Problem 5

October 23, 2019

[1]:
```
# For the column player
from gurobipy import *

# Create new model
m = Model("problem5-1")

# Create variables (lowerbound of 0 by default)
y1 = m.addVar(vtype=GRB.CONTINUOUS, name="y1", lb=0)
y2 = m.addVar(vtype=GRB.CONTINUOUS, name="y2", lb=0)
y3 = m.addVar(vtype=GRB.CONTINUOUS, name="y3", lb=0)
y4 = m.addVar(vtype=GRB.CONTINUOUS, name="y4", lb=0)
v = m.addVar(vtype=GRB.CONTINUOUS, name="v", lb=0)

# Update the model
m.update()

# Set Objective
m.setObjective(v, GRB.MAXIMIZE)

#Add constraints
m.addConstr(2*y1-y2+3*y3-2*y4 >= v)
m.addConstr(y1+4*y2-3*y3 >= v)
m.addConstr(-2*y2-y3+3*y4 >= v)
m.addConstr(y1+y2+y3+y4 == 1)


# Optimize (model is updated when we optimize)
m.optimize()

#print model status (2 is optimal)
#https://www.gurobi.com/documentation/6.5/refman/optimization_status_codes.html
print ('Model status:', m.status)

#print decision variables
for v in m.getVars():
    print (v.varName, v.x)
```

```
#print objective function value
print ('Obj:', m.objVal)
```

```
Academic license - for non-commercial use only
Optimize a model with 4 rows, 5 columns and 17 nonzeros
Coefficient statistics:
  Matrix range     [1e+00, 4e+00]
  Objective range  [1e+00, 1e+00]
  Bounds range     [0e+00, 0e+00]
  RHS range        [1e+00, 1e+00]
Presolve removed 1 rows and 1 columns
Presolve time: 0.03s
Presolved: 3 rows, 4 columns, 12 nonzeros

Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    3.0000000e+00   1.375000e+00   0.000000e+00      0s
       2    7.9487179e-01   0.000000e+00   0.000000e+00      0s

Solved in 2 iterations and 0.04 seconds
Optimal objective  7.948717949e-01
Model status: 2
y1 0.6923076923076922
y2 0.02564102564102566
y3 0.0
y4 0.28205128205128205
v 0.7948717948717948
Obj: 0.7948717948717948
```

[2]:
```
# For the column player
from gurobipy import *

# Create new model
m = Model("problem5-2")

# Create variables (lowerbound of 0 by default)
x1 = m.addVar(vtype=GRB.CONTINUOUS, name="x1", lb=0)
x2 = m.addVar(vtype=GRB.CONTINUOUS, name="x2", lb=0)
x3 = m.addVar(vtype=GRB.CONTINUOUS, name="x3", lb=0)
u = m.addVar(vtype=GRB.CONTINUOUS, name="u")

# Update the model
m.update()

# Set Objective
m.setObjective(u, GRB.MINIMIZE)

#Add constraints
```

```
m.addConstr(2*x1+x2 <= u)
m.addConstr(-x1+4*x2-2*x3 <= u)
m.addConstr(3*x1-3*x2-x3 <= u)
m.addConstr(-2*x1+3*x3 <= u)
m.addConstr(x1+x2+x3 == 1)


# Optimize (model is updated when we optimize)
m.optimize()

#print model status (2 is optimal)
#https://www.gurobi.com/documentation/6.5/refman/optimization_status_codes.html
print ('Model status:', m.status)

#print decision variables
for v in m.getVars():
    print (v.varName, v.x)

#print objective function value
print ('Obj:', m.objVal)
```

```
Optimize a model with 5 rows, 4 columns and 17 nonzeros
Coefficient statistics:
  Matrix range     [1e+00, 4e+00]
  Objective range  [1e+00, 1e+00]
  Bounds range     [0e+00, 0e+00]
  RHS range        [1e+00, 1e+00]
Presolve time: 0.01s
Presolved: 5 rows, 4 columns, 17 nonzeros

Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    0.0000000e+00   1.500000e+00   0.000000e+00      0s
       3    7.9487179e-01   0.000000e+00   0.000000e+00      0s

Solved in 3 iterations and 0.02 seconds
Optimal objective  7.948717949e-01
Model status: 2
x1 0.17948717948717952
x2 0.4358974358974359
x3 0.3846153846153846
u 0.7948717948717949
Obj: 0.7948717948717949
```

[ ]: