IN3701 - Modelamiento y Optimización

Auxiliar Nº 3 - Algoritmo de Ford-Fulkerson

Profesores: Victor Bucarey, Pablo Rey.
Auxiliares: Cristiam Gil, Dana Pizarro, Diego Bernstein, Diego Fuentealba, Eduardo Lara, Macarena Osorio, Mario Morales, Tomás Lagos.

P1.

Las placas madre de los computadores son manufacturados (en unidades de mil) por tres compañías c_1, c_2 y c_3. Luego, son distribuidos a dos fábricas de computadores, m_1 y m_2, a través de la red de transporte señalada a continuación.

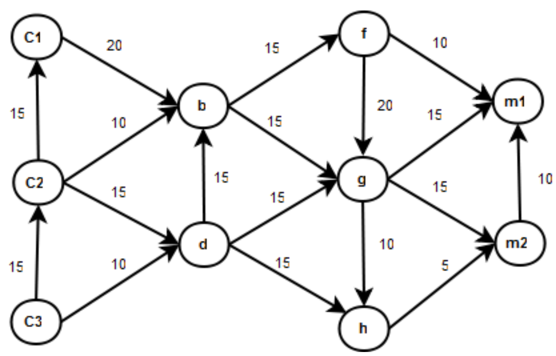
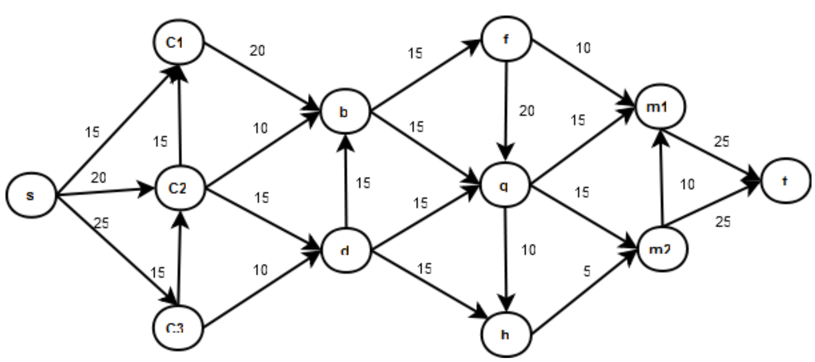


Fig. 1: Red para el problema 1.

La compañía c_1 puede producir hasta 15 unidades, la compañía c_2 hasta 20 unidades y la compañía c_3 hasta 25 unidades. Si cada fábrica necesita 25 unidades, ¿Cuántas unidades debe producir cada compañía de manera tal que en conjunto satisfagan la demanda de cada fábrica o al menos suministrarles la mayor cantidad posible que permita la red?

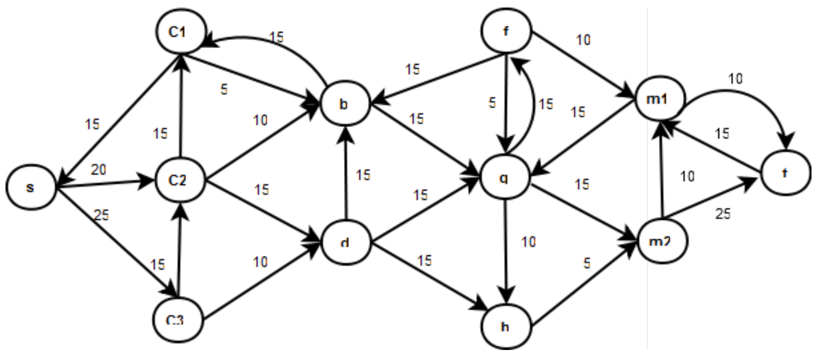
Pauta Pregunta 1:

Para modelar este ejercicio, agregamos el nodo origen s y el nodo destino t. La capacidad de producción de cada compañía serán las capacidades de los arcos (s, c_1), (s, c_2) y (s, c_3). Para los arcos (m_1, t) ,(m_2, t) las demandas son utilizadas como capacidad.



Debemos entonces aplicar el algoritmo de Ford-Fulkerson a dicha red.

En la primera iteración tomamos el camino aumentante $s \to c_1 \to b \to f \to g \to m_1 \to t$, por dicho camino pasan 15 unidades de flujo y la red residual resultante es la siguiente:



En la segunda iteración tomamos el camino $s \to c_2 \to d \to b \to g \to m_2 \to t$, por dicho camino pasan 15 unidades de flujo (por lo que el flujo total sería 30 unidades).

En la tercera iteración tomamos el camino $s \to c_2 \to b \to d \to h \to m_2 \to t$, por dicho camino pasan 5 unidades de flujo (por lo que el flujo total sería 35 unidades).

En la cuarta iteración tomamos el camino $s \to c_3 \to d \to g \to m_2 \to t$, por dicho camino pasan 10 unidades de flujo (por lo que el flujo total sería 45 unidades)

En la próxima iteración observamos que no hay un camino aumentante, con lo cual hemos llegado al flujo óptimo y es igual a 45 unidades.

P2.

Aplique el algoritmo de Ford-Fulkerson al siguiente Grafo:

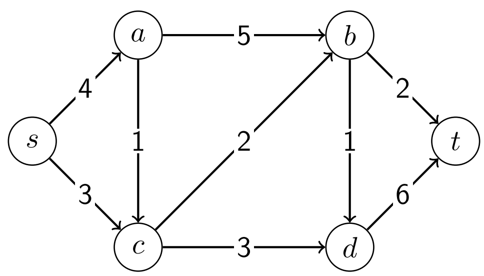
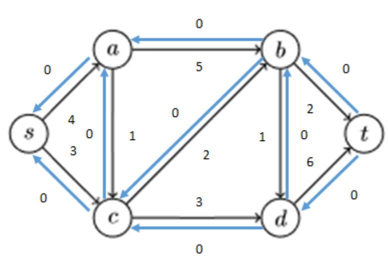


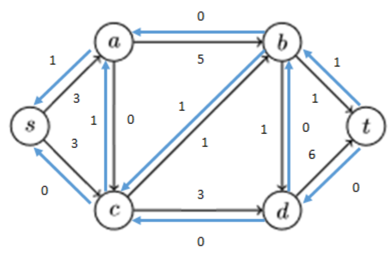
Fig. 2: Red para el problema 2.

Pauta Pregunta 2:

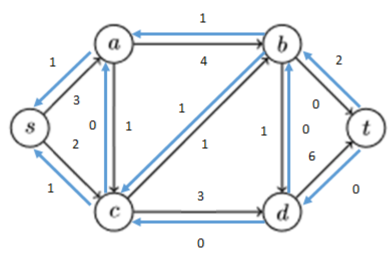
Creamos un grafo alternativo: G'(M,O):



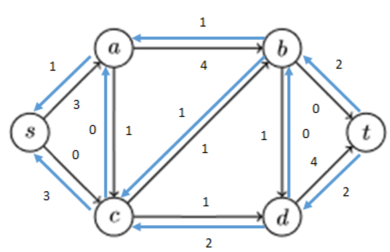
En la primera iteración se elige un camino factible al azar, en este caso elegiremos C1=(s,a),(a,c),(c,b),(b,t). Luego observamos mín(4,1,2,2)=1



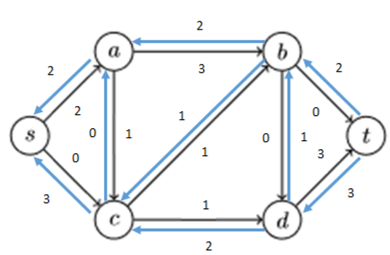
En la segunda iteración se elige el camino C2=(s,c),(c,a),(a,b),(b,t). Luego observamos mín(3,1,5,1)=1.



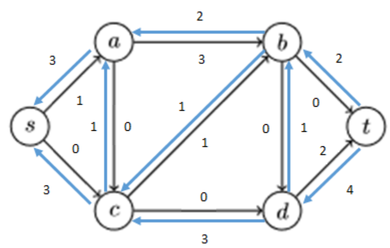
En la tercera iteración se elige el camino C3=(s,c),(c,d),(d,t). Luego observamos mín(2,3,6)=2.



En la cuarta iteración se elige el camino C4= (s,a),(a,b),(b,d),(d,t). Luego observamos mín(2,4,1,6)=1.



En la quinta iteración se elige el camino C5= (s,a),(a,c),(c,d),(d,t). Luego observamos mín(2,1,1,3)=1.



Finalmente sumamos todos los mínimos 1+1+2+1+1 =6. Por lo tanto, el flujo máximo es 6.

Nota: Recordar que no es el único camino a seguir, pues las combinaciones de caminos NO son únicas, pero el resultado del flujo máximo sí lo es.v

P3.

Un problema fundamental en el diseño urbano es la localización de servicios básicos como colegios, hospitales y áreas recreacionales. En este problema formularemos un modelo simplificado para decidir la localización de estaciones de bomberos en una ciudad.

La ciudad se puede dividir en I distritos, en que cada uno contiene p_i habitantes. Análisis preliminares (estudios de terrenos, factores políticos, etc.) han establecido que las estaciones de bomberos solo pueden ser ubicadas en J sitios predeterminados dentro de la ciudad. Sea $d_{ij} > 0$ la distancia desde el centro del distrito i hasta el sitio j. Se deben seleccionar los sitios en los cuales construir una estación (en un sitio cabe a lo más una) y además se debe asignar una estación a cada distrito. Es decir, cada distrito de la ciudad debe tener una (y solo una) estación de bomberos asociada. Una estación puede tener más de un distrito asociado. Construir una estación en el sitio j tiene un costo fijo asociado igual a c_j. Además, existe un costo variable que es linealmente proporcional (constante de proporcionalidad es f_j) a la cantidad total de gente que debe servir la estación. O sea, si se construye una estación en el sitio j, entonces el costo asociado es $c_j + f_j s_j$, en que s_j es la población total que debe servir la estación ubicada en j (es la suma de las poblaciones de todos los distritos asociados a esa estación). El presupuesto total destinado para construir las estaciones de bomberos es igual a B y no debe ser sobrepasado.

Formule un modelo de programación lineal que minimice la distancia máxima entre un distrito y su respectiva estación.

Pauta Pregunta 3:

Variables de Decisión:

- $X_j = \begin{cases} 1, & \text{si se construye una estación en el sitio j, } j \in J \\ 0, & \text{si no se construye estación en el sitio j} \end{cases}$
- $Y_{ij} = \begin{cases} 1, & \text{si se asocia el distrito i con el local ubicado en j, } j \in J,\, i \in I \\ 0, & \text{si no se asocia el distrito i con el local ubicado en j} \end{cases}$
- $s_j =$ Cantidad de gente asociada a la estación j, $j \in J$
- $z =$ Distancia máxima entre un distrito y su estación asociada.

Restricciones:

1. Naturaleza de las variables:

$$X_j, Y_{ij} \in \{0,1\},\ s_j, z \in \mathbb{R}^+$$

2. Si se asigna una estación, ésta se debe construir:

$$X_j \geq Y_{ij}, \forall\ j \in J, \forall i \in I$$

3. Cada distrito debe tener una estación asignada:

$$\sum_{j \in J} Y_{ij} = 1, \forall i \in I$$

4. No se puede sobrepasar el presupuesto:

$$B \geq \sum_{j \in J} X_j \cdot c_j + f_j s_j$$

5. Gente asociada a cada distrito:

$$s_j \geq \sum_{i \in I} Y_{ij} \cdot p_i, \qquad \forall j \in J$$

6. Relación distancia máxima:

$$z \geq Y_{ij} \cdot d_{ij}, \qquad \forall j \in J, \forall i \in I$$

Función Objetivo:

$$\text{mín } z$$

Resumen:

1. Un **Grafo simple** $G(V, A)$ se define por vertices y arcos (caso dirigido, o arístas en caso no-dirigido), tal que todo arco conecta dos vertices distintos, y no existen dos arcos que conectan los mismos dos vertices (en la misma dirección).

2. Un **camino** desde v hasta u es una forma de conectar a v con u utilizando los arcos del grafo en su dirección factible de manera de que ningún nodo es visitado dos veces.

3. Un grafo se dice que **está conectado** si es que para cada par de nodos existe un camino que los conectan.

4. El problema de **encontrar el flujo máximo** entre un par de nodos en un grafo se puede resolver mediante la siguiente formulación:

$$\begin{array}{cc} \max_{\{l_a \leq x_a \leq u_a : a \in E \setminus \{ts\}\}} & x_{ts} \\ div_i(x) = 0 & i \in V \end{array}$$

Donde $div_i(x) = \sum_{\{a:a=(i,j)\in E\}} x_a - \sum_{\{a:a=(j,i)\in E\}} x_a$ es el operador de divergencia en el nodo i sobre los arcos del grafo que se conectan con tal nodo.

5. El **algoritmo de Ford-Fulkerson** resuelve el problema anterior:

Algorithm 1 PROCEDURE W:

Require: $m(i) = null\ \forall i$;
Require: $m(s) = (r, +\infty)$; % r es el arco de retorno
Ensure: $L = [s]$;

while $t \notin L$ AND $\exists i \in L$ **do**
    Eliminar i de L;
    for $a = (i, j) \in A$ **do**
        if $m(j) == null$ AND $x_a < c_a$ **then**
            $m(j) = (a, \min\{c_a - x_a, \delta_i\})$;
            $L = L \cup [j]$;
    for $a = (j, i) \in A$ **do**
        if $m(j) == null$ AND $x_a > 0$ **then**
            $m(j) = (a, \min\{x_a, \delta_i\})$;
            $L = L \cup [j]$;

Algorithm 2 FORD-FULKERSON:

Require: Optimo=$False$;

while !Optimo **do**
    Aplicar Procedure W;
    if t no fue marcado **then**
        Optimo $= True$;
        Print x;
    else
        Aumentar el flujo en una cantidad δ_t;