MA4702. Programación Lineal Mixta : Teoría y Laboratorio. 2018.
Profesor: José Soto.
Profesor Auxiliar: Arturo Merino
Profesor Ayudante: Obed Ulloa

Tarea 3: Formulaciones extendidas mixtas pequeñas

Instrucciones.

- Usted debe entregar en ucursos antes de la hora de entrada al laboratorio los siguientes archivos: **`MMI-apellido.pdf`**, **`MMI-apellido.mod`**, **`MMI-apellido.run`**, **`MMI-apellido.txt`**.

Consideremos el siguiente problema.

MAX-MATCHING-IMPAR (MMI): Sea $G = (A \cup B, E)$ un grafo **bipartito** con partes A y B de tamaño n y tal que G contiene un matching perfecto. Sea $c \in \mathbb{R}^E$, encuentre un matching $M \subseteq E$ de cardinalidad **impar** que maximice $c(M) = \sum_{e \in M} c_e$.

Hay varias formas de atacar el problema de manera combinatorial, pero nuestra intención es encontrar una solución usando PL/PLE/PLM. En lo que sigue, llamemos $P_{\text{mf}}(G) = \{x \in \mathbb{R}^n_+ \colon x(\delta(v)) \leq 1\}$ al poliedro de matching fraccional de G y sea $I \subseteq [n]$ el conjunto de los números impares entre 1 y n.

Problema 1: Demuestre que para cada $k \in \mathbb{N}$ fijo, el dominio del problema $\max\{c^T x \colon x \in P_{\text{mf}}(G), x(E) = k\}$ es integral. Escriba su demostración en el archivo **`MMI-apellido.pdf`**.

Resolveremos **MMI** de 3 formas diferentes descritas a continuación

Forma 1: Usando que $(P) = \max_{k \in I} \max\{c^T x \colon x \in P_{\text{mf}}(G), x(E) = k\}$. Podemos resolver (P) resolviendo $|I|$ programas lineales y calculando el máximo global.

Forma 2: Esta forma es parecida a la anterior, pero resolvemos un solo PLM. Considere variables auxiliares $\{\lambda^k \in \{0,1\}\}_{k \in I}$, $\{y^k \in \mathbb{R}^E\}_{k \in I}$, donde $\lambda^k = 1$ se interpreta como $x(E) = k$. Considere el PLM siguiente:

$$
\begin{aligned}
(Q) \qquad & \max c^T x \\
& \sum_{k \in I} \lambda^k = 1; && \lambda \in \{0,1\}^I; \\
& y^k \leq \lambda^k \quad \forall k \in I; && y^k \in \mathbb{R}^E_+; \\
& x = \sum_{k \in I} y^k; && x(E) = \sum_{k \in I} k\lambda^k; \\
& x \in P_{\text{mf}}(G).
\end{aligned}
$$

Forma 3: Considere el siguiente PLM que agrega solo 1 variable entera (no binaria) para modelar el problema.

$$
\begin{aligned}
(R) \qquad & \max c^T x \\
& x \in P_{\text{mf}}(G) \\
& x(E) = 2t + 1 \\
& t \in \mathbb{Z}
\end{aligned}
$$

Problema 2: Demuestre que las formas 2 y 3 entregan, en x, el vector indicatriz de un max-matching-impar. Escriba su demostración en el archivo **MMI-apellido.pdf**.

Problema 3: Escriba un archivo **MMI-apellido.mod** que incluya todas las variables, restricciones y funciones objetivos necesarias para resolver el problema de las 3 formas propuestas. Suponga que en los datos recibirá conjuntos A, B y $E \subseteq A \times B$, así como una función de costos c. Use el comando **problem** para definir problemas separados que pueda invocar más adelante.

Escriba un script **MMI-apellido.run** que resuelva todas las instancias **bipartitoX.dat** con X de 1 a 10, usando los 3 métodos. Use lo siguiente como opciones globales al principio de su script.

```
problem Initial;
option solver cplex;
option cplex_options 'mipstartstatus=0 mipstartvalue=0 time=60';
```

Su script debe escribir en un archivo **MMI-apellido.txt** un reporte similar a la siguiente.

```
## Archivo (MMI-apellido.txt) ##

Instancia bipartito1.dat

num-vertices: xxxx
num-aristas : xxxx
tpo-metodo 1: xxxx  valor-MMI: xxxx  cardinal-MMI: xxxx
tpo-metodo 2: xxxx  valor-MMI: xxxx  cardinal-MMI: xxxx
tpo-metodo 3: xxxx  valor-MMI: xxxx  cardinal-MMI: xxxx

Instancia bipartito2.dat
...
```

La explicación de cada item está más abajo. Si el método i (para $i = 2$ o 3) no resuelven una instancia en el tiempo límite de 60 segundos, entonces debe reemplazar la línea que empieza por **tpo-metodo i:** por una línea que diga **el metodo i no resolvio la instancia**.

- **num-vertices**: es el número de vértices de la instancia.
- **num-aristas**: es el número de aristas de la instancia.
- **tpo-metodo i**: Tiempo de solver usado por el método i. Puede calcular esto, por ejemplo, definiendo en su script un parametro **auxiliar**, luego escribir **let auxiliar:=_total_solve_time;**, luego ejecutar su método en la instancia y finalmente escribir **display _total_solve_time - auxiliar;** para ver el tiempo de ejecución de dicho método.
- **valor-MMI**: el valor del MMI máximo calculado por su método.
- **cardinalidad-MMI**: el cardinal del MMI máximo calculado por su método.

Problema 4: Al final del archivo MMI-apellido.pdf, escriba conclusiones generales breves sobre los tres métodos usados. ¿Hay algún método consistentemente más rápido que otro? ¿Por qué cree usted que esto es así?