IN7K3-1 Flujos en Redes - Primavera 2011	8 de noviembre 2011

Control 1-Pauta

Profesor: Fernando Ordóñez *Auxiliar: Renaud Chicoisne*

P1.

Suponga que son conocidos los costos de las rutas mínimas para ir de un nodo s a todos los nodos de la red, denotados por $d_i^s, \forall i \in V$, y que existen arcos (pero no ciclos) de costo negativo. Considere el problema de calcular las rutas mínimas desde otro nodo s' al resto de los nodos. Para resolverlo, se propone la siguiente modificación de los costos:

$$c_{ij}^s = c_{ij} + d_i^s - d_j^s, \forall (i,j) \in A$$

a) Muestre que $c_{ij}^s \geqslant 0, \forall (i,j) \in A$

Se sabe que al óptimo, la distancia de s a cualquier nodo i no puede ser mejorada entonces:

$$\begin{array}{c} d_j^s \leqslant c_{ij} + d_i^s, \forall (i,j) \in A \\ \text{ie: } c_{ij}^s \geqslant 0, \forall (i,j) \in A \end{array}$$

b) Pruebe que al resolver el problema con costos modificados se obtienen exactamente las mismas rutas mínimas que se hubieran obtenido considerando los costos originales.

Sea $p = (i_0 = s', i_1, ..., i_{k-1}, i_k = t')$ un camino desde un nodo s' hasta un nodo t'. Calculamos su costo modificado $C^s(p)$:

$$\begin{aligned} C^s(p) &= \sum_{j=0}^{k-1} c_{i_j i_{j+1}}^s = \sum_{j=0}^{k-1} \left(c_{i_j i_{j+1}} + d_{i_j}^s - d_{i_{j+1}}^s \right) \\ &= \sum_{j=0}^{k-1} c_{i_j i_{j+1}} + \sum_{j=0}^{k-1} \left(d_{i_j}^s - d_{i_{j+1}}^s \right) \\ &= C(p) + d_{s'}^s - d_{t'}^s \end{aligned}$$

Entonces, el problema de encontrar el camino más corto con los costos modificados es lo mismo que con los costos originales más una constante que depende de los nodos extremos pero no del camino tomado.

c) Qué ventajas tiene realizar esta modificación en los costos?

Modificar los costos de esta manera les vuelve todos positivos sin cambiar las soluciones. Por lo tanto, se puede aplicar el algoritmo de Dijkstra en vez de un label correcting, lo que es más eficiente.

P2.

Demuestre brevemente o de un contraejemplo para las siguientes preguntas/afirmaciones:

(a) BONUS: Muestre que un algoritmo de búsqueda (depth first search - DFS o breadth first search - BFS) es $O(m)$.

El algoritmo genérico de búsqueda es el siguiente:

LISTA $\leftarrow \{s\}$;
marcar s;
Mientras LISTA $\neq \emptyset$:
    Elegir algún $i \in$LISTA;
    LISTA $\leftarrow$LISTA $\setminus \{i\}$;
    $\forall (i,j) \in \delta_i^+$:
        If j no marcado:
            marcar j;
            LISTA $\leftarrow$ LISTA $\cup \{j\}$;

BFS and DFS teniendo dos maneras distintas de elegir el nodo i de la lista.

Todos los nodos entran una sola vez a LISTA, y para cada uno de ellos, se revisan a lo más todos los arcos salientes, ie: $O(n) + \sum_{i \in V} O(\delta_i^+) = O(n) + O(m) = O(m)$.

(b) (V/F) En un problema de camino mínimo, si amplificamos todos los costos por $\alpha > 1$ de manera que $c'_{ij} = \alpha c_{ij}$, entonces el árbol de caminos mínimos se mantiene igual.

Sea $p = (i_0 = s, i_1, ..., i_{k-1}, i_k = t)$ un camino desde un nodo s hasta un nodo t. Calculamos su costo modificado $C^\alpha(p)$:

$$\begin{aligned} C^\alpha(p) &= \sum_{j=0}^{k-1} c'_{i_j i_{j+1}} \\ &= \alpha \sum_{j=0}^{k-1} c_{i_j i_{j+1}} \\ &= \alpha C(p) \end{aligned}$$

Entonces minimizar el costo de un camino p con los costos modificado es lo mismo que minimizar el costo del camino por α, lo que no depende del camino empleado. VERDADERO.

(c) (V/F) Si el arco (i,j) pertenece al unico corte mínimo y aumentamos su capacidad en δ, entonces el flujo máximo aumenta en δ.

En el siguiente grafo estan indicados el flujo máximo f_{ij} y la capacidad u_{ij} de cada arco (i,j) entre corchetes: $[f_{ij}, u_{ij}]$.

$$1 \xrightarrow{[1,2]} 2 \xrightarrow{[1,1]} 3$$

El único corte mínimo que existe aquí es el arco $(2,3)$, de capacidad $u_{2,3} = 1$. Si le aumentamos la capacidad de $\delta = 2$, tendremos el siguiente flujo máximo:

$$1 \xrightarrow{[2,2]} 2 \xrightarrow{[2,3]} 3$$

Cuyo valor es $2 \neq 1 + \delta = 3$. FALSO.

(d) (V/F) Si aumentamos la capacidad de todos los arcos en δ entonces el flujo máximo aumenta de al menos δ.

Sea $p = (i_0 = s, i_1, ..., i_{k-1}, i_k = t)$ un camino desde la fuente s hasta el sumidero t y $\left(f_{ij}^*\right)_{(i,j)\in E}$ el flujo máximo pasando por la red. Por optimalidad del flujo, no hay camino aumentante en el grafo residual, por lo tanto:

$$\min_{(i,j)\in p} \left\{c_{ij} - f_{ij}^*\right\} = 0$$

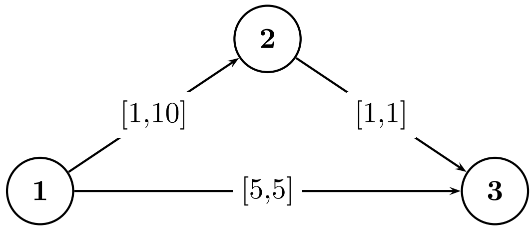
Entonces:

$$\min_{(i,j)\in p} \left\{ c_{ij} + \delta - f^*_{ij} \right\} = \delta$$

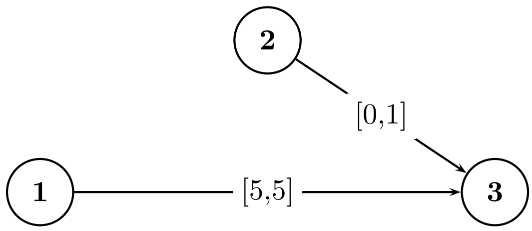
En consecuencia, existe por lo menos un camino aumentante donde se puede empujar δ unidades adicionales de flujo. El nuevo flujo máximo tiene por lo tanto un valor de por lo menos δ unidades más. VERDADERO.

(e) Un arco más vital es un arco cuya eliminación causa la mayor reducción en el valor del flujo máximo. (V/F) un arco más vital es un arco con el mayor u_{ij}.

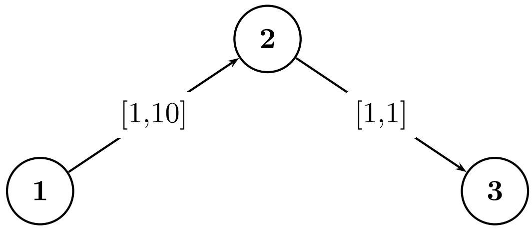
En el grafo que sigue estan indicados el flujo máximo f_{ij} y la capacidad u_{ij} de cada arco (i, j) entre corchetes: $[f_{ij}, u_{ij}]$.



El flujo máximo pasando de 1 a 3 vale 6 y el arco con mayor capacidad u_{ij} es $(1, 2)$. Si le borramos tenemos el siguiente flujo máximo:

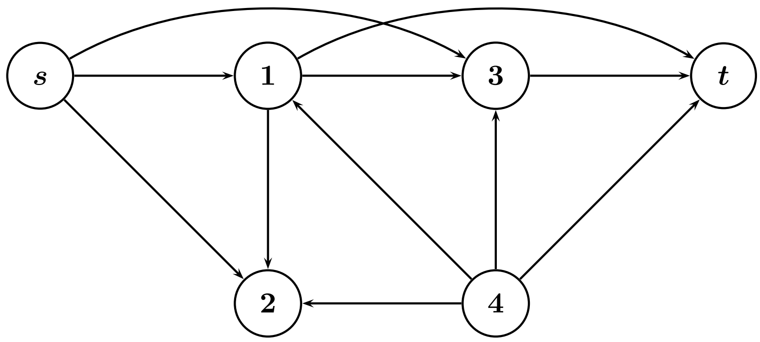


Que nos entrega un flujo máximo de 5 unidades. Sin embargo, si se hubiese removido el arco $(1, 3)$ en vez del $(1, 2)$, hubiesemos encontrado el siguiente flujo máximo:



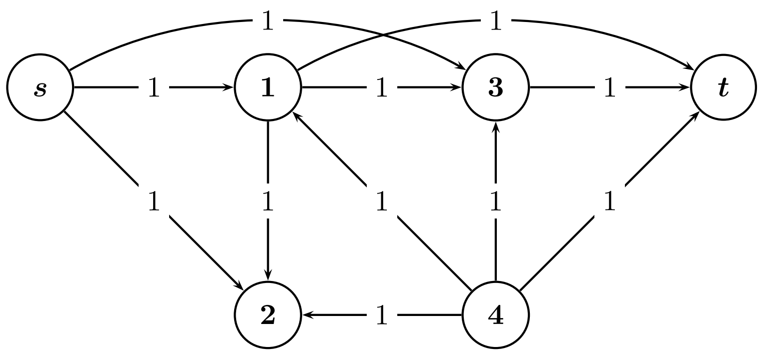
Que tiene valor $1 < 5$. Por lo tanto, el arco $(1, 2)$ no era un arco más vital. FALSO.

3. Considere la siguiente red de transmisión eléctrica, donde la dirección indica el sentido del flujo eléctrico y suponemos que todos los arcos tienen suficiente capacidad para transmitir toda la demanda.



(a) Formule el problema de encontrar el menor número de links que deben romperse para desconectar s de t como un problema de flujo máximo. Explique porque resuelve el problema en cuestion.

Asignamos a cada arco una capacidad de 1. En el ejemplo esto nos da:



Encontrar el flujo máximo entre s y t nos entregará el valor del corte mínimo entre s y t. Además, impusimos capacidad 1 a todos los arcos entonces el valor del corte mínimo es igual al número de arcos del corte, ie: el menor número de arcos para desconectar s de t.

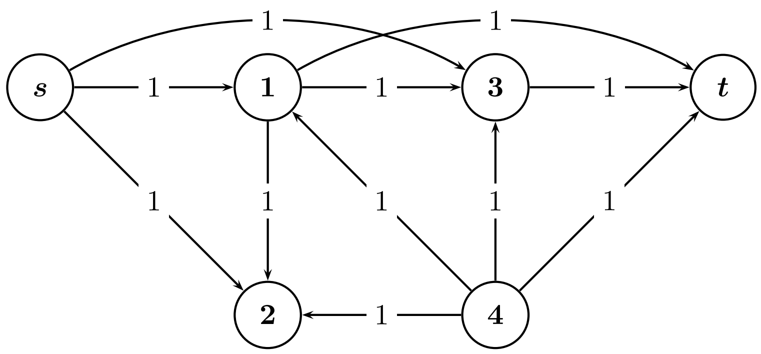
Esta formulación resuelve el problema en cuestión porque si se cortan estos arcos, entonces cortamos todos los caminos de s a t a menor costo (en terminos de número de arcos cortados).

(b) Resuelva este problema de flujo máximo con el algoritmo genérico de camino aumentante. Describa en cada iteración el grafo residual y el camino encontrado. Que arcos deben cortarse para separar s de t? Es esta solución única?

La capacidad de cada arco (i, j) en el grafo residual esta indicada en cada arco. Si la capacidad residual de un arco vale cero, no aparece en la figura.

Initialización:

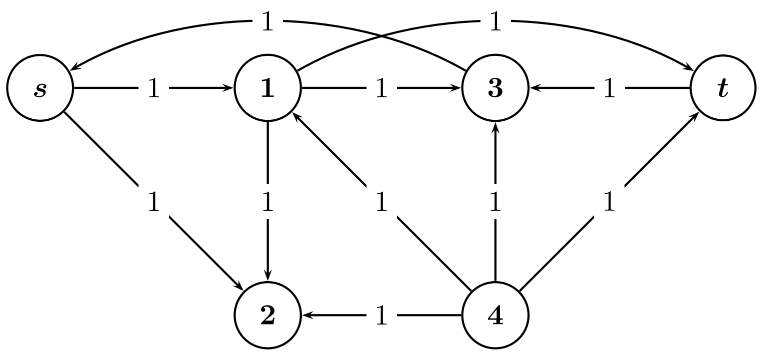
$f_{ij} := 0; \forall (i, j) \in E$
$u^f_{ij} := u_{ij}; \forall (i, j) \in E$



Iteración 1:

El camino $\{s, 3, t\}$ es aumentante y se puede empujar 1 unidad de flujo.

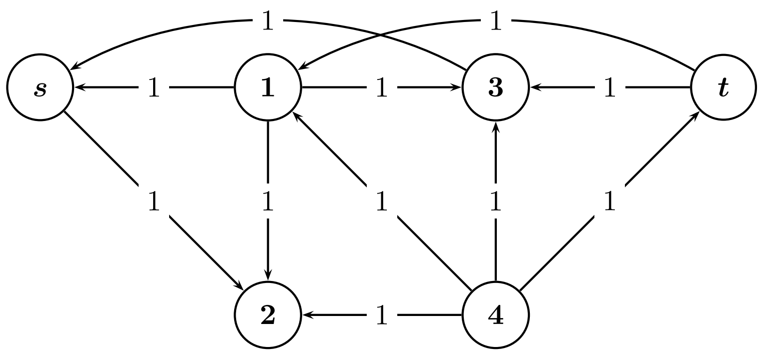
Se actualizan los flujos y las capacidades residuales de este camino: $f_{s,3} := f_{s,3}+1 = 1$; $u^f_{s,3} := u^f_{s,3} - f_{s,3} = 0$; $f_{3,t} := f_{3,t} + 1 = 1$; $u^f_{3,t} := u^f_{3,t} - f_{3,t} = 0$;



Iteración 2:

El camino $\{s, 1, t\}$ es aumentante y se puede empujar 1 unidad de flujo.

Se actualizan los flujos y las capacidades residuales de este camino: $f_{s,1} := f_{s,1}+1 = 1$; $u^f_{s,1} := u^f_{s,1} - f_{s,1} = 0$; $f_{1,t} := f_{1,t} + 1 = 1$; $u^f_{1,t} := u^f_{1,t} - f_{1,t} = 0$;



Iteración 3:

No hay caminos aumentantes entre s y t: el algoritmo para con el flujo actual que vale 2.

El corte $S = \{s, 2\}$ separa el nodo s del nodo t. Los arcos que van de S a $\bar{S} = V \setminus S$ son $(s, 1)$ y $(s, 3)$ y la suma de sus capacidades vale $u_{s,1} + u_{s,3} = 2$ lo que es exactamente el valor del flujo encontrado y por lo tanto confirma la optimalidad del flujo.

No es el único corte que se puede encontrar: en efecto, el corte $\bar{S} = \{4, t\}$ separa el nodo s del nodo t. Los arcos que van de S a $\bar{S} = V \setminus S$ son $(1, t)$ y $(3, t)$ y la suma de sus capacidades vale $u_{1,t} + u_{3,t} = 2$.