



Tópicos Avanzados en Optimización Combinatoria y Teoría de Grafos

Docentes: Graciela Nasini, Daniel Severin, Paola Tolomei, Pablo Torres

PRÁCTICA N° 7: MODELOS Y RESOLUCIÓN DE PROBLEMAS MEDIANTE PROGRAMACIÓN LINEAL ENTERA.

1. Sea $G = (V, E)$ un grafo conexo y $c \in \mathbb{R}_+^E$. Sea

$$(PE) \quad z = \min\{cx : \sum_{u \in S; v \notin S} x_{uv} \geq 1 \quad \forall S \subset V; x \in \{0, 1\}^E\}.$$

- Pruebe que si x es una solución factible de (PE), el subgrafo de G inducido por los arcos $\{uv \in E : x_{uv} = 1\}$ es un grafo conexo.
- Pruebe que (PE) modela el Problema del Árbol Generador Mínimo.

2. Se sabe que la relajación clique de un grafo G es un poliedro entero si y sólo si G es un grafo perfecto. Por otro lado, si G es un grafo perfecto, su complemento también lo es.

a) Probar que la relajación clique del complemento de G es

$$(PL) \quad \max\{\sum_{v \in V} x_v : \sum_{v \in S} x_v \leq 1 \quad \forall S \text{ estable de } G\}.$$

- Sea (D) el problema dual de (PL). Probar que (D) tiene una solución óptima con todas sus componentes menores o iguales a 1.
- Pruebe que si (D) tiene solución óptima entera, el valor de la función objetivo coincide con el número cromático de G .
- Pruebe que si G es perfecto, (D) tiene solución óptima entera.

3. Considere el problema de asignar viajes a conductores minimizando la cantidad de horas extras (ver [SLIDES1]). Recuerde que una instancia consiste en los siguientes elementos:

- $K = \{\text{conjunto de conductores}\}$; $V = \{\text{conjunto de viajes}\}$;
- $A = \{\text{pares ordenados de viajes que se pueden realizar consecutivamente}\}$
- $J = \{\text{pares ordenados primer-último viaje de una jornada factible}\}$
- $c_{uv} = \text{minutos extras de una jornada que comienza en } u \text{ y finaliza en } v, \forall (u, v) \in J$

Llamamos Modelo 1 del problema al que utiliza variables de asignación conductor-viaje (ver [SLIDES1]).

Sea $J^* = \{(b, a) : (a, b) \in J\}$. Dado el digrafo $D^* = (V, A^*)$ con $A^* = A \cup J^*$, cada jornada laboral puede representarse como un circuito dirigido en D^* y el problema de asignación de conductores a viajes puede ser pensado como asignar a cada conductor un

circuito dirigido en D^* con un único arco de J^* , de manera que los conjuntos de vértices que participan en cada circuito formen una partición de V .

Llamamos Modelo 2 a la formulación del problema como programa lineal entero, basado en las variables binarias:

$$z_{kuv} = \begin{cases} 1, & \text{si conductor } k \text{ toma viaje } u \text{ y luego } v \\ 0, & \text{si no} \end{cases}$$

para todo $k \in K$, $(u, v) \in A^*$ (ver [SLIDES 2]).

Compare la performance de los modelos 1 y 2, determinando, para cada modelo el tamaño máximo que el solver SCIP es capaz de resolver en a lo sumo una hora de CPU.

Nota: Trabaje sobre las instancias dadas en la página de la cátedra y realice una tabla comparativa con los tiempos necesarios para cada instancia resuelta para alguno de los modelos, y proponga una estimación de la mejor performance de un modelo respecto al otro. Ver final de la práctica.

4. Continuando con el Problema de asignar viajes a conductores, llamamos Modelo 3 al que utiliza las siguientes variables binarias (ver [SLIDES2]):

$$z_{uv} = \begin{cases} 1, & \text{si un mismo conductor toma viaje } u \text{ y luego } v \\ 0, & \text{si no} \end{cases}$$

para todo $(u, v) \in A^*$. Este modelo utiliza las siguientes restricciones

$$\sum_{e \in P} z_e \leq |P| - 1 \quad \forall P \text{ camino prohibido en } D = (V, A^*)$$

donde P es un *camino prohibido* si contiene al menos dos arcos, y su primer y último arco pertenece a J^* .

- a) Pruebe que si $z \in [0, 1]^{A^*}$ satisface las restricciones asociadas a caminos prohibidos que sólo contienen dos arcos de J^* en sus extremos, z satisface todas las restricciones de caminos prohibidos.

Nota: Esta propiedad nos permite restringirnos sólo a caminos prohibidos que no tengan arcos de J^* excepto en sus extremos.

Sugerencia: Pruebe que si P es un camino prohibido que contiene k arcos que pertenecen a J^* con $k \geq 3$, entonces existen caminos prohibidos P_i con $i = 1, \dots, k - 1$ tales que cada P_i sólo contiene dos arcos de J^* en sus extremos y que si $z \in [0, 1]^{A^*}$ satisface las restricciones asociadas a P_i para todo i entonces z también satisface la asociada a P .

- b) Consideremos el problema de separación de las restricciones de caminos prohibidos,

ENTRADA: $D^* = (V, A^*)$ digrafo, $J^* \subset A^*$, $z^* \in [0, 1]^{A^*}$.

PREGUNTA: Existe un camino prohibido P tal que $\sum_{e \in P} z_e^* > |P| - 1$?

y el siguiente procedimiento (de complejidad polinomial en el tamaño de la entrada):

- Sea $\tilde{J} = \{e \in J^* : z_e^* > 0\}$.
- Para todo $(u, v) \in \tilde{J}$ y todo $(r, t) \in \tilde{J} \setminus \{(u, v)\}$ tal que $z_{uv}^* + z_{rt}^* > 1$ hacer
 - 1) Si $v = r$ entonces contestar **SI**, proponer $P = (u, v)(r, t)$ y PARAR.
 - 2) Obtener un vr -camino más corto \bar{P} en $D = (V, A)$ con costos $c_e = 1 - z_e^*$ para todo $e \in A$.
 - 3) Si $c(\bar{P}) < z_{uv}^* + z_{rt}^* - 1$ entonces contestar **SI**, proponer $P = (u, v)\bar{P}(r, t)$ y PARAR.
- Contestar **NO** y PARAR.

Pruebe que el procedimiento propuesto resuelve el problema de separación de las restricciones de caminos prohibidos.

Notas:

El Ejercicio 3 requiere el uso de las herramientas LPSolve, Zimpl y SCIP. LPSolve es un editor y optimizador de programas lineales enteros (PLEs), pero en este ejercicio sólo lo utilizamos como editor. Zimpl es un compilador de lenguaje de modelado matemático a PLEs y SCIP es el mejor solver no comercial disponible para optimizar PLEs. Comience realizando los siguientes pasos:

- Descargue el archivo denominado `Instalador-lpsolve-5.5.2.0.exe` de la pág. de la Cátedra y ejecútelos. Luego de realizada la instalación, puede borrar este archivo.
- Cree en su máquina un directorio de trabajo y descomprima el archivo `modelos.zip` de la pág. de la Cátedra, en este directorio. En la carpeta se crearán los siguientes archivos:
 - `modeloi.zpl` (para $i = 1, 2$): Modelo i en lenguaje de modelado matemático.
 - `instanciaj.zpl` (para $j = 1, \dots, 8$): Instancias en lenguaje de modelado matemático.
 - `correr_modeloi.bat`: Programa que utiliza Zimpl/SCIP para resolver `modeloi.zpl`.
 - `manual_zimpl.pdf`: Manual del lenguaje de modelado matemático.
 - `zimpl.exe`, `scip.exe`, `scip-3.0.2...dll`: Archivos auxiliares.
- En `modelo1.zpl` se encuentra actualmente el Modelo 1 con los datos de la Instancia 1. Para resolver este problema ejecute `correr_modelo1.bat`. Luego de finalizada la optimización, se creará un archivo `modelo1.log` con los resultados de la optimización. Lea este archivo con un editor (puede usar LPSolve para tal fin). El archivo se compone de varias secciones. Una de las secciones se llama “primal solution” y contiene el valor de la función objetivo y los valores de aquellas variables de la solución entera óptima que no son cero. En la sección “Statistics”, en la línea “Total Time” figura el tiempo de CPU transcurrido (en segundos).
- Para resolver el problema que utiliza el Modelo 1 y otra Instancia j , reemplace las líneas que corresponden a la Instancia 1 en el archivo `modelo1.zpl` por las que figuran en `instanciaj.zpl`. Luego realice el paso anterior.

Nota: Para efectuar dicho reemplazo, ejecute LPSolve y abra `instanciaj.zpl`. Elija todas las líneas y cópielas con Ctrl-C. Luego abra `modelo1.zpl` y reemplace las líneas 3-12 con las de la instancia (elija esas líneas y luego presione Ctrl-V). Finalmente, grabe el archivo `modelo1.zpl`.
- Para resolver el problema que utiliza el Modelo 2 y la Instancia 1, primero debe completar el archivo `modelo2.zpl` con las restricciones del mismo (actualmente en el archivo sólo se encuentran definidos los datos de la Instancia 1, los conjuntos A^* , $\delta_+^*(v)$, $\delta_-^*(v)$, las variables $z(k, u, v)$ y la función objetivo) siguiendo la lógica del archivo `modelo1.zpl`. Luego utilice el programa `correr_modelo2.bat`. En caso de error, el mismo será mostrado en la ventana que corre dicho programa. Si la optimización se lleva a cabo sin problemas, se creará un archivo `modelo2.log` con los resultados de la optimización. Verifique que la solución óptima coincida con la obtenida al resolver esta instancia con el Modelo 1.
- Para resolver el problema que utiliza el Modelo 2 y otra Instancia j , realice los mismos pasos propuestos para el Modelo 1.

A continuación se da el detalle de cada instancia, por si se desean comprobar los resultados:

Número	Conductores	Viajes	Minutos extra
1	2	7	78
2	5	19	32
3	10	39	47
4	15	59	101
5	20	77	172
6	25	97	455
7	30	117	588
8	35	137	687

Bibliografía:

[LCO] W. Cook, W. Cunningham, W. Pulleyblank A. Schrijver. *Combinatorial Optimization*. Wiley-Interscience.

[SLIDES1] M. E. Alvarado, G. Nasini, D. Severin. *Optimización de la Asignación Conductores-Viajes en una empresa de transporte: un algoritmo de mejoras basado en un modelo de Programación Lineal Entera*. LXII Reunión anual de la UMA (2013).

[SLIDES2] F. Bertero, G. Nasini, D. Severin. *El Problema de Asignación Conductores-Viajes en empresas de transporte público: Nuevos modelos de Programación Lineal Entera*. 12º Simposio Argentino de Investigación Operativa, 43ª JAIIO (2014).