Pontificia Universidad Católica de Chile
Departamento de Ingeniería Industrial y de Sistemas
ICS1113 - Optimización

Tarea 3

Profesores: Margarita Castro, Raimundo Cuadrado, Marcel Favereau, Mathias Klapp, Esteban Salgado
Ayudante coordinador tareas: José Tomás Grez

Problema 1

La empresa Koka Kola produce K bebidas distintas en I fábricas. Estas bebidas se distribuyen a J países donde son demandadas. Se sabe que el costo de producir un litro de la bebida $k \in \{1, ..., K\}$ en la fábrica $i \in \{1, ..., I\}$ es v_{ik}; el costo de transportar un litro de bebida k de la fábrica i al país $j \in \{1, ..., J\}$ es c_{ijk}; y la demanda del país j por la bebida k es d_{jk} litros. Por otro lado, se sabe que cada fábrica $i \in \{1, ..., I\}$ se demora q_{ik} horas en producir un litro de la bebida k y puede funcionar a lo más Q_i horas (total de horas para todas las bebidas). Además, la cantidad máxima de litros de bebida k que puede producir la fábrica i es M_{ik}.

(a) Koka Kola lo contrata para que formule un modelo de programación lineal que minimicen los costos totales. Su modelo debe determinar las cantidades óptimas que se deben producir en cada fábrica y enviar a cada país (medido en litros), para cumplir con la demanda y no superar la capacidad de producción.

(b) Considere que si se produce algún litro de la bebida k en la fábrica i, se deben producir como mínimo m_{ik} litros. Escriba los cambios que le realizaría al modelo planteado en (a) para cumplir con la nueva situación.

Solución

(a) **Variables**

- x_{ijk}: cantidad de litros de la bebida $k \in \{1, ..., K\}$ a transportar desde la planta $i \in \{1, ..., I\}$ al país $j \in \{1, ..., J\}$
- z_{ik}: cantidad de litros de la bebida $k \in \{1, ..., K\}$ producidos en la planta $i \in \{1, ..., I\}$

Función Objetivo

$$\text{mín} \left(\sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{k=1}^{K} c_{ijk} x_{ijk} + \sum_{i=1}^{I} \sum_{k=1}^{K} v_{ik} z_{ik} \right)$$

Restricciones

1. Relación entre variables:

$$\sum_{j=1}^{J} x_{ijk} = z_{ik} \quad \forall i \in \{1, ..., I\}, \forall k \in \{1, ..., K\}$$

2. No sobrepasar las horas disponibles de cada fábrica:

$$\sum_{k=1}^{K} q_{ik} \cdot z_{ik} \leq Q_i \quad \forall i \in \{1, ..., I\}$$

3. Cumplir con la demanda:

$$\sum_{i=1}^{I} x_{ijk} = d_{jk} \quad \forall j \in \{1, ..., J\}, k \in \{1, ..., K\}$$

4. No sobrepasar la capacidad productiva de cada fábrica:

$$z_{ik} \leq M_{ik} \quad \forall i \in \{1, ..., I\}, k \in \{1, ..., K\}$$

5. Naturaleza de las variables:

$$\begin{aligned} x_{ijk} \geq 0 \quad & \forall i \in \{1, ..., I\}, j \in \{1, ..., J\}, k \in \{1, ..., K\} \\ z_{ik} \geq 0 \quad & \forall i \in \{1, ..., I\}, k \in \{1, ..., K\} \end{aligned}$$

(b) Se agregan las variables binarias:

y_{ik}: toma valor 1 si se fabrica la bebida k en la planta i y 0 en otro caso.

También se agregan las restricciones de mínima producción:

$$\begin{aligned} z_{ik} \leq M_{ik} y_{ik} \quad & \forall i \in \{1, ..., I\}, \forall k \in \{1, ..., K\} \\ z_{ik} \geq m_{ik} y_{ik} \quad & \forall i \in \{1, ..., I\}, \forall k \in \{1, ..., K\} \end{aligned}$$

La primera de estas reemplaza a la restricción 4 planteada en (a). Por último, hay que relajar la primera restricción de (a), ya que, por las producciones mínimas podría producirse más de lo que se demanda:

$$\sum_{j=1}^{J} x_{ijk} \leq z_{ik} \quad \forall i \in \{1, ..., I\}, \forall k \in \{1, ..., K\}$$

Problema 2

Considere el siguiente problema de programación lineal entera:

$$\begin{array}{llrcl} (P) & \text{máx} & 2x_1 + 5x_2 & & \\ & s.a. & 5x_1 + 7x_2 & \leq & 37 \\ & & 9x_1 + 11x_2 & \leq & 43 \\ & & -4x_1 + 5x_2 & \leq & 13 \\ & & x_1, x_2 & \in & \mathbb{Z} \end{array}$$

Resuelva el problema ejecutando el algoritmo de Branch & Bound, indicando claramente los subproblemas resueltos, sus soluciones y valor de la función objetivo. Construya el árbol con todos los subproblemas resueltos.
Nota: Para resolver cada sub-problema puede utilizar cualquier método que estimes conveniente.

Solución *En el anexo se puede ver el código utilizado en Gurboi para resolver cada sub-problema.* El primer problema será:

$$\begin{array}{llrcl} (P0) & \text{máx} & 2x_1 + 5x_2 & & \\ & s.a. & 5x_1 + 7x_2 & \leq & 37 \\ & & 9x_1 + 11x_2 & \leq & 43 \\ & & -4x_1 + 5x_2 & \leq & 13 \\ & & x_1, x_2 & \in & \mathbb{R} \end{array}$$

Cuya solución es:

x_1	x_2	Valor función objetivo
0.808989	3.24719	17.8539

Ramificando por x_1, por el lado izquierdo del árbol hay que agregar la restricción $x_1 \leq 0$, con lo que queda el problema:

$$\begin{array}{llrcl} (P1) & \text{máx} & 2x_1 + 5x_2 & & \\ & s.a. & 5x_1 + 7x_2 & \leq & 37 \\ & & 9x_1 + 11x_2 & \leq & 43 \\ & & -4x_1 + 5x_2 & \leq & 13 \\ & & x_1, x_2 & \in & \mathbb{R} \\ & & x_1 & \leq & 0 \end{array}$$

Cuya solución es:

x_1	x_2	Valor función objetivo
0	2.6	13

Mientras que por el lado derecho, hay que agregar la restricción $x_1 \geq 1$ el problema que queda es:

$$
\begin{array}{lllll}
(P2) & \text{máx} & 2x_1 + 5x_2 & & \\
& s.a. & 5x_1 + 7x_2 & \leq & 37 \\
& & 9x_1 + 11x_2 & \leq & 43 \\
& & -4x_1 + 5x_2 & \leq & 13 \\
& & x_1, x_2 & \in & \mathbb{R} \\
& & x_1 & \geq & 1
\end{array}
$$

Resolviendo con Gurobi: Con lo que la solucón del problema es:

x_1	x_2	Valor función objetivo
1	3.09091	17.454545

Todavía no se puede podar ninguna rama, por lo que ramificamos (P_1). Por la izquierda, agregamos la restricción $x_2 \leq 2$:

$$
\begin{array}{lllll}
(P3) & \text{máx} & 2x_1 + 5x_2 & & \\
& s.a. & 5x_1 + 7x_2 & \leq & 37 \\
& & 9x_1 + 11x_2 & \leq & 43 \\
& & -4x_1 + 5x_2 & \leq & 13 \\
& & x_1, x_2 & \in & \mathbb{R} \\
& & x_1 & \leq & 0 \\
& & x_2 & \leq & 2
\end{array}
$$

Resolviendo con Gurobi: Con lo que la solucón del problema es:

x_1	x_2	Valor función objetivo
0	2	10

Este es nuestro primer problema que entrega una solución entera, por lo que el incumbente ahora es 13 y se poda esta rama.

Por el lado derecho agregamos la restricción $x_2 \geq 3$:

$$
\begin{array}{lllll}
(P4) & \text{máx} & 2x_1 + 5x_2 & & \\
& s.a. & 5x_1 + 7x_2 & \leq & 37 \\
& & 9x_1 + 11x_2 & \leq & 43 \\
& & -4x_1 + 5x_2 & \leq & 13 \\
& & x_1, x_2 & \in & \mathbb{R} \\
& & x_1 & \leq & 0 \\
& & x_2 & \geq & 3
\end{array}
$$

Este problema es infactible, por lo que se poda esta rama.

Ahora, por el otro lado del árbol, agregamos la restricción $x_2 \leq 3$ por el lado izquierdo, quedando el problema:

$$
\begin{array}{llrcl}
(P5) & \text{máx} & 2x_1 + 5x_2 & & \\
 & s.a. & 5x_1 + 7x_2 & \leq & 37 \\
 & & 9x_1 + 11x_2 & \leq & 43 \\
 & & -4x_1 + 5x_2 & \leq & 13 \\
 & & x_1, x_2 & \in & \mathbb{R} \\
 & & x_1 & \geq & 1 \\
 & & x_2 & \leq & 3
\end{array}
$$

Cuya solución es:

x_1	x_2	Valor función objetivo
1.11	3	17.22

Por el lado derecho, se agrega la restricción $x_2 \geq 4$:

$$
\begin{array}{llrcl}
(P6) & \text{máx} & 2x_1 + 5x_2 & & \\
 & s.a. & 5x_1 + 7x_2 & \leq & 37 \\
 & & 9x_1 + 11x_2 & \leq & 43 \\
 & & -4x_1 + 5x_2 & \leq & 13 \\
 & & x_1, x_2 & \in & \mathbb{R} \\
 & & x_1 & \geq & 1 \\
 & & x_2 & \geq & 4
\end{array}
$$

Este problema es infactible, por lo que también se poda esta rama.
La única rama que sigue quedando por explorar es la que continúa por (P5). Por la izquierda le agregamos la restricción $x_1 \leq 1$:

$$
\begin{array}{llrcl}
(P7) & \text{máx} & 2x_1 + 5x_2 & & \\
 & s.a. & 5x_1 + 7x_2 & \leq & 37 \\
 & & 9x_1 + 11x_2 & \leq & 43 \\
 & & -4x_1 + 5x_2 & \leq & 13 \\
 & & x_1, x_2 & \in & \mathbb{R} \\
 & & x_1 & \geq & 1 \\
 & & x_2 & \leq & 3 \\
 & & x_1 & \leq 1 &
\end{array}
$$

Cuya solución es:

x_1	x_2	Valor función objetivo
1	3	17

Llegamos a otra solución entera, que es mejor que el incumbente. Ahora el incumbente es 17 y se poda esta rama.
Por el lado derecho, se agrega la restricción $x_1 \geq 2$:

$$
\begin{array}{llrcl}
(P8) & \text{máx} & 2x_1 + 5x_2 & & \\
& s.a. & 5x_1 + 7x_2 & \leq & 37 \\
& & 9x_1 + 11x_2 & \leq & 43 \\
& & -4x_1 + 5x_2 & \leq & 13 \\
& & x_1, x_2 & \in & \mathbb{R} \\
& & x_1 & \geq & 1 \\
& & x_2 & \leq & 3 \\
& & x_1 & \geq 2 &
\end{array}
$$

Problema que es infactible, por lo que también se poda esta rama y queda terminado el problema. La solución está dada por (P7): $x_1 = 1, x_2 = 3, z^* = 17$.

El árbol queda de la siguiente manera:

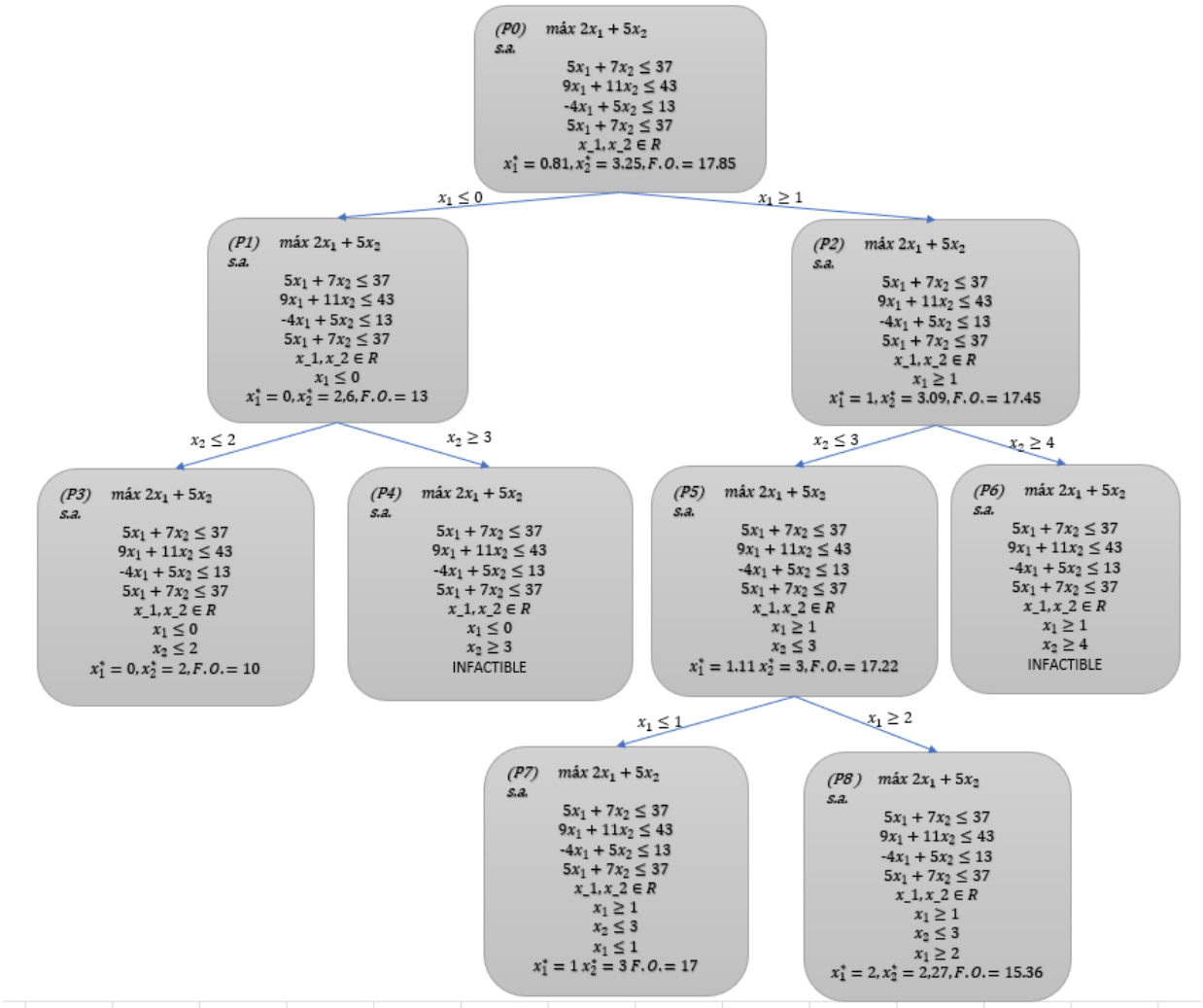


Figura 1: Árbol

Anexo

```python
from gurobipy import Model, GRB

m = Model()

# Variables
x_1 = m.addVar(vtype=GRB.CONTINUOUS , name="x")
x_2 = m.addVar(vtype=GRB.CONTINUOUS, name="y")

# Restricciones
m.addConstr(5*x_1 + 7*x_2 <= 37, "c0")
m.addConstr(9*x_1 + 11*x_2 <= 43, "c1")
m.addConstr(-4*x_1 + 5*x_2 <= 13, "c2")
# Función objetivo
m.setObjective(2*x_1 + 5*x_2, GRB.MAXIMIZE)

# Solución
m.optimize()
m.printAttr("x")
```

Figura 2: P_0

```python
from gurobipy import Model, GRB

m = Model()

# Variables
x_1 = m.addVar(vtype=GRB.CONTINUOUS , name="x")
x_2 = m.addVar(vtype=GRB.CONTINUOUS, name="y")

# Restricciones
m.addConstr(5*x_1 + 7*x_2 <= 37, "c0")
m.addConstr(9*x_1 + 11*x_2 <= 43, "c1")
m.addConstr(-4*x_1 + 5*x_2 <= 13, "c2")
m.addConstr(x_1 <= 0, "c3")
# Función objetivo
m.setObjective(2*x_1 + 5*x_2, GRB.MAXIMIZE)

# Solución
m.optimize()
m.printAttr("x")
print(m.objVal)
```

Figura 3: P_1

```python
from gurobipy import Model, GRB

m = Model()

# Variables
x_1 = m.addVar(vtype=GRB.CONTINUOUS , name="x")
x_2 = m.addVar(vtype=GRB.CONTINUOUS, name="y")

# Restricciones
m.addConstr(5*x_1 + 7*x_2 <= 37, "c0")
m.addConstr(9*x_1 + 11*x_2 <= 43, "c1")
m.addConstr(-4*x_1 + 5*x_2 <= 13, "c2")
m.addConstr(x_1 >= 1, "c3")
# Función objetivo
m.setObjective(2*x_1 + 5*x_2, GRB.MAXIMIZE)

# Solución
m.optimize()
m.printAttr("x")
print(m.objVal)
```

Figura 4: P_2

```python
from gurobipy import Model, GRB

m = Model()

# Variables
x_1 = m.addVar(vtype=GRB.CONTINUOUS , name="x")
x_2 = m.addVar(vtype=GRB.CONTINUOUS, name="y")

# Restricciones
m.addConstr(5*x_1 + 7*x_2 <= 37, "c0")
m.addConstr(9*x_1 + 11*x_2 <= 43, "c1")
m.addConstr(-4*x_1 + 5*x_2 <= 13, "c2")
m.addConstr(x_1 <= 0, "c3")
m.addConstr(x_2 <= 2)
# Función objetivo
m.setObjective(2*x_1 + 5*x_2, GRB.MAXIMIZE)

# Solución
m.optimize()
m.printAttr("x")
print(m.objVal)
```

Figura 5: P_3

```python
from gurobipy import Model, GRB

m = Model()

# Variables
x_1 = m.addVar(vtype=GRB.CONTINUOUS , name="x")
x_2 = m.addVar(vtype=GRB.CONTINUOUS, name="y")

# Restricciones
m.addConstr(5*x_1 + 7*x_2 <= 37, "c0")
m.addConstr(9*x_1 + 11*x_2 <= 43, "c1")
m.addConstr(-4*x_1 + 5*x_2 <= 13, "c2")
m.addConstr(x_1 <= 0, "c3")
m.addConstr(x_2 >= 3)
# Función objetivo
m.setObjective(2*x_1 + 5*x_2, GRB.MAXIMIZE)

# Solución
m.optimize()
m.printAttr("x")
print(m.objVal)
```

Figura 6: P_4

```python
from gurobipy import Model, GRB

m = Model()

# Variables
x_1 = m.addVar(vtype=GRB.CONTINUOUS , name="x")
x_2 = m.addVar(vtype=GRB.CONTINUOUS, name="y")

# Restricciones
m.addConstr(5*x_1 + 7*x_2 <= 37, "c0")
m.addConstr(9*x_1 + 11*x_2 <= 43, "c1")
m.addConstr(-4*x_1 + 5*x_2 <= 13, "c2")
m.addConstr(x_1 >= 1, "c3")
m.addConstr(x_2 <= 3)
# m.addConstr(x_1 >= 2)
# Función objetivo
m.setObjective(2*x_1 + 5*x_2, GRB.MAXIMIZE)

# Solución
m.optimize()
m.printAttr("x")
print(m.objVal)
```

Figura 7: P_5

```python
from gurobipy import Model, GRB

m = Model()

# Variables
x_1 = m.addVar(vtype=GRB.CONTINUOUS , name="x")
x_2 = m.addVar(vtype=GRB.CONTINUOUS, name="y")

# Restricciones
m.addConstr(5*x_1 + 7*x_2 <= 37, "c0")
m.addConstr(9*x_1 + 11*x_2 <= 43, "c1")
m.addConstr(-4*x_1 + 5*x_2 <= 13, "c2")
m.addConstr(x_1 >= 1, "c3")
m.addConstr(x_2 >= 4)
# m.addConstr(x_1 >= 2)
# Función objetivo
m.setObjective(2*x_1 + 5*x_2, GRB.MAXIMIZE)

# Solución
m.optimize()
m.printAttr("x")
print(m.objVal)
```

Figura 8: P_6

```python
from gurobipy import Model, GRB

m = Model()

# Variables
x_1 = m.addVar(vtype=GRB.CONTINUOUS , name="x")
x_2 = m.addVar(vtype=GRB.CONTINUOUS, name="y")

# Restricciones
m.addConstr(5*x_1 + 7*x_2 <= 37, "c0")
m.addConstr(9*x_1 + 11*x_2 <= 43, "c1")
m.addConstr(-4*x_1 + 5*x_2 <= 13, "c2")
m.addConstr(x_1 >= 1, "c3")
m.addConstr(x_2 <= 3)
m.addConstr(x_1 <= 1)
# Función objetivo
m.setObjective(2*x_1 + 5*x_2, GRB.MAXIMIZE)

# Solución
m.optimize()
m.printAttr("x")
print(m.objVal)
```

Figura 9: P_7

```python
from gurobipy import Model, GRB

m = Model()

# Variables
x_1 = m.addVar(vtype=GRB.CONTINUOUS , name="x")
x_2 = m.addVar(vtype=GRB.CONTINUOUS, name="y")

# Restricciones
m.addConstr(5*x_1 + 7*x_2 <= 37, "c0")
m.addConstr(9*x_1 + 11*x_2 <= 43, "c1")
m.addConstr(-4*x_1 + 5*x_2 <= 13, "c2")
m.addConstr(x_1 >= 1, "c3")
m.addConstr(x_2 <= 3)
m.addConstr(x_1 >= 2)
# Función objetivo
m.setObjective(2*x_1 + 5*x_2, GRB.MAXIMIZE)

# Solución
m.optimize()
m.printAttr("x")
print(m.objVal)
```

Figura 10: P_8

Problema 3

Determine si las siguientes afirmaciones son verdaderas o falsas. Argumente su respuesta.

(a) Para que un problema de flujo a costo mínimo (PFCM) sobre un grafo dirigido sea factible basta con que la oferta total sea igual a la demanda total.

(b) Si la relajación lineal de un problema entero admite solución óptima, entonces el problema entero también admite solución óptima.

(c) Considere un problema de programación lineal entera compuesto solo por variables binarias y resuelto con B&B. En el peor de los casos, se tendrán que resolver exactamente 2^{n+1} sub-problemas, siendo n el número de variables.

(d) Sea un problema de flujo a costo mínimo, donde las ofertas y demandas en los n nodos y las capacidades de los m arcos son números enteros. Si las variables de flujo tienen que ser enteras, en el peor de los casos se deben resolver 2^{n+m} sub-problemas ejecutando B&B.

Solución

(a) Falso. Es necesario pero no es lo único que basta. Puede ocurrir que la orientación de los arcos haga que sea imposible enviar flujo y cumplir ciertas demandas. También se puede argumentar que es falso ya que las capacidades de los arcos pueden hacer que el problema sea infactible.

(b) Falso. Contraejemplo:

$$\begin{array}{ll} \text{máx} & x \\ s.a. & 0.2 \leq x \leq 0.4 \\ & x \in \mathbb{Z} \end{array}$$

La relación lineal tiene solución óptima, con $x^* = 0.4$, pero el problema entero es infactible.

(c) Falso. Se tendrán que resolver como máximo $2^{n+1} - 1$ problemas.

(d) Falso. Dada la propiedad de integralidad de los problemas de flujo con lado derecho entero y cotas enteras, la relajación lineal tiene solución óptima entera, por lo que se requiere resolver solamente dicho subproblema.