

Profesores: Guillermo Duran, Daniel Espinoza G.

Semestre: Otoño 2007

Fecha: 6 de junio de 2007

IN34A Optimización

Control N°3

Problema 1 (33.3 %)

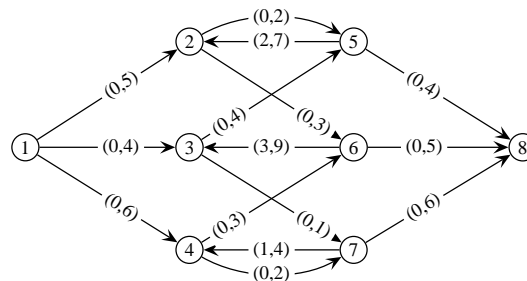


Figura 1: Red 1

1. (3ptos.) Resuelva el problema de flujo máximo del nodo 1 al nodo 8 del grafo en la figura 1. Use el algoritmo de Ford Fulkerson para resolverlo; la etapa de buscar un flujo factible inicial puede resolverlo de cualquier manera (inspección o fase I de Ford Fulkerson). Note que cada arco en la figura tiene asociado dos números, el primero es el flujo mínimo requerido en el arco, y el segundo la capacidad máxima de flujo.
2. (3ptos.) Considere que se agrega el arco (8,1) en el grafo descrito en la figura 1 con capacidades (4,5), y asuma que el costo de un arco está dado por $c_e = \frac{u_e}{1 + l_e}$, donde u_e es la capacidad superior del arco, y l_e el flujo mínimo requerido. Compute las distancias mas cortas (según la función de costo dada) del nodo 6 a todos los nodos del grafo, para esto utilice el algoritmo de Dijkstra.

Problema 2 (33.3 %)

Don King, extrañando mucho su ciudad natal, acaba de abandonar su posición académica en el Departamento de Ingeniería Industrial para asumir como intendente de la ciudad trasandina de Górdoba y quiere aprovechar todos sus conocimientos en programación matemática para congraciarse con los estudiantes de la ciudad (y con sus padres). Para ello ha decidido relocalizar todos los colegios de modo de hacer más cómoda la movilidad de los alumnos.

La ciudad se puede dividir en I distritos, en que cada uno contiene p_i alumnos. Análisis preliminares (estudios de terrenos, factores políticos, etc.) han establecido que las escuelas sólo pueden ser ubicadas en J sitios predeterminados dentro de la ciudad.

Sea $d_{ij} \geq 0$ la distancia desde el centro del distrito i hasta el sitio j . Se deben seleccionar los sitios en los cuales construir un colegio (en un sitio cabe a lo más uno) y además se debe asignar un colegio a cada distrito. Es decir, cada distrito de la ciudad debe tener uno (y sólo un) colegio asociado. En cambio, cada colegio puede tener hasta 2 distritos asociados.

Construir un colegio en el sitio j tiene un costo fijo asociado igual a c_j . Existe también un costo variable que es linealmente proporcional (la constante de proporcionalidad es f) a la cantidad total de alumnos que debe servir el colegio. O sea, si se construye un colegio en el sitio j , entonces el costo asociado es $c_j + fs_j$, en que s_j es la población total que debe servir el colegio ubicado en j (es la suma de las poblaciones de todos los distritos asociados a ese colegio).

La capacidad de alumnos que soporta un colegio construido en el sitio j es un dato conocido (T_j).

El presupuesto total destinado para construir los colegios es igual a B y no debe ser sobrepasado.

Además, La Dirección de Educación de la ciudad ha determinado que los distritos s y t deben ser atendidos por 2 colegios distintos.

Ayude a Don King a aumentar su popularidad formulando un modelo de programación lineal mixto que, respetando las condiciones planteadas, determine dónde construir los colegios y qué colegio atiende a qué distrito. El objetivo es minimizar la distancia máxima entre el centro de un distrito y su respectivo colegio.

Problema 3 (33.3 %)

1. Suponga que al resolver la relajación lineal de un problema de programación entera le da como óptimo un punto de coordenadas enteras. ¿Qué puede afirmar entonces? ¿Por qué? (1,5 pts.)
2. Suponga que al resolver un problema de programación entera, en la primera ramificación del algoritmo de B&B en una de las ramas le da un óptimo con coordenadas enteras. ¿Puede afirmar que es el óptimo del problema general? ¿Por qué? (1,5 pts.)
3. Suponga que tiene que resolver un problema de programación entera con poliedro factible de la relajación lineal acotado. Analice la veracidad de la siguiente frase: dado que el problema de programación entera puede resolverse mediante B&B resolviendo un número finito de problemas lineales, si uso un algoritmo polinomial para resolver dichos problemas lineales entonces tengo un algoritmo polinomial para el problema original. (1,5 pts.)
4. Muestre con un ejemplo por qué el algoritmo de Dijkstra para rutas más cortas puede no servir si hay arcos con costo negativo. Justifique. (1,5 pts.)

Pauta

Problema 2

1. Variables de Decisión: (1 pto en total)

- x_j : 1 si se construye un colegio en el sitio j . 0 en cualquier otro caso
 y_{ij} : 1 si el distrito i es asignado al colegio ubicado en el sitio j . 0 en cualquier otro caso
 z : Máxima distancia entre un distrito y un colegio

2. Restricciones:

- a) Cada distrito debe ser asignado a exactamente un colegio (0,5 ptos).

$$\sum_j y_{ij} = 1 \quad \forall i$$

- b) Cada colegio no puede ser asignado a más de dos distritos (0,5 ptos).

$$\sum_i y_{ij} \leq 2 \quad \forall j$$

- c) Solo se puede asignar distrito i a colegio en sitio j si está contruido (0,5 ptos).

$$y_{ij} \leq x_j \quad \forall i, j$$

- d) Respetar presupuesto (0,5 ptos).

$$\sum_j (x_j c_j + f \sum_i y_{ij} p_i) \leq B$$

- e) Capacidad máxima de alumnos por colegio (0,5 ptos).

$$\sum_i y_{ij} p_i \leq T_j \quad \forall j$$

- f) Distritos s y t en distintos colegios (0,5 ptos).

$$y_{sj} + y_{tj} \leq 1 \quad \forall j$$

- g) Establecer la máxima distancia recorrida (0,5 ptos).

$$z \geq y_{ij} d_{ij} \quad \forall i, j$$

h) Naturaleza de las variables (0,5 pts).

$$x_i, y_{ij} \in \{0, 1\} \quad \forall i, j$$

$$z \geq 0$$

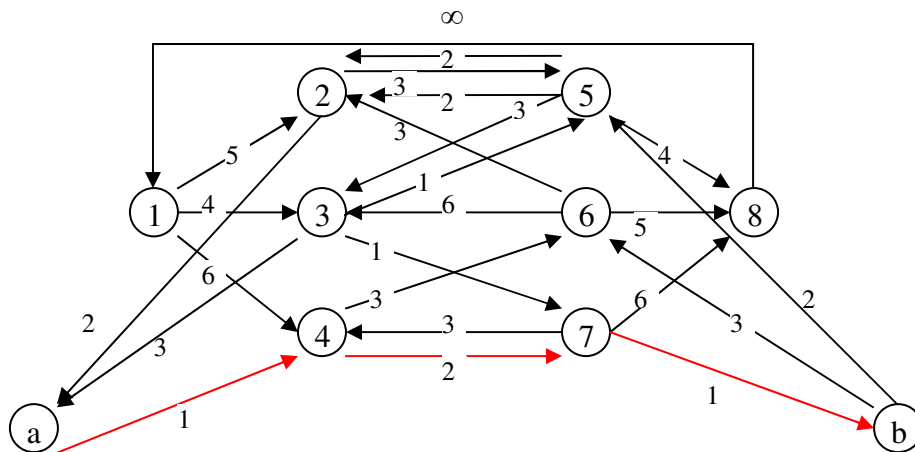
3. Función Objetivo: (1 pto)

Minimizar la maxima distancia entre un colegio y su distrito asociado.

$$\text{mín } z$$

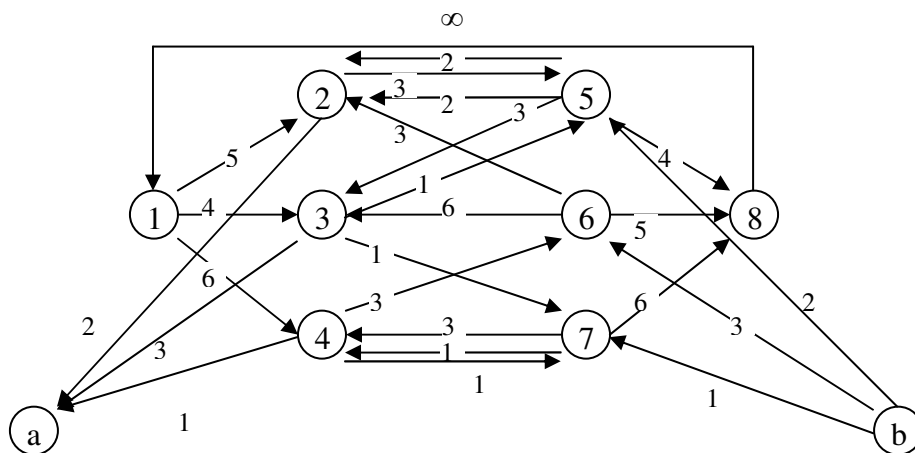
Nota de Corrección: De modelar el problema de una forma distinta (por lo general más extensa por poner más variables) redistribuir el puntaje de manera razonable.

Iteración 3:



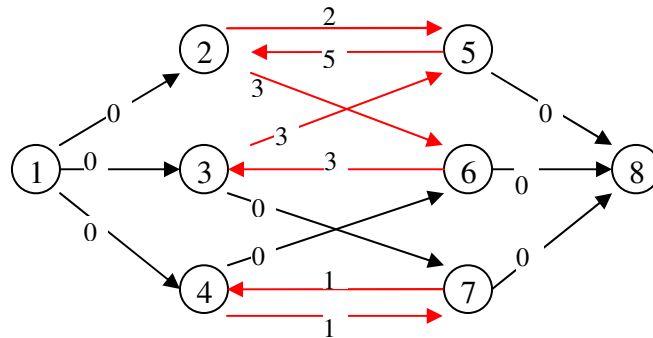
Se elige a-4-7-b con $\epsilon = 1$

Iteración 4:



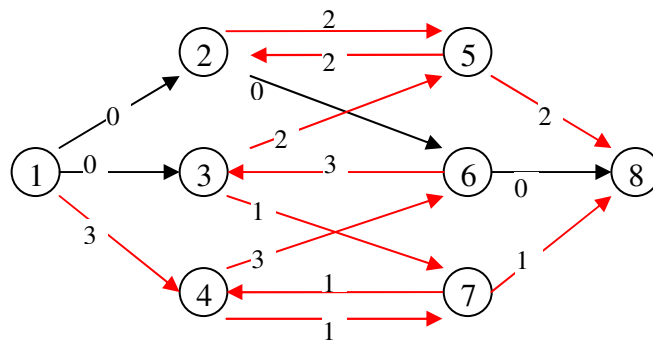
No existe ruta entre a y b \rightarrow estamos en el óptimo con $F^* = 6$ (que es igual a la suma de las capacidades inferiores de los arcos y por lo tanto es factible).

El flujo en el grafo original queda:



Observaciones:

- 1.- Esta es sólo una alternativa, existen muchos otros flujos factibles que cumplen las condiciones del problema. Dependiendo de el orden de las rutas elegidas puede cambiar la cantidad de iteraciones.
- 2.- No es necesario que en cada iteración dibujen el grafo nuevamente, lo importante es que se entienda lo que hicieron y sean consistentes.
- 3.- No es necesario hacer Fase I, se podía encontrar un flujo por inspección, por ejemplo:

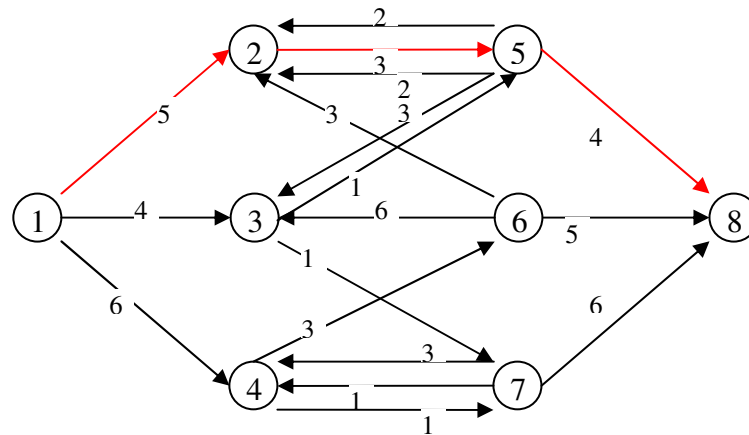


Resolución del problema original (1,5):

Partiendo del flujo encontrado con Fase 1:

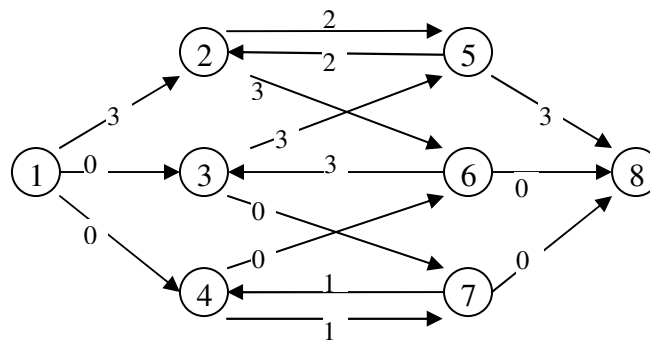
Iteración 1:

G':



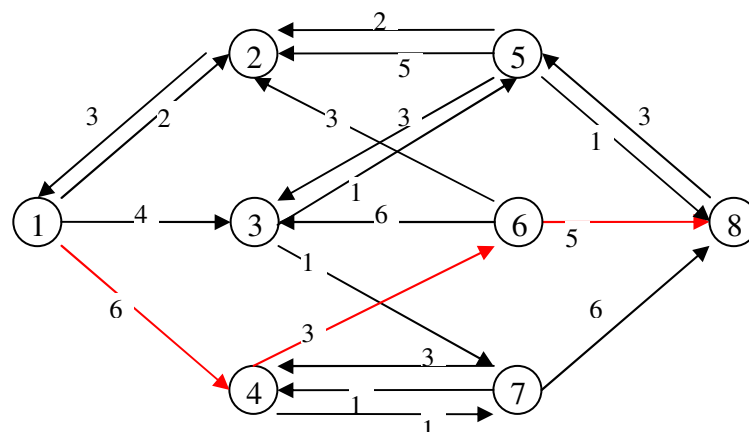
Se elige 1-2-5-8 con $\epsilon = 3$

G:



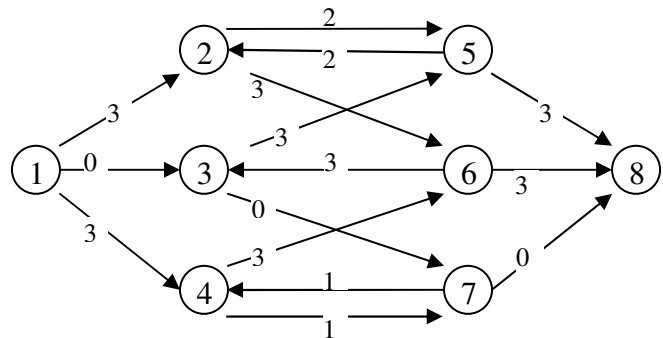
Iteración 2:

G':



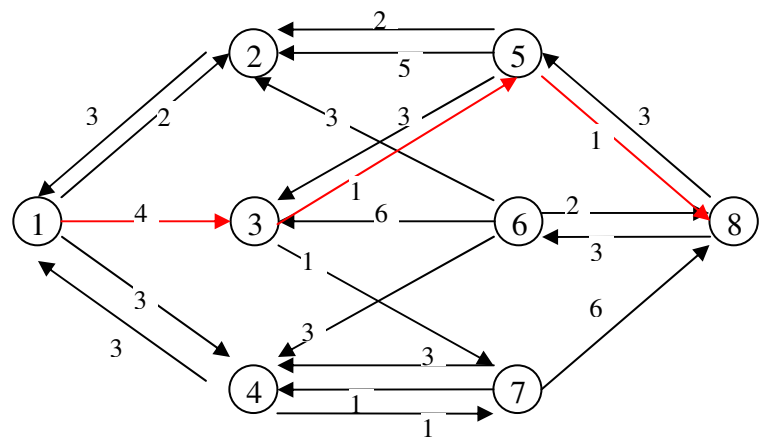
Se elige 1-4-6-8 con $\epsilon = 3$

G:



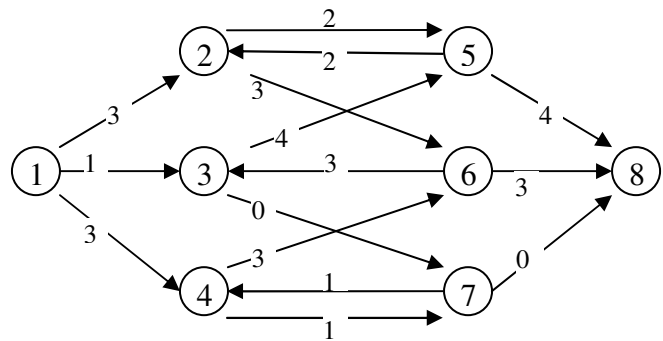
Iteración 3:

G':



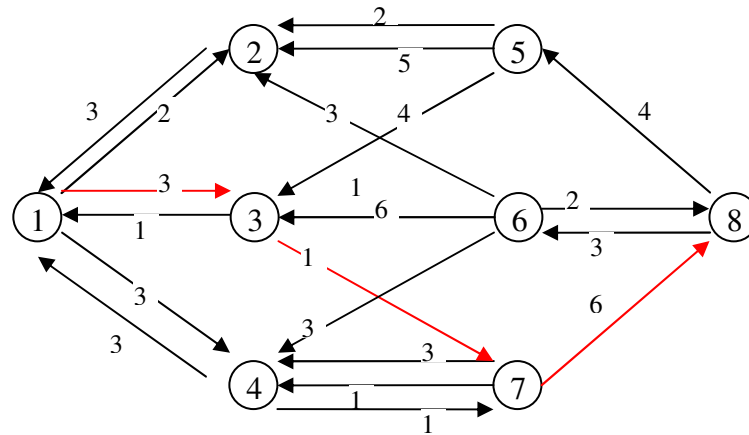
Se elige 1-3-5-8 con $\epsilon = 1$

G:



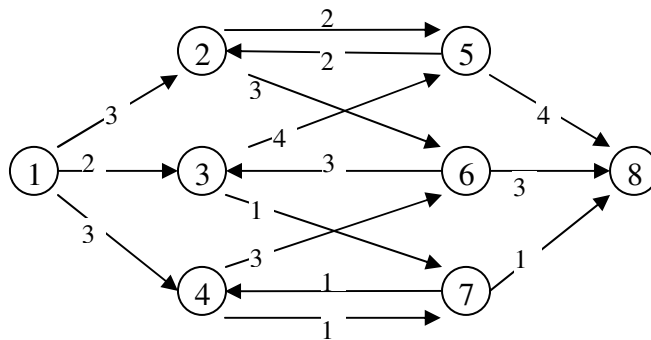
Iteración 4:

G':



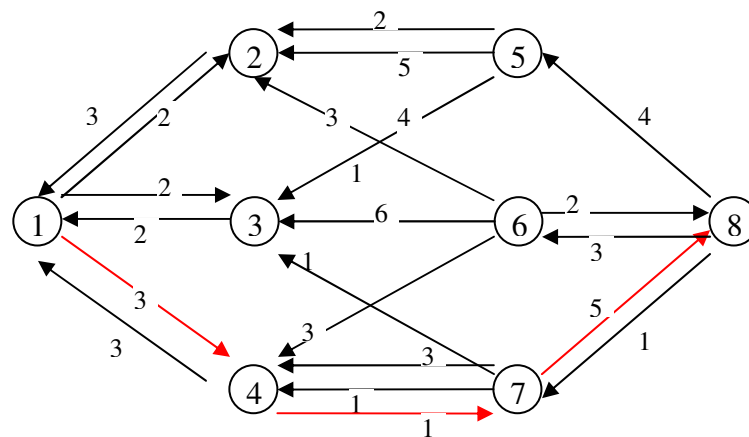
Se elige 1-3-7-8 con $\varepsilon = 1$

G:



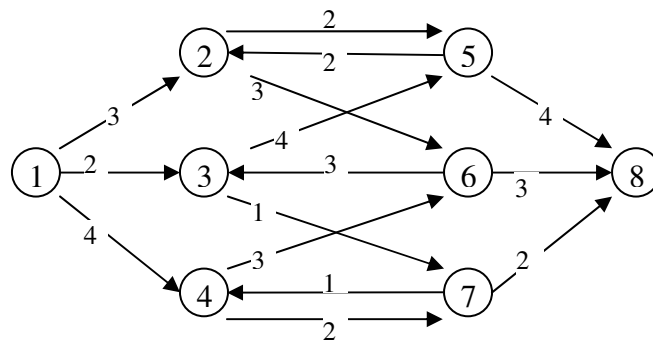
Iteración 5:

G':



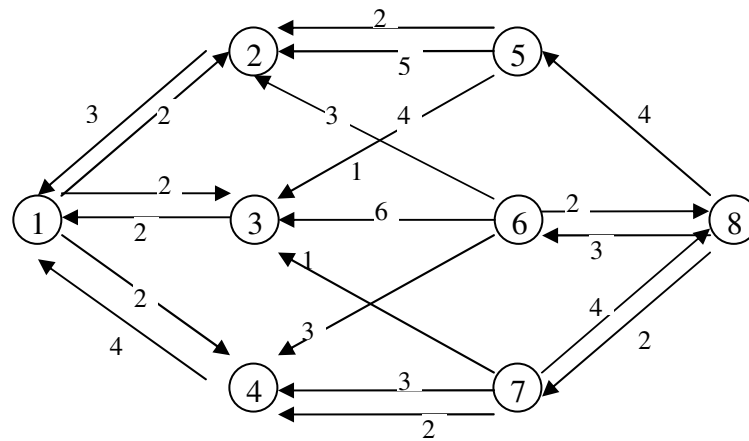
Se elige 1-4-7-8 con $\varepsilon = 1$

G:



Iteración 6:

G':

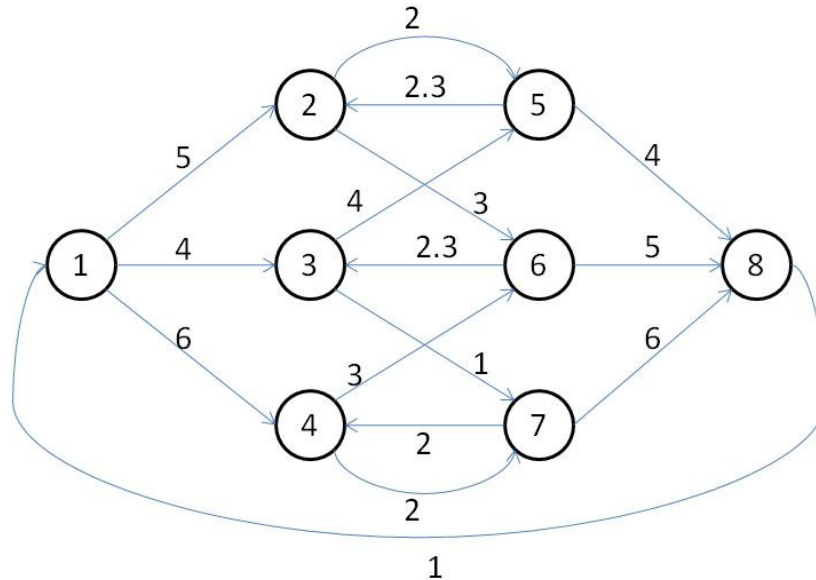


No existe ruta entre 1 y 8 → estamos en el óptimo, con flujo máximo 9.

2.

3 puntos (3,75 décimas cada iteración)

La red y sus costos asociados quedan:



Iteracion 0:

$S=\emptyset$; $T=V$

Solo se conoce la distancia al nodo 6, el resto son desconocidos.

$\Pi_6=0$; $\Pi_i=\infty$ $i \neq 6$

$\Rightarrow \min_{i \in T} \{ \Pi_i \} = 0$; $S=S \cup \{6\}$; $T=T/\{6\}$

Iteracion 1:

$S=\{6\}$; $T=V/\{6\}$

$\Pi_3 = \min\{\infty, 0+2.3\} = 2.3$ $L_3=6$;

$\Pi_8 = \min\{\infty, 0+5\} = 5$ $L_8=6$;

$\Pi_1 = \Pi_2 = \Pi_4 = \Pi_5 = \Pi_7 = \infty$

$\min_{i \in T} \{ \Pi_i \} = 2.3$; $\Rightarrow S=S \cup \{3\}$; $T=T/\{3\}$

Iteracion 2:

$S=\{3,6\}$; $T=\{1,2,4,5,7,8\}$

$\Pi_5 = \min\{\infty, 2.3+4\} = 6.3$ $L_5=3$;

$\Pi_7 = \min\{\infty, 2.3+1\} = 3.3$ $L_7=3$;

$\Pi_8=5$ $L_8=6$;

$\Pi_1 = \Pi_2 = \Pi_4 = \infty$

$\min_{i \in T} \{ \Pi_i \} = 3.3$; $\Rightarrow S=S \cup \{7\}$; $T=T/\{7\}$

Iteracion 3:

$$S=\{3,6,7\}; \quad T=\{1,2,4,5,8\}$$

$$\Pi_4=\min\{\infty, 3.3+2\}=5.3 \quad L_4=7;$$

$$\Pi_5=6.3 \quad L_5=3;$$

$$\Pi_8=\min\{5, 3.3+6\}=5 \quad L_8=6;$$

$$\Pi_1 = \Pi_2 = \infty$$

$$\text{Min}_{i \in T} \{ \Pi_i \} = 5; \quad \Rightarrow \quad S = S \cup \{8\}; \quad T = T / \{8\}$$

Iteracion 4:

$$S=\{3,6,7,8\}; \quad T=\{1,2,4,5\}$$

$$\Pi_1=\min\{\infty, 5+1\}=6 \quad L_1=8;$$

$$\Pi_4 = 5.3 \quad L_4=7;$$

$$\Pi_5=6.3 \quad L_5=3;$$

$$\Pi_2 = \infty$$

$$\text{Min}_{i \in T} \{ \Pi_i \} = 5.3; \quad \Rightarrow \quad S = S \cup \{4\}; \quad T = T / \{4\}$$

Iteracion 5:

$$S=\{3,4,6,7,8\}; \quad T=\{1,2,5\}$$

$$\Pi_1=6 \quad L_1=8;$$

$$\Pi_5=6.3 \quad L_5=3;$$

$$\Pi_2 = \infty$$

$$\text{Min}_{i \in T} \{ \Pi_i \} = 6; \quad \Rightarrow \quad S = S \cup \{1\}; \quad T = T / \{1\}$$

Iteracion 6:

$$S=\{1,3,4,6,7,8\}; \quad T=\{2,5\}$$

$$\Pi_2=\min\{\infty, 6+5\}=11 \quad L_2=1;$$

$$\Pi_5=6.3 \quad L_5=3;$$

$$\text{Min}_{i \in T} \{ \Pi_i \} = 6.3; \quad \Rightarrow \quad S = S \cup \{5\}; \quad T = T / \{5\}$$

Iteracion 7:

$$S=\{1,3,4,5,6,7,8\}; \quad T=\{2\}$$

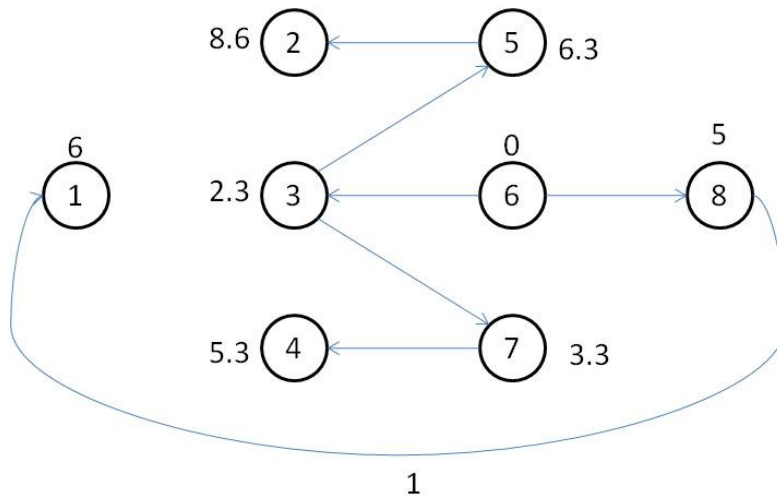
$$\Pi_2=\min\{11, 6.3+2.3\}=8.6 \quad L_2=5;$$

$$\text{Min}_{i \in T} \{ \Pi_i \} = 8.6; \quad \Rightarrow \quad S = S \cup \{2\}; \quad T = T / \{2\}$$

$\Rightarrow S=\{1,2,3,4,5,6,7,8\}$ $T=\emptyset$

Se conoce la ruta mas corta desde el nodo 6 a cada nodo, por lo que el algoritmo termina.

La ruta mas corta a cada nodo, desde el nodo 6, queda:



Problema 2

1. Variables de Decisión: (1 pto. en total)

x_j : 1 si se construye un colegio en el sitio j . 0 en cualquier otro caso

$y_{i,j}$: 1 si el distrito i es asignado al colegio ubicado en el sitio j . 0 en cualquier otro caso

z : Máxima distancia entre un distrito y un colegio

2. Restricciones:

a) Cada distrito debe ser asignado a exactamente un colegio (0,5 ptos).

$$\sum_j y_{ij} = 1 \quad \forall i$$

b) Cada colegio no puede ser asignado a más de dos distritos (0,5 ptos).

$$\sum_i y_{ij} \leq 2 \quad \forall j$$

c) Sólo se puede asignar distrito i a colegio en sitio j si está construido (0,5 ptos).

$$y_{ij} \leq x_j \quad \forall i, j$$

d) Respetar presupuesto (0,5 ptos).

$$\sum_j (x_j c_j + f \sum_i y_{ij} p_i) \leq B$$

e) Capacidad máxima de alumnos por colegio (0,5 ptos).

$$\sum_i y_{ij} p_i \leq T_j \quad \forall j$$

f) Distritos s y t en distintos colegios (0,5 ptos).

$$y_{sj} + y_{tj} \leq 1 \quad \forall j$$

g) Establecer la máxima distancia recorrida (0,5 ptos).

$$z \geq y_{ij} d_{ij} \quad \forall i, j$$

h) Naturaleza de las variables (0,5 ptos).

$$\begin{aligned}x_i, y_{ij} &\in \{0, 1\} & \forall i, j \\z &\geq 0\end{aligned}$$

3. Función Objetivo: (1 pto)

Minimizar la máxima distancia entre un colegio y su distrito asociado.

Mín z

Nota de Corrección: De modelar el problema de una forma distinta (por lo general más extensa por poner más variables) redistribuir el puntaje de manera razonable.

Problema 3

1,5 ptos c/u

1. Se puede afirmar que el óptimo de la relajación lineal es el óptimo del problema de programación entera. Esto se debe a que la solución de la relajación lineal, a pesar de que no se le exige, cumple también con las restricciones de integralidad que fueron relajadas del problema de programación entera.
2. No se puede afirmar que sea el óptimo del problema general pues se pueden presentar varias situaciones:
 - a. Si en la otra rama también da una solución entera, el óptimo del problema general es el de la rama con mejor valor en la función objetivo.
 - b. Si en la otra rama la solución no es entera y el valor de la función objetivo es peor que el de la rama con solución entera, entonces la rama con solución entera es el óptimo pues ramificar la otra rama sólo empeorará el valor de la función objetivo.
 - c. Si en la otra rama la solución no es entera pero el valor de la función objetivo es mejor que el de la rama con solución entera, entonces se debe ramificar la rama y no se puede afirmar que sea el óptimo del problema general.
3. Es cierto que un problema de programación entera se puede resolver con un número finito de problemas lineales, pero no es necesariamente cierto que el número de problemas lineales que se tenga que resolver sea polinomial en función del tamaño de la instancia del problema original. Es decir, es posible que para alguna instancia específica sea necesario examinar una cantidad exponencial de ramas y subproblemas en función del tamaño de la instancia. La afirmación es falsa pues de lo contrario programación entera sería polinomial.
4. Si hay arcos con costo negativo hay casos en que el algoritmo de Dijkstra no encuentra la ruta más corta. Por ejemplo para el siguiente grafo, si se busca la ruta más corta del nodo 1 al nodo 3, Dijkstra encuentra que la ruta más corta es ir directamente del nodo 1 al 3 con costo 2. Sin embargo, existe una ruta de costo menor que es ir del nodo 1 al 2 y luego del nodo 2 al 3, con costo 1.

