \leq u_{ij}$) y cada uno de ellos tiene un cierto valor. Con estos elementos estamos en condiciones de formular el problema que estudiaremos de aquí en adelante.

FORMULACIÓN DEL PROBLEMA DEL FLUJO MÁXIMO

Dados una red G = (N , A); una función **capacidad** $u: A \rightarrow Z^+$; dos nodos s y t (origen y destino respectivamente). Hallar un flujo $x: A \rightarrow Z^+$ de s a t que **respete** u ($x_{ij} \leq u_{ij}$) y cuyo valor sea máximo.

Formalmente:

Max v(x)

s.a

$$\sum_{\{j/(i,j)\in A\}} x_{ij} - \sum_{\{j/(i,j)\in A\}} x_{ij} = \begin{cases} v(x);\ si\ i = s \\ 0\ ;si\ \ i \in N-\{s,t\} \\ -v(x); si\ i = t \end{cases} \qquad (10)$$

$$0 \leq x_{ij} \leq u_{ij} \quad \forall (i, j) \in A \qquad (11)$$

En la siguiente sección definiremos algunos conceptos para proceder a formular los algoritmos para resolver el problema planteado.

CORTES Y RED RESIDUAL

A continuación definimos un tipo especial de corte que depende de dos nodos s y t llamado corte s-t.

Definición (Corte s-t)

Dados una Red G = (N , A) dos nodos s y t, un **corte** s-t es cualquier **corte** donde $s \in S$ y $t \in \overline{S}$ $(s \notin \overline{S}\ y\ t \notin S)$

Observaciones

- A un **corte** s-t también se le conoce como **corte que separa** s de t.

- Dado un par de nodos s y t es claro que existe un número finito de **cortes** s-t.

De la definición anterior, un **corte** s-t está caracterizado por la partición del conjunto de nodos N en los subconjuntos S y $\overline{S}$ donde s$\in$ S y t$\in$ $\overline{S}$.

Notación

Un corte que separa s y t será denotada por [S, $\overline{S}$],donde evidentemente s$\in$S y t $\in$ $\overline{S}$.

Veamos a continuación el siguiente ejemplo para ilustrar las definiciones anteriores.

Ejemplo

Consideremos la red de la Figura (1), donde s=1 y t=6, entonces S={1,2,3} y $\overline{S}$ = {4,5,6} constituye un **corte** s-t (corte que separa s y t). Similarmente T={1,3,5} y $\overline{T}$ = {2,4,5} también es un **corte** s-t tal como se muestra en la Figura (3).

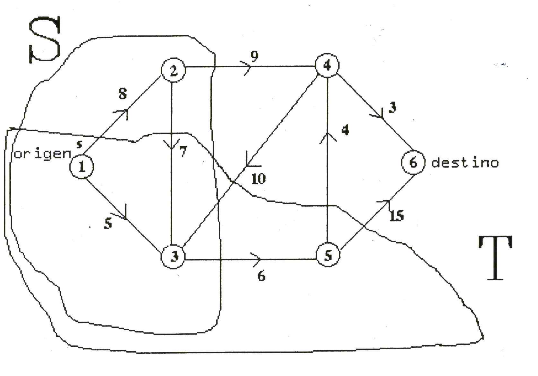

Fig. 4 Cortes [S, $\overline{S}$] y [T,$\overline{T}$] que separan s=1 y t =6

En la red anterior, para el corte[S, $\overline{S}$ se tiene el conjunto de arcos **hacia delante** (S, $\overline{S}$)={(4,3)}.Mientras que para el corte[T, $\overline{T}$] se tiene el conjunto de arcos **hacia delante** (T,$\overline{T}$)= {(1,2) , (5,4) , (5,6)} y el conjunto de arcos **hacia atrás** (T,$\overline{T}$)={(2,3) , (4,3)}

Definición (Capacidad de un corte s-t)

Dado el **corte** s-t [S, $\overline{S}$] definimos la capacidad u [S, $\overline{S}$] de dicho corte como la suma de las capacidades de los arcos **hacia delante,** es decir:

$$u[S,\overline{S}\,] := \sum_{(i,j)\in(S,\overline{S})} u_{ij} \qquad (12)$$

Definición (Corte Mínimo)

Se llama así a aquel corte s-t cuya capacidad es mínima entre todos los cortes s-t.

RED RESIDUAL

Dados una red G = (N , A) una función capacidad u: A$\rightarrow$Z$^+$ y un flujo x: A$\rightarrow$Z$^+$; definimos la **capacidad residual** como una función r sobre el conjunto de arcos definido mediante:

$$r : A \rightarrow Z^+$$

$$(i , j) \rightarrow r(i , j) := r_{ij}=(u_{ij} - x_{ij}) + x_{ij}$$

Por definición $r_{ij} \geq 0$, en términos prácticos r_{ij} representa el flujo adicional máximo que aún puede ser enviado a través de los arcos (i , j) y (j, i).

Definición (Red residual)

Dados una red G = (N , A) con capacidad u, un flujo x de s a t definimos y denotamos la red residual mediante:

$$G(x) = (N , \overline{A}) \text{ donde}$$
$$\overline{A} =\{ (i , j)\in A / r_{ij} > 0 \}$$

Es decir, la red residual G(x) es una red que tiene los mismo nodos que G y arcos cuya capacidad residual es estrictamente positiva (r_{ij} > 0).

Observaciones

a) Si x = 0 entonces $r_{ij} = u_{ij}$ $\forall$ (i,j) $\in$ A, en consecuencia la red residual para este flujo coincide con la red original, es decir G(0) = G.

b) $\begin{cases} r_{ij} = u_{ij} - x_{ij} + x_{ij} \\ r_{ij} = u_{ij} - x_{ij} + x_{ij} \end{cases} \rightarrow r_{ij} + r_{ij} = u_{ij}$ =constante

c) Dados una red capacitada G = (N , A) y un flujo x , existe una correspondencia biunívoca entre la red original (G) y la red residual G(x) en el siguiente sentido:

- Dados G = (N , A), u y x ; G(x) es obtenido mediante:

$$r_{ij} := (u_{ij} - x_{ij}) + x_{ij}$$

mientras que

- Dados G(x) y u ; x (y con él G) es obtenido mediante:

$$x_{ij} := \max_{(i,j)\in A}\{0, u_{ij} - r_{ij}\}$$

Definición (Capacidad residual de un corte s-t)

Definimos la capacidad residual $r[S,\overline{S}]$ de un corte s-t como la suma de las capacidades residuales sobre los arcos adelante del corte correspondiente es decir:

$$r[S,\overline{S}] = \sum_{(i,j)\in(S,\overline{S})} r_{ij} \qquad (13)$$

Ejemplo

Sean G = (N , A) una red con capacidad u; y x un flujo mostrados en la figura (13); se observa que existe una correspondencia entre G y G(x)

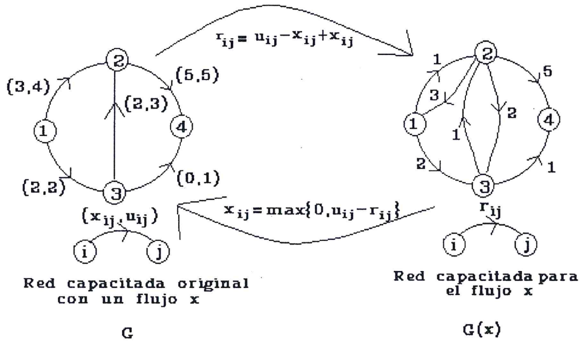

Fig. 5 Correspondencia biunívoca entre las redes G y G(x).

La siguiente proposición relaciona los conceptos de corte$(S,\overline{S})$ y flujo x.

Proposición

Sean G = (N , A) una red capacitada, los nodos s (origen) y t (destino); un flujo x de s a t cuyo valor es v(x) y $[S,\overline{S}]$un corte que separa s y t entonces:

$$v(x) = \sum_{i\in S}\left[\sum_{\{j/(i,j)\in A} x_{ij} - \sum_{\{j/(i,j)\in A} x_{ij}\right] = \sum_{(i,j)\in(S,\overline{S})} x_{ij} - \sum_{(i,j)\in(\overline{S},S)} x_{ij}$$

Teorema

El valor de cualquier flujo es menor o igual a la capacidad de cualquier corte en la red.
Para la prueba ver [1].

Corolario 3

Si x_0 es un flujo de s a t y
$[S_0, \overline{S}_0]$ es un corte s-t donde:

$$v(x_0) = u[S_0, \overline{S}_0]$$

entonces x_0 es un flujo cuyo valor es máximo y $[S_0, \overline{S}_0]$ es un corte cuya capacidad es mínima,

Corolario

Si $[S_0, \overline{S}_0]$ es un corte que separa s de t, donde se cumple que:

$$r_{ij} = 0 \qquad \forall (i,j) \in (S,\overline{S})$$

entonces $[S,\overline{S}]$es un corte de capacidad mínima, además el flujo x calculado según:

$$x_{ij} := \max\{u_{ij} - r_{ij}, 0\}$$

Definición

Dados dos flujos x e y cuyos valores son respectivamente v(x) y v(y), decimos que y mejora x si v(y) > v(x), dicha mejora es denotada por Δv con $\Delta v = v(y) - v(x) > 0$.

Los algoritmos que estudiaremos a continuación trabajan del siguiente modo: en cada iteración se tiene un flujo x de valor v(x) y en el siguiente paso se trata de mejorar este flujo, es decir construir otro flujo y con valor v(y) tal que
v(y) >v(x), el siguiente corolario establece una cota superior para Δv.

Corolario

Si x es un flujo de valor v(x), entonces el flujo adicional que aún puede ser enviado del origen s al destino t está limitado superiormente por la capacidad residual de cualquier corte, esto es:

$$\Delta v \leq r[S, \overline{S}\,]$$

MÉTODO GENÉRICO DE CAMINOS AUMENTANTES

A continuación describimos el método más sencillo e intuitivo para resolver el problema del flujo máximo .

Entrada

Una red capacitada G = (N , A) con capacidad u y dos nodos distintos s y t (origen y destino respectivamente).

Salida

Un flujo x de s a t que respeta u cuyo valor es máximo.

```
x:=0
mientras existe un camino aumentante de s a t en G(x) hacer:
    Identificar un camino aumentante P en G(x)
    Calcular δ =min{r_ij / (i,j) ∈ P}
    para cada ( i , j) ∈ P hacer:
        r_ij : = r_ij - δ;
        r_ij : = r_ij + δ
para cada arco (i , j) en A hacer:
    x_ij:=max{u_ij-r_ij , 0}
Retornar x
```

Fig. 6 Algoritmo genérico de caminos aumentantes.

Ilustramos el método anterior con el problema de flujo máximo planteado en la Figura 7(a) con origen en el nodo 1 y destino en el nodo 4.

Dado que en la red residual G(x) figura 7(d) no existe camino de s a t, entonces el método termina calculando el flujo para cada arco según la fórmula.

$$xij:=\max\{0 , u_{ij} - r_{ij}\} \text{ para todo arco } (i , j)$$

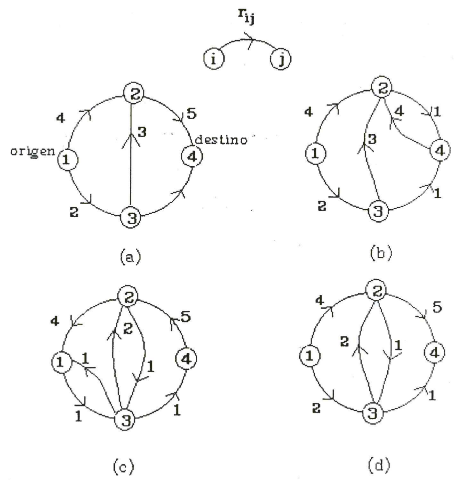

Fig 7 Ilustración del método genérico de caminos aumentantes (a) Red residual para el flujo x = 0; (b) Red residual luego de aumentar 4 unidades de flujo a través del camino 1-2-4; (c) Red residual después de enviar 1 unidad de flujo a través del camino 1-3-2-4 (d) Red residual después de enviar 1 unidad de flujo a través del camino 1-3-4, note que ya no existe camino de 1 hacia 4 en la red.

El flujo máximo obtenido se muestra en la siguiente figura:

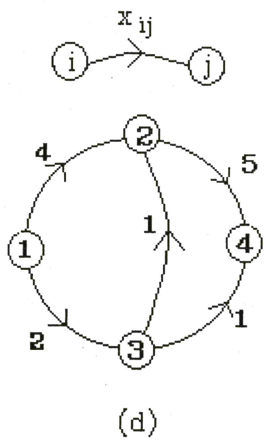

Fig. 8 Flujo máximo obtenido.

ALGORITMO DE ETIQUETAS

Entrada:

Una red G = (N , A) una función capacidad u: A→ Z^+ y dos nodos s y t.

Salida

Un flujo x de s a t que respeta u y cuyo valor es máximo.

```
r:=u;
repetir
    pred(j)=0 para cada j∈N
    S:={s}; LIST :={s};
    mientras (LIST≠∅) y (t∉S) hacer:
        retirar un elemento i de LIST;
        para cada (i , j)∈A(i) hacer:
            si (rij > 0) y (j ∉ S)
                entonces
                    pred(j):=i;
                    S:=S∪{j};
                    LIST:=LIST∪{j}
    si t ∈ S entonces
        use pred para encontrar un camino orientado P de s a t;
        δ=min{rij /  (i , j) es arco de P}
        para cada arco (i , j) de P hacer:
            rij : =rij - δ
            rij : =rij + δ
hasta que ( LIST = ϕ )
para cada arco (i , j) ∈ A hacer:
xij : =max{uij - rij , 0}
```

Fig. 9 Algoritmo de Etiquetas.

CORRECTITUD Y COMPLEJIDAD

A continuación estableceremos las razones por las cuales el flujo que se obtiene al terminar el algoritmo de etiquetas es máximo (correctitud) y cuánto tiempo tarda el algoritmo en resolver el problema (complejidad).

Correctitud

Para demostrar la correctitud del algoritmo, notemos que el algoritmo mantiene las siguientes propiedades invariantes al inicio de cada iteración, (en particular en la última).

a) Un flujo x, definido mediante:
$$x_{ij} := \max\{u_{ij} - r_{ij}, 0\}$$
b) Para cada k en S, pred determina un camino orientado P de s a k, cada uno de cuyos arcos tiene capacidad residual positiva.
c) O el valor de x aumenta o el valor de x no se altera y el conjunto S \ LIST aumenta.

Al terminar el algoritmo ocurre que el nodo destino t no puede ser etiquetado, si consideramos el conjunto S* que consiste de los nodos etiquetados, además $\overline{S}^* = N \setminus S^*$ es el conjunto de nodos no etiquetados, luego entonces es claro que $s \in S^*$ y $t \in \overline{S}^*$, es decir, $[S^*, \overline{S}^*]$ es un corte que separa s y t. Consideremos también el flujo x* obtenido al terminar el algoritmo, calculado según:

$$x^*_{ij} := \max\{u_{ij} - r_{ij}, 0\} \text{ para todo arco } (i, j) \in A$$

Dado que t no puede ser etiquetado entonces debe ocurrir que:

$$r_{ij} = 0 \text{ para todo arco } (i, j) \text{ de } (S^*, \overline{S}^*) \qquad (14)$$

pues de lo contrario, si existiese un arco $(i_0, j_0) \in (S^*, \overline{S}^*)$ tal que $r_{i_0 j_0} > 0$ entonces el algoritmo aún no terminaría lo cual no puede ser, pues estamos asumiendo que el algoritmo terminó.

Un corolario anterior y la ecuación (14) establecen que x* es un flujo de s a t cuyo valor es máximo, lo cual muestra la correctitud de algoritmo de etiquetas.

Nótese que el algoritmo además de hallar el flujo máximo, nos devuelve un corte $[S^*, \overline{S}^*]$ cuya capacidad es mínima.

La prueba de correctitud del algoritmo anterior nos permite establecer dos resultados teóricos que enunciamos a continuación:

TEOREMA (TEOREMA DEL FLUJO MÁXIMO-CORTE MÍNIMO)

Sean dados una red capacitada G = (N , A) dos nodos, origen(s) y destino (t), entonces el valor del flujo máximo de s a t en es igual a la capacidad del corte mínimo.

Teorema

Dados una red capacitada G = (N , A) dos nodos : origen (s) y destino(t); Un flujo x* en G es máximo si y solo si no existe ningún camino aumentante en la red residual G(x*).

COMPLEJIDAD (EN EL PEOR CASO)

$$T(m,n) = O(m)O(nU) = O(mnU)$$

Los detalles del cálculo pueden ser consultados en [2].

Esta complejidad no es polinomial pues podría ocurrir, por ejemplo, que $U=2^n$, fenómeno que estudiaremos en la siguiente sección.

VENTAJAS Y DESVENTAJAS

Entre las pocas ventajas que podemos mencionar para este algoritmo se encuentran su facilidad de implementación, su simplicidad y principalmente su utilidad en la demostración del teorema del flujo máximo-corte mínimo y el hecho de ser el punto de partida para el desarrollo de otros algoritmos mas eficientes.

Por otra parte de las desventajas más saltantes son :

1. El algoritmo no es polinomial (su complejidad es O(mnU), esto ocurre principalmente cuando el valor de U es demasiado grande; por ejemplo, si $U=2^n$ el algoritmo sería exponencial y en consecuencia intratable en la práctica . Ilustramos este caso con el problema del flujo máximo propuesto en la Figura (16).

2. El algoritmo es "olvidadizo" en el siguiente sentido: en cada iteración, el algoritmo genera una serie de etiquetas, los cuales contiene información acerca de los caminos aumentantes desde la fuente (origen) hacia los otros nodos. La implementación descrita elimina toda esta información al terminar un iteración, para iniciar la siguiente, realizando nuevamente los cálculos anteriores, cuando lo ideal sería utilizar la información entre una iteración y la siguiente.

A continuación ilustramos por qué el algoritmo no es polinomial , consideremos el problema del flujo máximo dado en la figura 16(a) donde el nodo origen es 1 y el nodo destino es 4, nótese que ésta red tiene capacidades de orden bastante grandes (es decir, U es grande). Podemos deducir para este caso especial que el número de iteraciones es igual a $2 \times 10^6 = 2x\ U = O(10^6)$, luego el tiempo que toma al algoritmo resolver el problema será del orden $O(10^6) = O(U)$, lo cual es demasiado alto.

Esto se debe a que la complejidad del algoritmo O(mnU), depende explícitamente del número U sobre el cual no tenemos ningún control, entonces debemos diseñar algoritmos cuya complejidad sea más manejable y para eso la complejidad debe depender de Log(U) en lugar de depender de U o mejor aún ni siquiera depender de U.

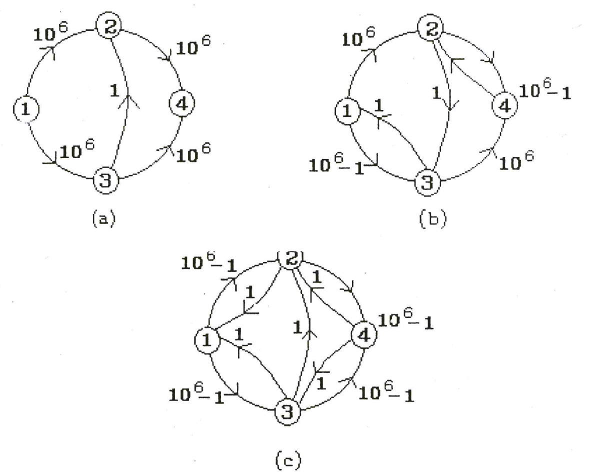

Fig. 10 *(a) Problema del flujo máximo dado con origen en 1 y destino en 4, (b) Red residual luego de enviar una unidad de flujo a través del camino aumentante 1-3-2-4; (c) Red residual luego de enviar una unidad de flujo a través del camino aumentante 1-2-3-4; el algoritmo continua así hasta saturar todas las capacidades.*

REFERENCIAS

1. **R. K. Ahuja, T. L. Magnanti, J. B. Orlin,** Network Flows Theory Algorithms and Aplications. Prentice Hall, 1993.
2. **Rósulo Pérez Cupe,** Algoritmos Polinomiales parael problema del flujo máximo. Tesis de Licenciatura, 2001.
3. **T. H. Cormen, C. E. Leiserson, R. L. Rivest,** Introduction to algorithms. MIT Press and McGraw-Hill, 1990.