Hands-on with Gurobi

章 宇

y.zhang@swufe.edu.cn

A Simple LOP Example

$$\begin{aligned}
\max \quad & x_1 + 2x_2 + 5x_3 \\
\text{s.t.} \quad & -x_1 + x_2 + 3x_3 \leq -5, \\
& x_1 + 3x_2 - 7x_3 \leq 10, \\
& x_1 \leq 10, \\
& x_1, x_2, x_3 \geq 0.
\end{aligned}$$

A Simple LOP Example

```python
# Create a new model
m = Model("simplelp")

# Create variables
x1 = m.addVar(ub = 10, name="x1")
x2 = m.addVar(name="x2")
x3 = m.addVar(name="x3")

# Set objective
m.setObjective(x1 + 2 * x2 + 5 * x3, GRB.MAXIMIZE)

# Add constraint:
m.addConstr(-x1 + x2 + 3*x3 <= -5, "c0")

# Add constraint:
m.addConstr(x1 + 3*x2 - 7*x3 >= 10, "c1")

m.optimize()
```

A Simple LOP Example

- An alternative way

```
# Create a new model
m = Model("simplelp")

# Create variables
ubd = [10, GRB.INFINITY, GRB.INFINITY]
x = m.addVars(3, ub = ubd, name="x")

# Set objective
c = [1, 2, 5]

m.setObjective(  sum(x[i] * c[i] for i in range(3)) , GRB.MAXIMIZE)

# Add constraints:
b = [-5, -10]
A = np.array([[-1, 1, 3], [-1, -3, 7]])

m.addConstrs( quicksum(A[i,j] * x[j] for j in range(3)) <= b[i] for i in range(2))

m.optimize()
```

Transportation Problem

A company has three PC assembly plants at locations, 1, 2 and 3, with monthly production capacity of 1700 units, 2000 units, and 1700 units, respectively. Their PC's are sold through four retail outlets in locations A, B, C,D, with monthly orders of 1700 units, 1000 units, 1500 units, and 1200 units respectively.

Shipping costs ($/unit)	Retailer A	Retailer B	Retailer C	Retailer D
Plant 1	5	3	2	6
Plant 2	7	7	8	10
Plant 3	6	5	3	8

Transportation Problem

- Formulation

$$
\begin{aligned}
\min \quad & \sum_{i\in P}\sum_{j\in R} c_{ij}x_{ij} && \\
\text{s.t.} \quad & \sum_{i\in P} x_{ij} = d_j && \forall j \in R \\
& \sum_{j\in R} x_{ij} \leq s_i && \forall i \in P \\
& x_{ij} \geq 0 && \forall i \in P, j \in R
\end{aligned}
$$

Transportation Problem

```
from gurobipy import *
import numpy as np

# data
supply = [1700, 2000, 1700]
demand =[1700, 1000, 1500, 1200]
c = np.array([[5, 3, 2, 6],
              [7, 7, 8, 10],
              [6, 5, 3, 8]])

# model
m = Model('transportation')

# decision variable
x = m.addVars(3, 4, name='transp')

# objective
m.setObjective(quicksum(x[i,j]*c[i,j] for i in range(3) for j in range(4)),sense=GRB.MINIMIZE)

# constraints
m.addConstrs(((quicksum(x[i,j] for i in range(3)) == demand[j]) for j in range(4)),name='demand_con')
m.addConstrs(((quicksum(x[i,j] for j in range(4)) == supply[i]) for i in range(3)),name='supply_con')


# optimize
m.optimize()
```

Transportation Problem

```python
from gurobipy import *

# data
n_plant = 3
n_retailer = 4
cost = [[5, 3, 2, 6],
        [7, 7, 8, 10],
        [6, 5, 3, 8]]
cost_dict = {(i,j):cost[i][j] for i in range(n_plant) for j in range(n_retailer)}
supply = [1700, 2000, 1700]
demand = [1700, 1000, 1500, 1200]

# model
m = Model('transp')

# decision varialbes
x = m.addVars(n_plant, n_retailer, vtype=GRB.INTEGER, name='transp')

# objective
m.setObjective(x.prod(cost_dict), sense=GRB.MINIMIZE)

# constraints
m.addConstrs((x.sum('*',j) == demand[j] for j in range(n_retailer)), name='demand')
m.addConstrs((x.sum(i,'*') <= supply[i] for i in range(n_plant)), name='supply')

# optimize
m.optimize()
```

Investment under Taxation

- You purchased s_i shares of stock i at price $p_i, i = 1, \dots, n$
- Current price of stock i is q_i
- You expect that the price of stock i one year later will be r_i
- You pay a capital gain tax at the rate of 30% on any capital gains at the time of sale
- You want to raise K amount of cash after taxes
- You pay 1% in transaction costs
- Example: You sell 1,000 shares at \$50 per share; you have bought them at \$30; Net Cash is,

$$50 \times 1,000 - 0.3 \times (50 - 30) \times 1,000 - 0.01 \times 50 \times 1,000 = \$43,500.$$

Investment under Taxation

- Objective: Maximize the expected return (next year)
- Constraints: Able to raise the fund K.

Investment under Taxation

- Formulation

Let x_i be the amount of share i to sell.

$$\begin{aligned} \max \quad & \sum_{i=1}^{N} r_i(s_i - x_i) \\ \text{s.t.} \quad & \sum_{i=1}^{N} \left(q_i x_i - 0.3 \max\{q_i - p_i, 0\} x_i - 0.01 q_i x_i\right) \geq K \\ & 0 \leq x_i \leq s_i \quad \forall i \in \{1, \ldots, N\} \end{aligned}$$

- Data
  - Given in data1.csv
  - $K = 9000$

References

- Detailed tutorial
  - https://www.bilibili.com/video/BV1V7411X7pn/?spm_id_from=333.788.recommend_more_video.0
  - From 44'35''
- User manual
  - https://www.gurobi.com/wp-content/plugins/hd_documentations/documentation/9.0/refman.pdf