IEOR 4004

Prof. Kroer

Due date: Oct 24, 2019

involving consultation with Yixuan Ma

HW#3

Zihui Zhou

1\. obj: max $2x_1 - x_2 + x_3$

cons: $3x_1 + x_2 + x_3 \geq 40$

$x_1 - x_2 + 2x_3 = 15$

$x_1 + x_2 - x_3 \leq 30$

$x_1 \geq 0$, $x_2 \leq 0$, x_3 unrestricted.

1) dual. obj: min $40y_1 + 15y_2 + 30y_3$

cons: $3y_1 + y_2 + y_3 \geq 2$

$y_1 - y_2 + y_3 \leq -1$

$y_1 + 2y_2 - y_3 = 1$

$y_1 \leq 0$, y_2 unrestricted, $y_3 \geq 0$

2) x $(25, 0, -5)^T$ $\Rightarrow$ $x_1 = 25$, $x_2 = 0$, $x_3 = -5$ is feasible for primal

- check binding: $3x_1 + x_2 + x_3 = 3 \times 25 - 5 = 70 \neq 40$ $\rightarrow$ $y_1 = 0$

$x_1 - x_2 + 2x_3 = 25 + 2 \times (-5) = 15$

$x_1 + x_2 - x_3 = 25 - (-5) = 30 = 30$

- since $x_1 = 25$, $x_3 = -5$, they don't equal to zero

then $3y_1 + y_2 + y_3 = 2$

$y_1 + 2y_2 - y_3 = 1$

a feasible primal solution x is an optimal solution to primal if and only if there exists y that is a feasible solution to the dual, and x and y are complementary.

solving the system $y_1 = 0$

$3y_1 + y_2 + y_3 = 2$

$y_1 + 2y_2 - y_3 = 1$

$\Rightarrow \begin{cases} y_1 = 0 \\ y_2 = 1 \\ y_3 = 1 \end{cases}$

$(0, 1, 1)^T$ is complementary to x, but it is not feasible

Since the 2nd constraint $y_1 - y_2 + y_3 = 0 - 1 + 1 = 0 \nleq -1$

Therefore, $(25, 0, -5)^T$ is not an optimal solution to primal

* $(40,-45,-35)^T \ni x_1=40,\ x_2=-45,\ x_3=-35$

- check constraints: $3x_1+x_2+x_3 = 3\times 40-45-35 = 40$

  $x_1-x_2+2x_3 = 40-(-45)+2(-35) = 15$

  $x_1+x_2-x_3 = 40-45-(-35) = 30 = 30$

- since $x_1, x_2, x_3 \neq 0$, then

$$\left.\begin{aligned} 3y_1+y_2+y_3 &= 2 \\ y_1-y_2+y_3 &= -1 \\ y_1+2y_2-y_3 &= 1 \end{aligned}\right. \Rightarrow \left\{\begin{aligned} y_1 &= -\tfrac{3}{2} \\ y_2 &= 3 \\ y_3 &= \tfrac{7}{2} \end{aligned}\right.$$

along with $y_j=0$.

Since $(-\frac{3}{2}, 3, \frac{7}{2})^T$ is feasible and complementary to x $(40,-45,-35)^T$ is optimal for the primal, $(-\frac{3}{2}, 3, \frac{7}{2})^T$ is optimal for dual

* $(13,0,1)^T \ni x_1=13,\ x_2=0,\ x_3=1$

- check constraints. $3x_1+x_2+x_3 = 13\times 3+1 = 40 = 40$.

  $x_1-x_2+2x_3 = 13-0+2 = 15 = 15$

  $x_1+x_2-x_3 = 13+0-1 = 12 \neq 30 \Rightarrow y_3=0$

- since $x_1, x_3 \neq 0$, then $3y_1+y_2+y_3=2$

  $y_1+2y_2-y_3=1$ (x_3 unrestricted, it always holds)

Solving the system
$$\left.\begin{aligned} 3y_1+y_2+y_3 &= 2 \\ y_1+2y_2-y_3 &= 1 \\ y_3 &= 0 \end{aligned}\right. \Rightarrow \left\{\begin{aligned} y_1 &= \tfrac{3}{5} \\ y_2 &= \tfrac{1}{5} \\ y_3 &= 0 \end{aligned}\right.$$

check $(\frac{3}{5}, \frac{1}{5}, 0)^T$ feasiblity: since $y_1=\frac{3}{5}>0$

it's not feasible Therefore, $(13,0,1)^T$ is not an optimal solution.

2. a) obj: max $3x_1+2x_2$

cons: $2x_1+5x_2 \le 8$

$3x_1+7x_2 \le 10$ $\Rightarrow$

$x_1, x_2 \ge 0$

dual obj: min $8y_1+10y_2$

cons: $2y_1+3y_2 \ge 3$

$5y_1+7y_2 \ge 2$

$y_1, y_2 \ge 0$

Then, solving for the optimal solution of the dual of the LP

Reading from the optimal tableau: $x_1=\frac{10}{3}$, $S_1=\frac{4}{3}$, $x_2=S_2=0$.

Therefore, * dual constraints: $2x_1+5x_2 = 2\times\frac{10}{3} \ne 8 \Rightarrow y_1=0$

$3x_1+7x_2 = 3\times\frac{10}{3} = 10 = 10$

* since $x_1 \ne 0 \Rightarrow 2y_1+3y_2=3$

Solving the system $\begin{cases} y_1=0 \\ 2y_1+3y_2=3 \end{cases} \Rightarrow \begin{cases} y_1=0 \\ y_2=1 \end{cases}$

since $(0,1)^T$ is feasible for the dual $\Rightarrow$ $(0,1)^T$ is dual's optimal solution

b) current object of the optimal tableau is $z+5x_2+S_2=10$

$\frac{1}{3}x_2+S_1-\frac{2}{3}S_2=\frac{4}{3}$

$x_1+\frac{7}{3}x_2 \quad +\frac{1}{3}S_2=\frac{10}{3}$

* basis: $x_1=-\frac{7}{3}x_2-\frac{1}{3}S_2+\frac{10}{3}$

$S_1=-\frac{1}{3}x_2+\frac{2}{3}S_2+\frac{4}{3}$

let new $b_2=10+\Delta$, then the system transforms to:

max $z=3x_1+2x_2$

$2x_1+5x_2+S_1 \quad =8$

$3x_1+7x_2 \quad +S_2=10+\Delta \Rightarrow 3x_1+7x_2 \quad +(S_2-\Delta)=10$

$\Rightarrow 3x_1+7x_2 \quad +S_2' = 10.$

Then, basis $x_1=-\frac{7}{3}x_2-\frac{1}{3}S_2'+\frac{10}{3} = -\frac{7}{3}x_2-\frac{1}{3}(S_2-\Delta)+\frac{10}{3} = -\frac{7}{3}x_2-\frac{1}{3}S_2+\frac{1}{3}\Delta+\frac{10}{3}$

$S_1=-\frac{1}{3}x_2+\frac{2}{3}S_2'+\frac{4}{3} = -\frac{1}{3}x_2+\frac{2}{3}(S_2-\Delta)+\frac{4}{3} = -\frac{1}{3}x_2+\frac{2}{3}S_2-\frac{2}{3}\Delta+\frac{4}{3}$

Since the dictionary is optimal when it's feasible

as $z = -5x_2 - s_2 + 10$. Then

$$\begin{cases} \frac{1}{3}\Delta + \frac{10}{3} \geq 0 \\ -\frac{2}{3}\Delta + \frac{4}{3} \geq 0 \end{cases} \quad \text{gives } -10 \leq \Delta \leq 2.$$

注意Δ可以为负!!!

Therefore when $0 \leq b_2 = 10 + \Delta \leq 12$, the current basis remains optimal

* when we change $b_2 = 5$, $\Delta = 5 - 10 = -5$

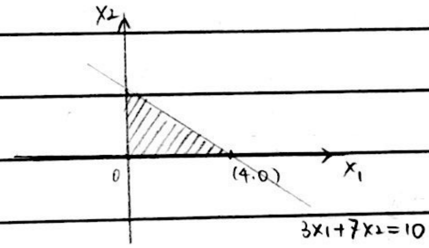
$x_1 = -\frac{7}{3}x_2 - \frac{1}{3}s_2 + \frac{1}{3}\Delta + \frac{10}{3} = -\frac{7}{3}x_2 - \frac{1}{3}s_2 + \frac{1}{3}\times(-5) + \frac{10}{3} \Rightarrow x_1 = \frac{5}{3}$

$s_1 = -\frac{1}{3}x_2 + \frac{2}{3}s_2 - \frac{2}{3}\Delta + \frac{4}{3} = -\frac{1}{3}x_2 + \frac{2}{3}s_2 - \frac{2}{3}(-5) + \frac{4}{3} \Rightarrow s_1 = \frac{14}{3}$

with $z = 10 - 5x_2 - s_2' = 10 - (s_2 - \Delta) = 10 - s_2 + \Delta = 10 - 5 = 5$.

optimal solution: $(\frac{5}{3}, 0)^T$

c) graphing the system on the plane. we can notice that it always has a feasible region. Therefore, no matter how we changes the object of the given (primal LP), it always would realize optimal.



~~Sin~~ Therefore, the system could not be infeasible. Since dual will be infeasible if and only if the primal is. The dual couldn't become infeasible in this case.

primal

3. a. obj: max $z = 4x_1 + x_2 + 2x_3$

cons: $8x_1 + 3x_2 + x_3 \le 2$

$6x_1 + x_2 + x_3 \le 8$ $\Rightarrow$

$x_1, x_2, x_3 \ge 0$

dual

obj: min $2y_1 + 8y_2$

$8y_1 + 6y_2 \ge 4$

$3y_1 + y_2 \ge 1$

$y_1 + y_2 \ge 2$

$y_1, y_2 \ge 0$

The optimal tableau gives $z + 12x_1 + 5x_2 + 2s_1 = 4$

$8x_1 + 3x_2 + x_3 + s_1 = 2$

$-2x_1 - 2x_2 - s_1 + s_2 = 6$

with basis $x_3 = 2$, $s_2 = 6$, $x_1 = x_2 = s_1 = 0$

* check constraints: $8x_1 + 3x_2 + x_3 = 0 + 0 + 2 = 2$

$6x_1 + x_2 + x_3 = 0 + 0 + 2 < 8 \rightarrow y_2 = 0$

* Since $x_3 = 2 > 0$, then $y_1 + y_2 = 2$

Solving the system $\begin{cases} y_1 + y_2 = 2 \\ y_2 = 0 \end{cases} \Rightarrow \begin{cases} y_1 = 2 \\ y_2 = 0 \end{cases}$

check feasibility: $(2, 0)^T$ is a feasible solution

and complementary to x. Therefore, $(2, 0)^T$ is an optimal solution

b. let $c_3 = 2 + \Delta$

$z = 4x_1 + x_2 + (2+\Delta)x_3$

from the optimal tableau $x_3 = 2 - 8x_1 - 3x_2 - s_1$

then $z' = 4x_1 + x_2 + (2+\Delta) \times (2 - 8x_1 - 3x_2 - s_1)$

$= 4x_1 + x_2 + 4 - 16x_1 - 6x_2 - 2s_1 + 2\Delta - 8\Delta x_1 - 3\Delta x_2 - \Delta s_1$

$= (-12 - 8\Delta)x_1 + (-5 - 3\Delta)x_2 + (-2 - \Delta)s_1 + (4 + 2\Delta)$

for z' to remain optimal, let $-12 - 8\Delta \le 0$

$-5 - 3\Delta \le 0 \Rightarrow \Delta \ge -1.5$

$-2 - \Delta \le 0$

Therefore $2 + \Delta = c_3 \ge \frac{1}{2}$ for the current basis remains optimal

c. let $c_1 = 4 + \Delta'$

since the objective $= (4+\Delta')x_1 + x_2 + 2x_3$

$$= (4+\Delta')x_1 + x_2 + 2(2 - 8x_1 - 3x_2 - s_1)$$

$$= (-12+\Delta')x_1 - 5x_2 - 2s_1 + 4$$

$-12 + \Delta' \leq 0 \Rightarrow \Delta' \leq 12 \Rightarrow c_1 = 4 + \Delta' \leq 16$

Therefore, the coefficient of x_1 should be at most 16 for which the current basis remains optimal.

4. a.

$$\begin{array}{cc|ccc} A & & & \text{cp} & \\ & & P & S & R \\ \hline & P & 0 & 1 & -1 \\ \text{rp} & S & -1 & 0 & 1 \\ & R & 1 & -1 & 0 \end{array}$$

rp choose the best strategy $\vec{x}$

cp chooses the best strategy $\vec{y}$

* for the column player:

$$\max_{y} \min_{i \in \{R,P,S\}} [Ay]_i$$

$$y_P + y_R + y_S = 1$$

$$y_P, y_R, y_S \geq 0$$

$\Rightarrow$

$$\max v$$

$$[Ay]_R \geq v$$

$$[Ay]_S \geq v$$

$$[Ay]_P \geq v$$

$$y_P + y_R + y_S = 1$$

$$y_P, y_R, y_S \geq 0$$

* for the row player

$$\min_{x} \max_{i \in \{R,P,S\}} [x^T A]_i$$

$$x_P + x_R + x_S = 1$$

$$x_P, x_R, x_S \geq 0$$

$\Rightarrow$

$$\min u$$

$$[x^T A]_R \leq u$$

$$[x^T A]_S \leq u$$

$$[x^T A]_P \leq u$$

$$x_P + x_R + x_S = 1$$

$$x_P, x_R, x_S \geq 0$$

Using Gurobi to solve the two systems gives us

the optimal solution for row player is $(x_P, x_S, x_R) = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$

the optimal solution for column player is $(y_P, y_S, y_R) = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$

b. $(x_R, x_S, x_P) = (\frac{2}{5}, \frac{1}{10}, \frac{1}{2})$

*? 如何体现 min?

from A: object is $\max \frac{2}{5}(y_P - y_S) + \frac{1}{10}(y_R - y_P) + \frac{1}{2}(y_S - y_R)$

$= \frac{3}{10} y_P + \frac{1}{10} y_S - \frac{3}{5} y_R$

constraint: $y_P + y_S - y_R = 1$

$y_P, y_S, y_R \geq 0$

Therefore, set $y_R = 0$, $y_S = 0$, $y_P = 1$ will maximize the object.

$\Rightarrow (y_R, y_S, y_P) = (0, 0, 1)$ with the optimal objective as 3/10 = 0.3

5.

$$A = \begin{array}{c} \\ \\ \\ rp: x \end{array} \begin{array}{c} cp\ \vec{y} \\ \begin{pmatrix} 2 & -1 & 3 & -2 \\ 1 & 4 & -3 & 0 \\ 0 & -2 & -1 & 3 \end{pmatrix} \end{array}$$

for A_{ij}, it's the gain of the column player when he plays j and the row player plays i.

zero-sum for matrix 不用 $\sum_i \sum_j A_{ij} = 0$?

* for the column player

$\max_{y} \min_{i \in X} [Ay]_i \Rightarrow \max_{y} v$

$2y_1 - y_2 + 3y_3 - 2y_4 \geq v$

$y_1 + 4y_2 - 3y_3 \geq v$

$-2y_2 - y_3 + 3y_4 \geq v$

$y_1, y_2, y_3, y_4 \geq 0$

$y_1 + y_2 + y_3 + y_4 = 1$

Using Gurobi to solve for the solution:

$(y_1, y_2, y_3, y_4) = (0.6923, 0.02564, 0, 0.2821)$

with objective = 0.79487

* for the row player:

$$\min_x \max_y [Ay] \Rightarrow \min u$$

$$2x_1 + x_2 \le u$$

$$-x_1 + 4x_2 - 2x_3 \le u$$

$$3x_1 - 3x_2 - x_3 \le u$$

$$-2x_1 + 3x_3 \le u$$

$$x_1, x_2, x_3 \ge 0$$

$$x_1 + x_2 + x_3 = 1$$

Using Gurobi to solve for the system $(x_1, x_2, x_3) = (0.17948, 0.43589, 0.3846)$ with $u = v = 0.79487$

b. initialization: set $d_s = 0$ and $d_u = +\infty$ for every node u. arcs $A = a_1, a_2 \ldots a_m$

main body of algorithm:

* for each $(u,v) \in$ arcs A ($= a_1, a_2 \ldots a_m$), if $d_v > d_u + c(u,v)$ then we reset $d_v = d_u + c(u,v)$ and $pred_v = u$.
* repeats the above step $|V|-1$ times
* repeat the first step one more time

  if there exists d_v such that $d_v > d_u + c(u,v)$, then, there exists a cycle of negative cost.

bounded by $|V| \times \frac{|V| \times (|V|-1)}{2} = O(|V|^3)$ times

7. if there exists a cycle such that the product of the weight of edges $c(S,V_1), c(V_1,V_2) \ldots c(V_n,S)$ is greater than 1 we may trade the currency starting with \$100 and ending up with more than \$100. (with $\$100 \times c(S,V_1) \times \ldots \times c(V_n,S)$)

Since $\log(ab) = \log a + \log b$, we replace the cost of each arc, say $c(S,V_1)$ with $\log(c(S,V_1))$ & then for a path $P = a_1 \ldots a_k$ we have

* $|V|-1$ scans.

$$\log\left(\prod_{i=1}^{k} c(a_i)\right) = \sum_{i=1}^{k} \log(c(a_i)) \quad a_i \in A \text{ arcs.}$$

Then, let's change the weight $c(a_i)$, $a_i \in A$ to negative in the graph. Using the algorithm developed in problem 6, if there exists a cycle of negative cost, which, let's say $-\log(c(a_1)) - \log(c(a_2)) - \cdots - \log(c(a_n)) \leq 0 = -\log 1$

$$-[\log(c(a_1) \cdot c(a_2) \cdots c(a_n))] \leq -\log 1$$

$$[\log(c(a_1) \cdot c(a_2) \cdots c(a_n))] \geq \log 1$$

$$c(a_1) \cdot c(a_2) \cdots c(a_n) \geq 1$$

Therefore, the ~~wei~~ product of the weight is greater than 1 which will return more than \$100 currency

Problem 4

October 23, 2019

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

# Create new model
m = Model("problem4-1")

# Create variables (lowerbound of 0 by default)
y2 = m.addVar(vtype=GRB.CONTINUOUS, name="y2", lb=0)
y3 = m.addVar(vtype=GRB.CONTINUOUS, name="y3", 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((y2-y3) >= v)
m.addConstr((y3-(1-y2-y3)) >= v)
m.addConstr(((1-y2-y3)-y2) >= v)

# 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)
```

```
  Bounds range     [0e+00, 0e+00]
```

```
  RHS range        [1e+00, 1e+00]
Presolve removed 1 rows and 1 columns
Presolve time: 0.01s
Presolved: 2 rows, 2 columns, 4 nonzeros

Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    1.3333333e-03   4.333333e-03   0.000000e+00      0s
       1   -0.0000000e+00   0.000000e+00   0.000000e+00      0s

Solved in 1 iterations and 0.02 seconds
Optimal objective -0.000000000e+00
Model status: 2
y2 0.3333333333333333
y3 0.3333333333333333
v 0.0
Obj: -0.0
```

[2]:

```
# For the row player
from gurobipy import *

# Create new model
m = Model("problem4-2")

# Create variables (lowerbound of 0 by default)
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", lb=0)

# Update the model
m.update()

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

#Add constraints
m.addConstr(-x2+x3 <= u)
m.addConstr((1-x2-x3)-x3 <= u)
m.addConstr(-(1-x2-x3)+x2 <= u)

# 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 3 rows, 3 columns and 9 nonzeros
Coefficient statistics:
  Matrix range     [1e+00, 2e+00]
  Objective range  [1e+00, 1e+00]
  Bounds range     [0e+00, 0e+00]
  RHS range        [1e+00, 1e+00]
Presolve time: 0.01s
Presolved: 3 rows, 3 columns, 9 nonzeros

Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    0.0000000e+00   5.000000e-01   0.000000e+00      0s
       2    0.0000000e+00   0.000000e+00   0.000000e+00      0s

Solved in 2 iterations and 0.02 seconds
Optimal objective  0.000000000e+00
Model status: 2
x2 0.3333333333333333
x3 0.33333333333333337
u 0.0
Obj: 0.0
```

[ ]:

```

```

Problem 5

October 23, 2019

[1]:
```python
# 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
```

[ ]: