PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
Departamento de Ingeniería Industrial y de Sistemas

Curso: ICS1113-Optimización
Semestre: 2021-1
Profesores: G. Angulo, R. Cuadrado, J. González, S. Vásquez, F. Verastegui

Solución Tarea 5

Fecha de entrega: viernes 18 de junio - 23:59:00

Problema 1.

a) Considere la siguiente instancia del problema de la mochila con variables binarias:

$$\begin{array}{lrcl} \text{máx} & 17x_1 + 5x_2 + 10x_3 + x_4 & & \\ s.a & 2x_1 + x_2 + 4x_3 + 3x_4 & \leq & 5 \\ & x_1, x_2, x_3, x_4 & \in & \{0,1\} \end{array}$$

i. (**1.8ptos.**) Encuentre la solución óptima utilizando el algoritmo Branch & Bound. Está permitida la utilización de solvers de optimización para resolver cada nodo.

ii. (**0.2ptos. c/u**) Determine cuáles de las siguientes desigualdades de cubrimiento (covers) son válidas:

1) $x_1 + x_2 \leq 1$
2) $x_1 + x_2 + x_3 \leq 2$
3) $x_3 + x_4 \leq 1$
4) $x_1 + x_4 \leq 1$
5) $x_1 + x_2 + x_4 \leq 2$
6) $x_1 + x_2 + x_3 + x_4 \leq 3$

b) Suponga que está resolviendo un problema de optimización utilizando el algoritmo Branch & Bound donde el valor objetivo actual del incumbente es 60 y solo le queda revisar el nodo número 130. Al simplificar el problema, reemplazando el valor de las variables que ya fueron fijadas en el proceso de ramifiación que define el nodo considerado, se obtiene el siguiente problema:

$$\begin{array}{rlrcl} z_{130} = \text{mín} & cx_1 + 4x_2 + 5 & & \\ \text{s.a.} & ax_1 + 4x_2 & \geq & 5 \\ & x_1, x_2 & \in & \{0,1\} \end{array}$$

donde $a \in \mathbb{Z}$ y $c \in \mathbb{Z}^+$ son parámetros desconocidos. Defina los rangos de a y c para los cuales se cumplen los siguientes casos y evalúe cuál sería el valor de la función objetivo para cada uno de ellos:

i) (**1pto**) Nodo 130 es infactible

ii) (**1pto**) Nodo 130 es el nuevo incumbente.

iii) (**1pto**) Nodo 130 con solución entera peor que el incumbente.

Solución Problema 1.

a) i. Primero resolvemos la relajación lineal del problema:

$$\begin{array}{lrcl} \text{máx} & 17x_1 + 5x_2 + 10x_3 + x_4 & & \\ s.a & 2x_1 + x_2 + 4x_3 + 3x_4 & \leq & 5 \\ & x_1 & \leq & 1 \\ & x_2 & \leq & 1 \\ & x_3 & \leq & 1 \\ & x_4 & \leq & 1 \\ & x_1, x_2, x_3, x_4 & \geq & 0 \end{array}$$

cuya solución óptima es $x^0 = (1, 1, 0.5, 0)$ con valor $z^0 = 27$. Como x_3^0 es fraccionario, ramificamos con respecto a esta variable. Primero, fijamos $x_3 = 0$ y resolvemos el siguiente problema:

$$\begin{array}{lrcl} \text{máx} & 17x_1 + 5x_2 + x_4 & & \\ s.a & 2x_1 + x_2 + 3x_4 & \leq & 5 \\ & x_1 & \leq & 1 \\ & x_2 & \leq & 1 \\ & x_4 & \leq & 1 \\ & x_1, x_2, x_4 & \geq & 0 \end{array}$$

cuya solución óptima es $x^{(1)} = (1, 1, 0, 0.667)$ con valor $z^{(1)} = 22.667$. Como x_4^0 es fraccionario, ramificamos con respecto a esta variable. Primero, fijamos $x_4 = 0$ y resolvemos el siguiente problema:

$$\begin{array}{lrcl} \text{máx} & 17x_1 + 5x_2 & & \\ s.a & 2x_1 + x_2 & \leq & 5 \\ & x_1 & \leq & 1 \\ & x_2 & \leq & 1 \\ & x_1, x_2 & \geq & 0 \end{array}$$

cuya solución óptima es $x^{(1,1)} = (1, 1, 0, 0)$ con valor $z^{(1,1)} = 22$. Como esta es la mejor solción entera encontrada hasta el momento, actualizamos el valor del incumbente a 22.

Luego, resolvemos el segundo subproblema del subproblema 1 fijando $x_4 = 1$:

$$\begin{array}{lrcl} \text{máx} & 17x_1 + 5x_2 + 1 & & \\ s.a & 2x_1 + x_2 & \leq & 2 \\ & x_1 & \leq & 1 \\ & x_2 & \leq & 1 \\ & x_1, x_2 & \geq & 0 \end{array}$$

cuya solución óptima es $x^{(1,2)} = (1, 0, 0, 1)$ con valor $z^{(1,2)} = 18$. Como el valor de la solución es peor que el incumbente, podamos.

Ahora resolvemos el segundo subproblema fijando $x_3 = 1$:

$$\begin{array}{lrcl} \text{máx} & 17x_1 + 5x_2 + 10 + x_4 & & \\ s.a & 2x_1 + x_2 + 3x_4 & \leq & 1 \\ & x_1 & \leq & 1 \\ & x_2 & \leq & 1 \\ & x_4 & \leq & 1 \\ & x_1, x_2, x_4 & \geq & 0 \end{array}$$

cuya solución óptima es $x^{(2)} = (0.5, 0, 1, 0)$ con valor $z^{(2)} = 18.5$. Como el valor de la solución es peor que el incumbente, podamos. Como ya no se puede seguir ramificando, la solución óptima es $x^* = x^{(1,1)} = (1, 1, 0, 0)$ con valor $z^* = z^{(1,1)} = 22$.

ii. Sean $p_1, ..., p_5$ los pesos de cada artículo. Una desigualdad de cover $\sum_{i \in C} x_i \leq |C| - 1$ será válida si $\sum_{i \in C} p_i > 5$. Con esto, podemos determinar qué desigualdades de cubrimiento son válidas:

1) Inválida, porque $\sum_{i \in C} p_i < 5$
2) Válida, pero no es un cubrimiento mínimo porque $p_1 + p_3 = 5$
3) Válida
4) Inválida, porque $\sum_{i \in C} p_i = 5$
5) Válida
6) Válida, pero no es un cubrimiento mínimo porque $p_1 + p_3 = 5$

b) i) Para que el nodo sea infactible, basta que $a + 4 \leq 4 \Rightarrow a \leq 0$. En tal caso, el nodo recién procesado se elimina. Luego, la solución óptima es igual a 60.

ii) En este caso, basta con que el valor de la función objetivo del nodo 130 sea tal que $z_{130} \leq 59$, y sea entera, por lo que se actualiza el incumbente. Luego, se debe cumplir que $a + 4 \geq 5 \Rightarrow a \geq 1$ de modo de asegurar la factibilidad del problema.

Suponiendo que $\frac{c}{a} \leq \frac{4}{4}$ (me conviene más x_1) se tiene:

- Si $a \geq 5$, entonces $x_1 = \frac{5}{a}$ y $x_2 = 0$. Entonces para que la solución sea entera se tiene que imponer que $a = 5$, con lo que $x_1 = 1$, y entonces $z_{130} = 5 + c$.
- Si $a < 5$, entonces $4 \geq 5 - a$ con lo que $x_1 = 1$ y $x_2 = \frac{5-a}{4}$. Entonces para que la solución sea entera se tiene que imponer que $a = 1$, con lo que $x_2 = 1$, y entonces $z_{130} = 5 + c + 4$.

iii) En este caso, basta con que el valor de la función objetivo del nodo 130 sea tal que $z_{130} > 60$ y que la solución sea entera. Además, se debe cumplir que $a + 4 \geq 5 \Rightarrow a \geq 1$ de modo de asegurar la factibilidad del problema.

Suponiendo que $\frac{c}{a} \leq \frac{4}{4}$ (me conviene más x_1) se tiene:

- Si $a \geq 5$, entonces $x_1 = \frac{5}{a}$ y $x_2 = 0$. Entonces para que la solución sea entera se tiene que imponer que $a = 5$, con lo que $x_1 = 1$, y entonces $z_{130} = 5 + c$.
- Si $a < 5$, entonces $4 \geq 5 - a$ con lo que $x_1 = 1$ y $x_2 = \frac{5-a}{4}$. Entonces para que la solución sea entera se tiene que imponer que $a = 1$, con lo que $x_2 = 1$, y entonces $z_{130} = 5 + c + 4$.

Problema 2.

Dado un conjunto N de ciudades y costos de viaje $c_{ij} \in \mathbb{R}^{N\times N}$ entre cada par de ciudades, el problema del vendedor viajero (o TSP, por sus siglas en inglés) consiste en encontrar la forma mas barata de visitar todas las ciudades exactamente una vez, y volver al punto de partida. Dos formulaciones conocidas para este problema son:

$$
\begin{aligned}
\mathcal{P}_{MTZ}) \quad & \text{mín} && \sum_{i\in N}\sum_{j\in N:i\neq j} c_{ij}x_{ij} && && \\
& s.a && \sum_{i\in N:i\neq j} x_{ij} && = \quad 1 && \forall j \in N \\
& && \sum_{j\in N:i\neq j} x_{ij} && = \quad 1 && \forall i \in N \\
& && u_i - u_j + nx_{ij} && \leq \quad n-1 && \forall i \in \{2,...,|N|\}, \forall j \in \{2,...,|N|\} \\
& && u_i && \geq \quad 1 && \forall i \in \{2,...,|N|\} \\
& && u_i && \leq \quad n-1 && \forall i \in \{2,...,|N|\} \\
& && u_i && \in \quad \mathbb{Z} && \forall i \in \{2,...,|N|\} \\
& && x_{ij} && \in \quad \{0,1\} && \forall i \in N, \forall j \in N
\end{aligned}
$$

$$
\begin{aligned}
\mathcal{P}_{DFJ}) \quad & \text{mín} && \sum_{i\in N}\sum_{j\in N:i\neq j} c_{ij}x_{ij} && && \\
& s.a && \sum_{i\in N:i\neq j} x_{ij} && = \quad 1 && \forall j \in N \\
& && \sum_{j\in N:i\neq j} x_{ij} && = \quad 1 && \forall i \in N \\
& && \sum_{i\in Q}\sum_{j\in Q:j\neq i} x_{ij} && \leq \quad |Q|-1 && \forall Q \subsetneq N, |Q| \geq 2 \\
& && x_{ij} && \in \quad \{0,1\} && \forall i \in N, \forall j \in N
\end{aligned}
$$

donde $n = |N|$.

a) (**1pto.**) Escriba el significado de las variables, función objetivo y restricciones para cada formulación.

b) (**3ptos.**) Implemente un programa en Python/Gurobi que resuelva la instancia adjunta del TSP para las formulaciones $\mathcal{P}_{MTZ}$) y $\mathcal{P}_{DFJ}$). Adjunte *screenshots* de su código.

c) (**2ptos.**) Compare y comente los resultados obtenidos en cuanto al valor óptimo, valor óptimo de la relajación lineal, cantidad de nodos explorados y tiempo de ejecución.

Solución Problema 2.

a) Las variables binarias $x_{ij} \in \mathbb{R}^{N\times N}$ valen 1 si el arco $(i,j) \in \mathbb{R}^{N\times N}$ está presente en el tour.

La función objetivo minimiza el costo total del viaje sumando sobre todos los posibles arcos.

Los primeros dos conjuntos de restricciones obligan a que en todas las ciudades exista solo un arco que entre a ella y solo uno que salga.

Para la formulación $\mathcal{P}_{MTZ}$), $u_i \in N$ son variables auxiliares enteras que indican el orden del tour. La restricciones que modelan este comportamiento son:

$$u_i - u_j + nx_{ij} \leq n-1 \quad \forall i \in \{2,...,|N|\}, \forall j \in \{2,...,|N|\}$$

mientras que las cotas $1 \leq u_i \leq n-1$ aseguran que las ciudades estén en secuencia y elimina simetrías entre distintas soluciones que representan el mismo tour.

Para la formulación $\mathcal{P}_{DFJ}$), el tercer conjunto de restricciones impide la formación de **subtours**. De esta forma, se asegura que la solución sea un solo tour y no la union de subtours.

b) Los *scripts* necesarios para resolver ambas formulaciones del TSP son:

```
# MTZ
m = Model()
x = m.addVars(N, N, vtype=GRB.BINARY)
u = m.addVars(range(1, n), vtype=GRB.INTEGER, lb=1, ub=n-1)
m.setObjective(sum(c[i, j] * x[i, j]
                for i in N for j in N if i != j))
m.addConstrs(sum(x[i, j] for i in N if i != j) == 1 for j in N)
m.addConstrs(sum(x[i, j] for j in N if i != j) == 1 for i in N)
m.addConstrs(u[i] - u[j] + n*x[i, j] <= n - 1
             for i in range(1, n) for j in range(1, n) if i != j)
m.optimize()
```

```
# DFJ
m = Model()
x = m.addVars(N, N, vtype=GRB.BINARY)
m.setObjective(sum(c[i, j] * x[i, j]
                for i in N for j in N if i != j))
m.addConstrs(sum(x[i, j] for i in N if i != j) == 1 for j in N)
m.addConstrs(sum(x[i, j] for j in N if i != j) == 1 for i in N)
for Q in get_sets(N):
    m.addConstr(sum(x[i, j] for i in Q for j in Q if i != j)
                <= len(Q) - 1)
m.optimize()
```

c) Si bien los tiempos pueden cambiar, el modelo $\mathcal{P}_{DFJ}$) debería ser considerablemente más lento debido a la cantidad de restricciones que se deben procesar. Cabe notar que la formulación $\mathcal{P}_{DFJ}$) permite encontrar la solución óptima solo resolviendo la relajación lineal del problema.

	$\mathcal{P}_{MTZ}$	$\mathcal{P}_{DFJ}$
Valor óptimo	461.891	461.891
Valor óptimo relajación lineal	392.435	461.891
Cantidad de nodos explorados	155	0
Tiempo de ejecución	0,1s	25s