PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DPTO. DE INGENIERÍA DE TRANSPORTE Y LOGÍSTICA

Curso: ICT2233
Semestre: 2024-01
Profesor: Mathias Klapp

EXAMEN
FLUJO EN REDES - ICT2233

TIEMPO: 135 MINUTOS

- *La prueba tiene 70 puntos distribuidos entre todas sus preguntas y la nota se calcula con 60 puntos. Es decir:*

$$\textit{Nota Examen} = \min(\textit{puntos obtenidos}, 60)/10 + 1$$

  se logra con 60 puntos.
- *Sea estratégico, hay algunas preguntas más difíciles que otras.*
- *La prueba es sin apuntes y/o equipos electrónicos. Se permite los resúmenes escritos a mano.*

Pregunta 1 (21 puntos, 3p c/u,)

Indique para cada una de las siguientes afirmaciones si es **verdadera** o podría ser **falsa**. Argumente matemáticamente cada alternativa que usted indique como verdadera. Alternativamente, encuentre un contraejemplo para las afirmaciones que usted señale falsas.

(a) (I1) Es posible construir un grafo no dirigido con tres nodos que poseen grado 3, dos nodos de grado 2, cuatro nodos de grado 4 y un nodo de grado 6.

Respuesta: Falso, es imposible al haber un número impar de nodos de grado impar.

(b) (I1) En un grafo no dirigido, el conjunto de arcos de una caminata cualquiera se pueden descomponer en la unión de un número finito de ciclos y de una sola ruta.

Respuesta: Verdadero. Considere una caminata cualquiera de p nodos entre $i_1 \in N$ e $i_p \in N$ dada la secuencia $(i_1, i_2, \ldots, i_p)$. Hay dos posibilidades:

- La caminata no repite nodos y, por lo tanto, es una sola ruta.
- La caminata repite al menos un nodo. Considere que $j \in N$ es el primer nodo repetido en la secuencia en las posiciones k_1 y k_2 tal que $1 \leq k_1 < k_2 \leq p$ y que $j = i_{k_1} = i_{k_2}$. En este caso se puede remover de la caminata a los arcos del ciclo $C = (i_{k_1}, i_{k_1+1}, \ldots, i_{k_2})$. La estructura sobrante $(i_0, i_1, i_2, \ldots, i_{k_1}, i_{k_2+1}, i_{p-1})$ es una caminata con un ciclo menos. Basta repetir el mismo argumento removiendo ciclos hasta que no queden más y sea una sola ruta.

NOTA: : Cualquier argumento en esta línea es razonable.

(c) (I1) Para todo árbol $T = (N, A)$ existe un corte $Q \subseteq N$ tal que solo un arco del árbol cruza el corte (es decir, $|\delta_Q| = 1$).

Respuesta: Tome $Q = \{i\}$, donde i es un nodo "hoja" del árbol (Todo árbol tiene ≥ 2 hojas). Se cumple que $|\delta_i| = 1$.

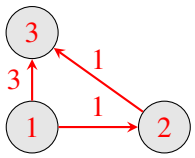
(d) (I1) Si $G = (N, A)$ es un árbol y $|N| \geq 3$, entonces para cualquier arco $e \notin A$ el grafo $(N, A \cup \{e\})$ tiene un ciclo.

Respuesta: Verdadero. Considere un arco $e = \{i, j\} \notin A$. Como G es un árbol, existe exactamente una ruta $(i = i_1, i_2, \ldots, i_{k-1}, i_k = j)$ entre i y j. Si agrega el arco $\{i, j\}$, entonces inmediatamente se forma el ciclo: $(i, i_2, \ldots, i_{k-1}, j, i)$

(e) (I2) Considere un problema de ruta mínima desde un nodo s a otro t sobre un grafo dirigido y suponga que la ruta mínima es única y usa p arcos. Si el costo de cada arco de la red aumenta en $k > 0$ unidades, entonces el costo de la ruta mínima aumentará en $k \times p$.

Respuesta: Falso. La afirmación es correcta solo si la ruta mínima no cambia con la alteración de costos.

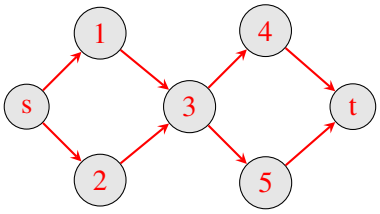
Considere el siguiente caso de 1 a 3:



La ruta mínima es $1-2-3$. Al sumar $k=2$ a cada arco, la ruta mínima cambia a $1-3$ y el costo de ruta mínima aumenta de 2 a $3+2$ (3 unidades). En este caso $3 < k \times p = 4$.

(f) En un grafo dirigido, el número mínimo de arcos a remover para que un nodo s no acceda a otro nodo t es menor o igual al número máximo de rutas disjuntas en nodos desde s hacia t.

Respuesta: Falso. Por dualidad, el número mínimo de arcos a remover para que un nodo s no acceda a otro nodo t es igual al número máximo de rutas disjuntas en arcos. Considere el siguiente ejemplo que posee dos rutas disjuntas en arcos entre s y t, pero una sola ruta disjunta en nodos.



(g) Considere un problema de flujo máximo desde $s \in N$ hacia $t \in N$ sobre un grafo dirigido $G = (N, A)$, con capacidades en arcos $u_a \in \mathbb{Z}_+, a \in A$. Siempre existe un vector óptimo de flujo $x^* \in \mathbb{R}^m$ tal que no posee flujo positivo sobre ciclos dirigidos.

Respuesta: Verdadero. Consideremos (por contradicción) que no es cierta y que solo existen flujos óptimos $x*$ con al menos un ciclo $C \subset A$ de flujo positivo. El ciclo no puede tocar a s o t (la solución no sería óptima ¿PQ?). Reste $\delta = \text{mín}_{a \in C} x_a$ a x_a para cada arco $a \in C$. El resultado es otra solución factible de igual valor de flujo entre s a t sin flujo positivo por C. Repetir hasta eliminar todos los ciclos y se obtiene una solución óptima sin ciclos de flujo positivo.

Pregunta 2 (10 puntos, I2)

Un procesador de texto, por ejemplo Word o LaTeX, está equipado con un algoritmo para justificar texto. Un texto es una secuencia de n palabras separadas por espacios y sin saltos de línea. Por ejemplo, un texto plano es:

"Estamos aprendiendo a justificar texto. ¡Nunca pensé que tenía relación con flujo en redes!"

El algoritmo que justifica texto decide cuántos saltos de línea colocar y dónde colocarlos entre palabras consecutivas del texto para embellecer y ajustarlo al máximo a sus márgenes izquierdo y derecho.

Para modelar este problema, suponga que un texto consiste en una secuencia de n palabras ordenadas desde la palabra 1 hasta la n. En nuestro ejemplo hay 14 palabras: "Estamos", "aprendiendo", "a", "justificar", "texto.", "¡Nunca", "pensé", "que", "tenía", "relación", "con", "flujo", "en" y "redes!". Note que hay palabras que comienzan o terminan con punto (.), coma (,), pregunta (¿,?), exclamación (¡,!) y otros símbolos ortográficos que no son parte del abecedario. En este caso el símbolo pasa a ser parte de la palabra a la que está unido. Por ejemplo, "texto." es una palabra.

El algoritmo que justifica el texto requiere con parámetro de entrada una matriz triangular superior $b_{ij} > 0$ para cada $i = 1, \ldots, n$ y $j = i, \ldots, n$. Cada valor b_{ij} indica un valor de "belleza" para una potencial línea que de texto que comienza justo antes de la palabra i y termina justo después de la palabra j. Básicamente, mide que tan bien podría ajustarse esta línea a sus márgenes. Mientras mayor sea b_{ij}, más bella y ajustada será la línea. Asuma también que existe otro parámetro $a_{ij} \in \{0,1\}$ que vale uno si es que la línea asociada cabe entre los márgenes de la hoja. Si no cabe, entonce no se puede usar dicha línea.

El objetivo de la rutina computacional es cortar el texto en una o más líneas factibles para hacerlo lo más bello posible (medido como la suma de las bellezas de cada una de sus líneas de texto).

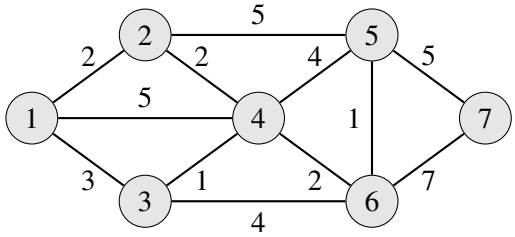
Formule el problema que maximiza la belleza del texto escogiendo donde colocar cada salto de línea como una instancia de uno de los problemas visto en este curso. Defina cada elemento del problema (nodos, arcos, costos, etc...).

Respuesta: Forme los siguientes conjuntos de nodos $N = \{0, \ldots, n\}$ y arcos $A = \{(i,j) : i, j \in N : j > i, a_{i+1,j} = 1\}$. El arco (i,j) representa la línea que comienza en la palabra $i+1$ y que termina con la j. Sea c_a, el costo de cada arco $a \in A$ definidos como $c_{i,j} = b_{i+1,j}$.

El algoritmo de justificación es un problema de ruta máxima desde 0 hacia n sobre un grafo acíclico.

Pregunta 3 (10 puntos)

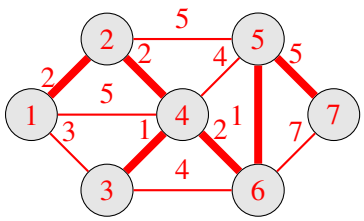
Considere siguiente el grafo no dirigido cuyos costos están detallados sobre sus arcos:



La suma de todos los costos es 41.

(a) (4p, I1) Determine un AMEM. Aplique un algoritmo válido.

Respuesta: Se puede aplicar Prim, Kruskal, Reverse deleting o cualquier algoritmo para AMEM. Esta es la solución óptima a costo 13:



(b) (6p) Determine el costo de una solución de Cartero Chino, es decir, el mínimo costo de una caminata cerrada que visita a todo arco del grafo al menos una vez. Aplique un algoritmo válido.

Respuesta: Hay 4 nodos de grado impar: $\{1,2,3,4\}$, por lo que hay que obtener rutas mínimas entre estos 4.

Desde 1, la ruta mínima a 2 cuesta 2, a 3 cuesta 3, a 4 cuesta 4. Desde 2, la ruta mínima a 3 cuesta 3 y a 4 cuesta 2. Finalmente, la ruta mínima entre 3 y 4 cuesta 1.

Luego, el mejor emparejamiento de estos 4 nodos es: 3 con 4 a costo 1 y 1 con 2 a costo 2. El costo del Matching es 41, por lo que es costo del PCC será 44.

Pregunta 4 (12 puntos, 4 c/u)

Responda a cada una de las siguientes preguntas de forma independiente.

Parte a (I2)

Considere un grafo dirigido $G = (N,A)$, donde N es su conjunto de nodos y A el de arcos. Tome un nodo $h \in N$ cualquiera. Diseñe un algoritmo eficiente que entregue un ciclo dirigido que visita a h y que posea la menor cantidad de arcos. Determine su complejidad computacional.

Respuesta: Copie el nodo h con todos sus arcos entrantes. Sea h' dicha copia. Luego haga BFS desde h hacia h'. Es $O(m)$.

Parte b (I2)

Suponga un grafo dirigido (N,A) que modela una red vial. Cada nodo $i \in N$ representa una intersección y cada arco $a \in A$ es un segmento de calle entre dos intersecciones que consume un tiempo $t_a \geq 0$ de viaje. Suponga que se ha identificado un conjunto $\mathscr{L}$ de pares de arcos consecutivos $((i,j),(j,k)) \in \mathscr{L}$ tal que al recorrerse (i,j) inmediatamente antes de (j,k) se debe hace un viraje a la izquierda. Diseñe un algoritmo eficiente que genere el árbol de rutas de menor tiempo de viaje desde el nodo 1 hacia el resto de los nodos entre todas las rutas que no viran a la izquierda. Determine su complejidad computacional.

Respuesta: En este problema es necesario crear un grafo con "memoria" en cada nodo. Para ello se puede crea un nuevo conjunto de nodos $N' = N \cup \{(i,j) : i \in N, j \in N \setminus \{i,1\}, (i,j) \in A\}$. Acá, el nodo (i,j) representa estar en j habiendo visitado inmediatamente antes a i.

El conjunto de arcos sería: $A := A_1 \cup A_2 \cup A_3$, donde:

- $A_1 := \{(1,(1,j)) : (1,j) \in N'\}$ a costo $c_{(1,(j,1))} = t_{1,j}$.

- $A_2 := \{((i,j),(j,k)) : (i,j),(j,k) \in N', ((i,j),(j,k)) \notin \mathscr{L}\}$ a costo $c_{((i,j),(j,k))} = t_{j,k}$ (acá se elimina cada viraje a la izquierda).
- $A_3 := \{((i,j),j) : (i,j) \in N'\}$ a costo $c_{((i,j),j)} = 0$.

Luego, las rutas desde 1 hacia cada nodo $j \in N\{1\}$ entregan lo solicitado.

Lo anterior implica ejecutar Dijkstra sobre un grafo de $n' = O(n^2)$ nodos y $m' = O(nm)$ arcos. Aplicando la versión con heap de Fibonacci sería $O(nm + n^2 \log(n^2)) \approx O(nm)$.

Parte c

Considere un problema de flujo máximo desde un nodo s hacia otro nodo t en un grafo (N,A), donde N es el conjunto de nodos, A es el conjunto de arcos y $u_a \in A$ es la capacidad de cada arco (entera y no-negativa). Suponga que el problema de flujo máximo desde s hacia t es acotado. Llame **arco crítico hacia arriba** a cualquier arco a que al aumentar unilateralmente su capacidad de flujo u_a en una unidad genera un aumento en el valor de flujo máximo desde s hacia t.

Diseñe un algoritmo eficiente que permita identificar a **todos** los arcos críticos hacia arriba. Determine su complejidad computacional (debe ser órdenes de magnitud más rápido que resolver el problema de flujo máximo m veces).

Respuesta: Para que un arco $(i,j) \in A$ sea crítico hacia arriba deben suceder tres condiciones:

- Debe estar a capacidad en cualquier solución óptima, pues de lo contrario, la restricción no estaría activa y no tendría efecto aumentar la capacidad.
- Debe haber una ruta con capacidad de flujo mayor o igual a 1 en la red residual desde s hacia i (para que el flujo adicional pueda ir desde s hacia i)
- Debe haber una ruta con capacidad de flujo mayor o igual a 1 en la red residual desde j hacia t (para que el flujo adicional pueda ir desde j hacia t)

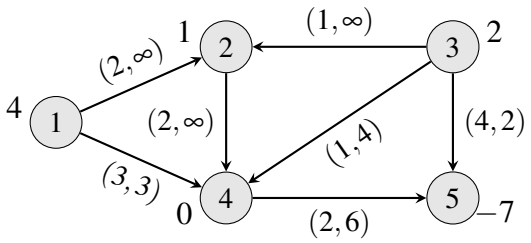
Por lo tanto el algoritmo sería el siguiente:

- Hacer Ruta aumentable desde s hacia t.
- Identificar todo arco (i,j) a capacidad en la solución. Definir el conjunto $A^c \subseteq A$ con estos arcos.
- Sobre la red residual hacer DFS (o BFS) desde s hacia adelante. Llamar N' al conjunto de nodos accesibles desde s.
- Sobre la red residual hacer DFS (o BFS) desde t hacia atrás. Llamar N'' al conjunto de que acceden a t.
- Un arco (i,j) crítico hacia arriba cumple que $(i,j) \in A^c$, $i \in N'$ y $j \in N''$.

Lo anterior ejecuta en $O(m)$ operaciones adicionales sobre calcular una vez el problema flujo máximo (Que es $O(n^3)$ con push-relabel versión FiFo o con el algoritmo Dinic versión MKM).

Pregunta 4 (17 puntos)

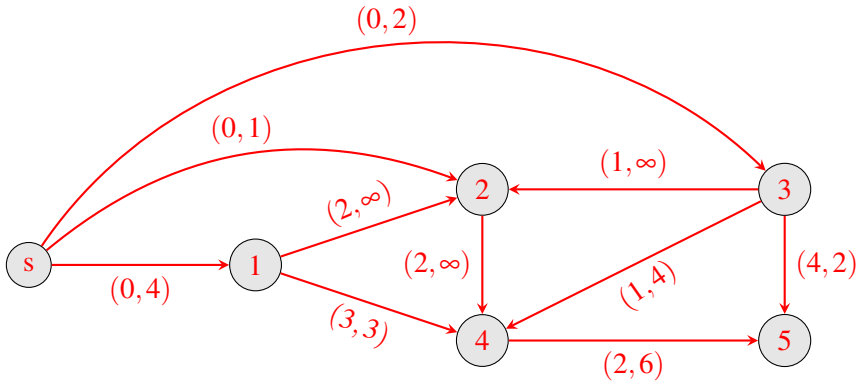
Considere la siguiente instancia de un Problema de Flujo a Mínimo Costo (PFMC) sobre un grafo dirigido $G = (N, A)$. Cada arco $a \in A$ detalla (c_a, u_a), es decir, el costo por unidad de flujo c_a y la capacidad máxima de flujo por el arco u_a, respectivamente. Las ofertas netas están detalladas al costado de cada nodo en el grafo.



(a) (2p) Convierta el problema a uno equivalente de flujo máximo a costo mínimo (PFMCM). Dibuje claramente cada elemento del PFMCM incluyendo sus nodos, arcos, costos, capacidades, fuente s y sumidero t. ¿Cuánto debe valer el flujo máximo desde s hacia t para que el PFMC original sea factible?

Respuesta:

Sería la siguiente instancia de PFMCM desde s hacia 5. Se requiere que F^* sea 7.

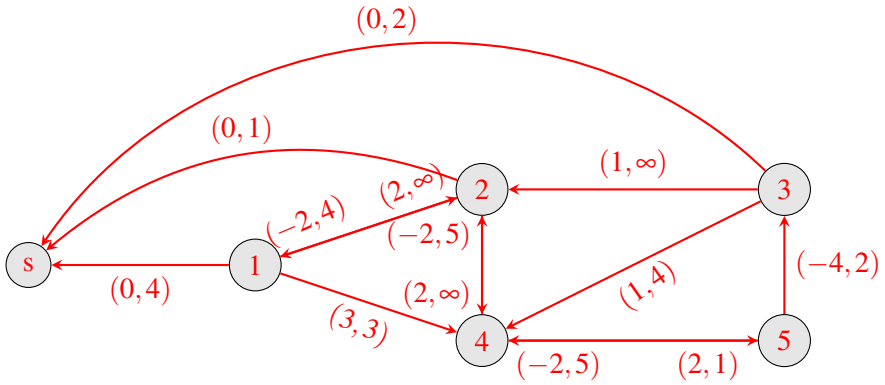


(b) (4p) Obtenga una solución de flujo máximo desde s hacia t ejecutando el algoritmo de ruta aumentable. Determine el corte saturado asociado a su solución y derive la solución factible para el PFMC original.

Respuesta: Tomemos el problema anterior y aumentemos por las rutas:

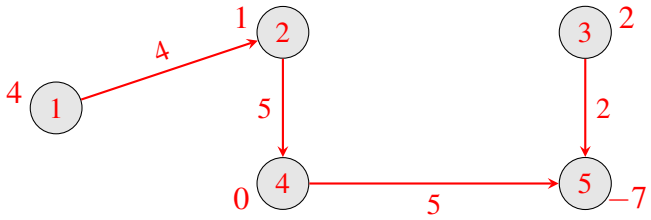
- $s - 3 - 5$ (2 unidades)
- $s - 1 - 2 - 4 - 5$ (4 unidades)
- $s - 2 - 4 - 5$ (1 unidades)

El grafo resultante queda:



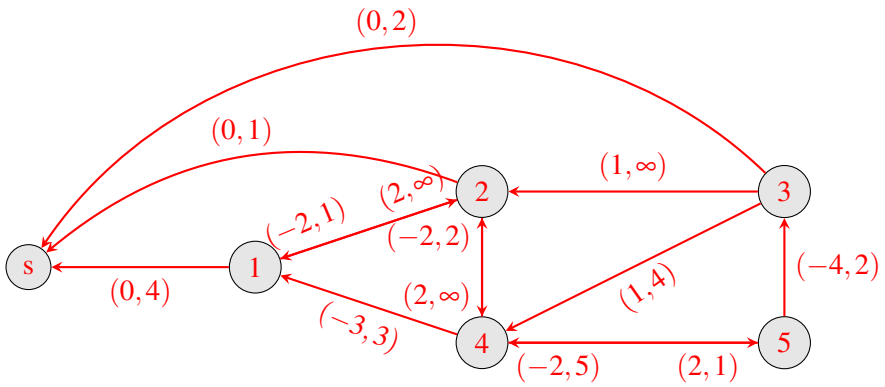
En este momento, el corte $Q = \{s\}$ está saturado.

Luego, una solución factible para el problema original es:

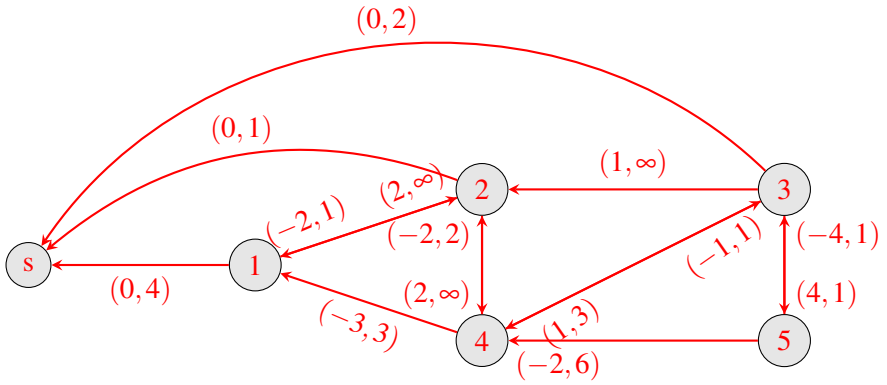


(c) (5p) Desde la solución de flujo máximo encontrada en el punto anterior: Ejecute el algoritmo de cancelación de ciclos de costo negativo y obtenga una solución óptima para el PFMCM.

Respuesta: Podemos comenzar tomando la solución de flujo máximo anterior y eliminando el ciclo negativo $1-4-2-1$ de costo -1 pasando por el un flujo de 3 unidades. El resultado es:



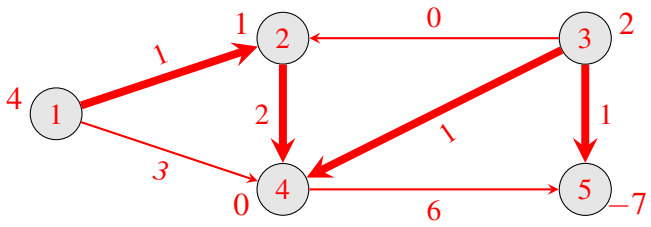
Seguimos pasando una unidad más de flujo por el ciclo $5-3-4-5$ de costo -1.



Ya no quedan ciclos negativos. La solución es óptima.

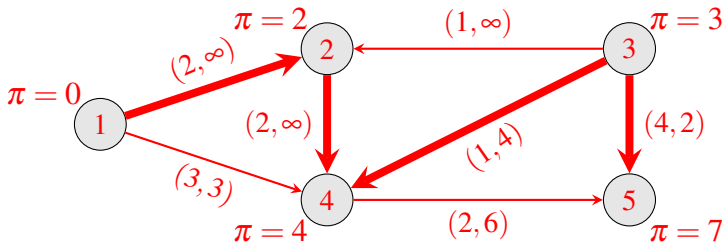
(d) (4p, I2) Derive la solución óptima para el PFMC original desde la solución encontrada para el PFMCM. Luego, verifique que es óptima mediante el criterio de costos reducidos.

Respuesta: La solución óptima encontrada para el PFMC es:



Los arcos destacados serán la base del problema (los otros estarán fuera de base en *upper* o *lower bound*.)

A continuación se detalla el cálculo de potenciales (potencial nulo en nodo 1):



Los costos reducidos fuera de base son:

- $r_{14} = 3 - (4 - 0) = -1$ (ok para variable en *upper*).
- $r_{32} = 1 - (2 - 3) = 2$ (ok para variable en *lower*).
- $r_{54} = 2 - (7 - 4) = -1$ (ok para variable en *upper*).

(e) (2p, I2) Determine el rango de valores para el costo del arco $(3,2)$ (ahora igual a 1) para que se mantenga la solución óptima encontrada.

Respuesta: Se requiere que $r_{32} = c_{32} - (2 - 3) = c_{32} + 1 \geq 0$. Es decir $c_{32} \geq -1$.