VG441 Problem Set 3

Pan, Chongdan ID:516370910121

July 19, 2020

1 Problem 1

The problem can be consider as a MILP in the following ways:

$$\min \sum_{i=1}^{m} s_i$$

s.t.

$$s_i = \{0, 1\}, \forall i = 0, 1 \cdots m$$

$$x_{ij} = \begin{cases} 1 & e_j \in S_i \\ 0 & e_j \notin S_i \end{cases}$$

$$y_j = \max\{s_1 x_{1j}, s_2 x_{2j} \cdots s_i x_{ij}\} = 1, \forall j = 0, 1 \cdots n$$

i is the number of sets, s_i represents whether we select it, $s_i = 1$ if we select the i_{th} set, otherwise 0 j is the number of elements, and x_{ij} represents whether $e_j \in S_i$, if it's covered by S_i, then 1, otherwise 1. y_j represents whether the j_{th} element e_j is covered by our selection, it's 1 if covered, otherwise 0. Hence all y_j should be 1 because we want to cover all elements.

Our objective is to minimize the sum of x_{s_i}, which represents the number of set selected.

```
Using license file C:\Users\PANDA\gurobi.lic
Academic license - for non-commercial use only
Gurobi Optimizer version 9.0.2 build v9.0.2rc0 (win64)
Optimize a model with 48 rows, 53 columns and 61 nonzeros
Model fingerprint: 0x2644511e
Model has 8 general constraints
Variable types: 48 continuous, 5 integer (5 binary)
Coefficient statistics:
  Matrix range     [1e+00, 1e+00]
  Objective range  [1e+00, 1e+00]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 1e+00]
Presolve removed 48 rows and 53 columns
Presolve time: 0.01s
Presolve: All rows and columns removed

Explored 0 nodes (0 simplex iterations) in 0.02 seconds
Thread count was 1 (of 8 available processors)

Solution count 1: 4

Optimal solution found (tolerance 1.00e-04)
Best objective 4.000000000000e+00, best bound 4.000000000000e+00, gap 0.0000%
s= [1.0, 0.0, 1.0, 1.0, 1.0]
```

Figure 1: Result from Gurobi

According to solution from Gurobi, we get that we should choose S_1, S_3, S_4, S_5 so that all elements are covered, the minimize number of set is 4.

2 Problem 2

For a Fractional Knapsack problem, the object is:

$$\max \sum_{i=1}^{n} v_i x_i$$

s.t.

$$\sum_{i=1}^{n} s_i x_i \leq B$$

$$0 \leq x_i \leq 1, \forall i$$

According to greedy algorithm, we need to create a new list $[\frac{v_1}{s_1} \geq \frac{v_2}{s_2} \cdots \geq \frac{v_i}{x_i}]$, and the solution is $x = [1, 1 \cdots 1, x_k, 0 \cdots, 0]$ such that $\sum_{i=1}^{k-1} s_i + x_k s_k = B$, then the value of object function is $\sum_{i=1}^{k-1} v_i + x_k v_k$.
Assume we can have a better solution by taking items with total $\triangle s$ from the greedy algorithm solutions and add items with total $\triangle s'$ to it. $\triangle s \geq \triangle s' \geq 0$ because $\sum_{i=1}^{k} s_i \leq B$
The value we take with $\triangle s$ is bounded by $\frac{v_k}{s_k} \triangle s \leq \triangle v \leq \frac{v_1}{s_1} \triangle s$
The value we can add with $\triangle s'$ is also bounded by $0 \leq \triangle v' \leq \frac{v_k}{s_k} \triangle s' \leq \frac{v_k}{s_k} \triangle s$
Hence the net value we can add $\triangle v' - \triangle v \leq \frac{v_k}{s_k} \triangle s - \frac{v_k}{s_k} \triangle s = 0$
Therefore, we can't add any value to the solution from greedy algorithm anymore, and it gives the optimal solution.

Python Code

```
import numpy as np
import pandas as pd
from gurobipy import *
import matplotlib.pyplot as plt

#
data = pd.DataFrame(pd.read_csv('D:\PANDA\Study\VG441\Homework\Problem Set 3\Data.csv'))
data = data.iloc[:, 1:]
m = data.shape[0]
n = data.shape[1]
#
X = []
t = []
for i in range(m): X.append(list(data.loc[i]))
WW = Model()
s = WW.addVars(m, vtype=GRB.BINARY, name="set_if_selected") # m sets
y = WW.addVars(n, name="element_if_covered") # n elements covered
WW.setObjective(quicksum(s), GRB.MINIMIZE)
for i in range(0, m):
    t.append(WW.addVars(n))
    WW.addConstrs(t[i][j] == s[i]*X[i][j] for j in range(n))
WW.addConstrs(y[j] == max_(t[i][j] for i in range(m)) for j in range(n))
WW.addConstrs(y[j] == 1 for j in range(n))
WW.optimize()
print('s=',WW.getAttr('X',s).values())
```