Karen Loscocco

ISYE 4133 Assignment 4

1 (a)

x_j: starting time of job j
p_j: process time of job j
r_j: release time of job j
d_j: due date of job j
w_j: weight of job j
$j = \{1, 2, 3\}$
Set M $= -(\sum_j p_j + r_j)$

$$\min \sum_j w_j(x_j + p_j)$$
$$x_j \geq r_j \forall j$$
$$x_j + p_j \leq d_j \forall j$$
$$x_i \geq x_j + p_j + M(1 - y_{ij}) \forall i < j$$
$$x_j \geq x_i + p_i + M(1 - y_{ji}) \forall i < j$$
$$y_{ij} + y_{ji} \geq 1 \forall i < j$$
$$y_{ij}, y_{ji} \in \{0, 1\}$$

In [2]:
```
p = np.array([15,6,9])
r = np.array([5,10,0])
d = np.array([20,27,38])
w = np.array([6,10,40])
M = -np.sum(p+r)

m = Model()
X = m.addVars(3, name = 'x')
y = m.addVars(list(permutations([0,1,2],2)),vtype = GRB.BINARY, name =
'y')

m.addConstrs(X[j]>=r[j] for j in range(len(r)))

m.addConstrs(X[j]+p[j] <= d[j] for j in range(len(d)))

m.addConstrs((X[i] >= X[j] + p[j] + M*(1-y[i,j]) for i in X for j in X
if i < j))

m.addConstrs((X[j] >= X[i] + p[i] + M*(1-y[j,i]) for j in X for i in X
if i < j))

m.addConstrs((y[i,j] + y[j,i] >= 1 for i in X for j in X if i < j))

m.setObjective(np.dot(w,np.add(p,X.values())), GRB.MINIMIZE)

m.optimize()

printOptimal(m)
```

```
Academic license - for non-commercial use only
Optimize a model with 15 rows, 9 columns and 30 nonzeros
Variable types: 3 continuous, 6 integer (6 binary)
Coefficient statistics:
  Matrix range     [1e+00, 4e+01]
  Objective range  [6e+00, 4e+01]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 4e+01]
Presolve removed 15 rows and 9 columns
Presolve time: 0.00s
Presolve: All rows and columns removed

Explored 0 nodes (0 simplex iterations) in 0.01 seconds
Thread count was 1 (of 8 available processors)

Solution count 1: 1780

Optimal solution found (tolerance 1.00e-04)
Best objective 1.780000000000e+03, best bound 1.780000000000e+03, ga
p 0.0000%


Variable Values:
x[0] 5.0
x[1] 20.0
x[2] 26.0
y[0,1] 0.0
y[0,2] 0.0
y[1,0] 1.0
y[1,2] 0.0
y[2,0] 1.0
y[2,1] 1.0

Objective Value: 1780.0
```

1 (b)

x_j: starting time of job j
p_j: process time of job j
r_j: release time of job j
d_j: due date of job j
w_j: weight of job j
$j = \{1, 2, 3\}$
Set M $= -(\sum_j p_j + r_j)$

$$\min z$$
$$x_j \geq r_j \forall j$$
$$x_j + p_j \leq d_j \forall j$$
$$x_i \geq x_j + p_j + M(1 - y_{ij}) \forall i < j$$
$$x_j \geq x_i + p_i + M(1 - y_{ji}) \forall i < j$$
$$y_{ij} + y_{ji} \geq 1 \forall i < j$$
$$x_j + p_j \leq z \forall j$$
$$y_{ij}, y_{ji} \in \{0, 1\}$$

In [3]:
```
p = [15,6,9]
r = [5,10,0]
d = [20,27,38]
w = [6,10,40]
M = -np.sum(p+r)

m = Model()
X = m.addVars(3, name = 'x')
y = m.addVars(list(permutations([0,1,2],2)),vtype = GRB.BINARY, name =
'y')
z = m.addVar(name = 'z')

m.addConstrs(X[j]>=r[j] for j in range(len(r)))

m.addConstrs(X[j]+p[j] <= d[j] for j in range(len(d)))

m.addConstrs((X[i] >= X[j] + p[j] + M*(1-y[i,j]) for i in X for j in X
if i < j))

m.addConstrs((X[j] >= X[i] + p[i] + M*(1-y[j,i]) for j in X for i in X
if i < j))

m.addConstrs((y[i,j] + y[j,i] >= 1 for i in X for j in X if i<  j))

m.addConstrs((X[j] + p[j] <= z for j in range(3)))

m.setObjective(z,GRB.MINIMIZE)

m.optimize()

printOptimal(m)
```

```
Optimize a model with 18 rows, 10 columns and 36 nonzeros
Variable types: 4 continuous, 6 integer (6 binary)
Coefficient statistics:
  Matrix range     [1e+00, 4e+01]
  Objective range  [1e+00, 1e+00]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 4e+01]
Presolve removed 18 rows and 10 columns
Presolve time: 0.00s
Presolve: All rows and columns removed

Explored 0 nodes (0 simplex iterations) in 0.01 seconds
Thread count was 1 (of 8 available processors)

Solution count 1: 35

Optimal solution found (tolerance 1.00e-04)
Best objective 3.500000000000e+01, best bound 3.500000000000e+01, ga
p 0.0000%


Variable Values:
x[0] 5.0
x[1] 20.0
x[2] 26.0
y[0,1] 0.0
y[0,2] 0.0
y[1,0] 1.0
y[1,2] 0.0
y[2,0] 1.0
y[2,1] 1.0
z 35.0

Objective Value: 35.0
```

1 (c)

x_j: starting time of job j
p_j: process time of job j
r_j: release time of job j
d_j: due date of job j
w_j: weight of job j
t_j: tardiness for job j
$j = \{1, 2, 3\}$
Set M $= -(\sum_j p_j + r_j)$

$$\min \sum_j w_j(x_j + p_j)$$
$$x_j \geq r_j \forall j$$
$$t_j \geq (x_j + p_j - d_j)$$
$$t_j \geq 0$$
$$\sum_j t_j \leq 100$$
$$x_i \geq x_j + p_j + M(1 - y_{ij}) \forall i < j$$
$$x_j \geq x_i + p_i + M(1 - y_{ji}) \forall i < j$$
$$y_{ij} + y_{ji} \geq 1 \forall i < j$$
$$y_{ij}, y_{ji} \in \{0, 1\}$$

In [4]:
```
p = np.array([15,6,9])
r = np.array([5,10,0])
d = np.array([20,27,38])
w = np.array([6,10,40])
M = -np.sum(p+r)

m = Model()
X = m.addVars(3, name = 'x')
y = m.addVars(list(permutations([0,1,2],2)),vtype = GRB.BINARY, name =
'y')
t = m.addVars(3, name = 't')

m.addConstrs(X[j]>=r[j] for j in range(len(r)))

m.addConstrs(X[j] + p[j] - d[j] <= t[j] for j in range(len(d)))

m.addConstr(sum(t.values()) <= 100)

m.addConstrs((X[i] >= X[j] + p[j] + M*(1-y[i,j]) for i in X for j in X
if i < j))

m.addConstrs((X[j] >= X[i] + p[i] + M*(1-y[j,i]) for j in X for i in X
if i < j))

m.addConstrs((y[i,j] + y[j,i] >= 1 for i in X for j in X if i < j))

m.setObjective(np.dot(w,np.add(p,X.values())), GRB.MINIMIZE)

m.optimize()

printOptimal(m)
```

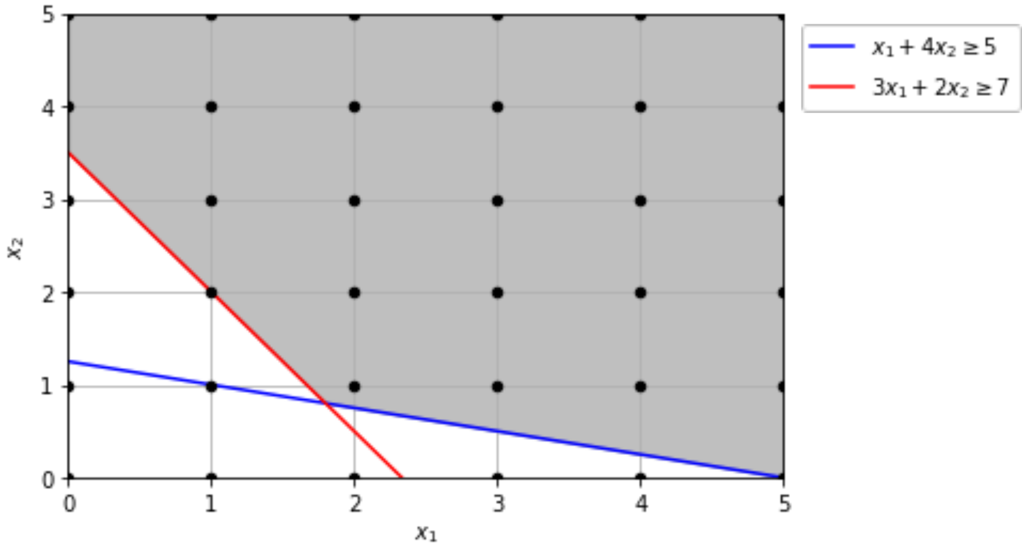
```
Optimize a model with 16 rows, 12 columns and 36 nonzeros
Variable types: 6 continuous, 6 integer (6 binary)
Coefficient statistics:
  Matrix range     [1e+00, 4e+01]
  Objective range  [6e+00, 4e+01]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 1e+02]
Presolve removed 7 rows and 4 columns
Presolve time: 0.00s
Presolved: 9 rows, 8 columns, 25 nonzeros
Variable types: 5 continuous, 3 integer (3 binary)

Root relaxation: objective 6.400000e+02, 2 iterations, 0.00 seconds

    Nodes    |    Current Node    |     Objective Bounds      |
Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/N
```

```
ode Time

     0     0  640.00000    0    2          -  640.00000      -     -
0s
H    0     0                    706.0000000  640.00000  9.35%     -
0s
     0     0     cutoff    0       706.00000  706.00000  0.00%     -
0s

Explored 1 nodes (2 simplex iterations) in 0.01 seconds
Thread count was 8 (of 8 available processors)

Solution count 1: 706

Optimal solution found (tolerance 1.00e-04)
Best objective 7.060000000000e+02, best bound 7.060000000000e+02, ga
p 0.0000%


Variable Values:
x[0] 16.0
x[1] 10.0
x[2] 0.0
y[0,1] 1.0
y[0,2] 1.0
y[1,0] 0.0
y[1,2] 1.0
y[2,0] -0.0
y[2,1] -0.0
t[0] 11.0
t[1] 89.0
t[2] 0.0

Objective Value: 706.0
```

2

$$\begin{gathered} \min 4x_1 + 5x_2 \\ x_1 + 4x_2 \geq 5 \\ 3x_1 + 2x_2 \geq 7 \\ x_1, x_2 \in \mathbb{Z}_+ \end{gathered}$$

The feasible region is the set of integral points within the shaded region.



Put the IP in standard form, relax the integer constraints, and solve over the larger region using Gurobi:

$$\min 4x_1 + 5x_2$$
$$x_1 + 4x_2 - x_3 = 5$$
$$3x_1 + 2x_2 - x_4 = 7$$
$$x_1, x_2, x_3, x_4 \geq 0$$

```
Variable Values:
X[1] 1.8
X[2] 0.8
X[3] 0.0
X[4] 0.0

Objective Value: 11.2
```

LP in Canonical Form for $B = \{1, 2\}$:

$$\min \begin{bmatrix} 0 & 0 & \frac{7}{10} & \frac{11}{10} \end{bmatrix} x + \frac{56}{5}$$

$$\begin{bmatrix} 1 & 0 & \frac{1}{5} & -\frac{2}{5} \\ 0 & 1 & -\frac{3}{10} & \frac{1}{10} \end{bmatrix} x = \begin{bmatrix} \frac{9}{5} \\ \frac{4}{5} \end{bmatrix}$$

$$x \geq 0$$

Solution is not integral, so pick x_1 and take the floor:

$x_1 + \lfloor \frac{1}{5} \rfloor x_3 + \lfloor -\frac{2}{5} \rfloor x_4 \leq \frac{9}{5}$

Given that x_i 's are integers:

$x_1 + \lfloor \frac{1}{5} \rfloor x_3 + \lfloor -\frac{2}{5} \rfloor x_4 \leq \lfloor \frac{9}{5} \rfloor$

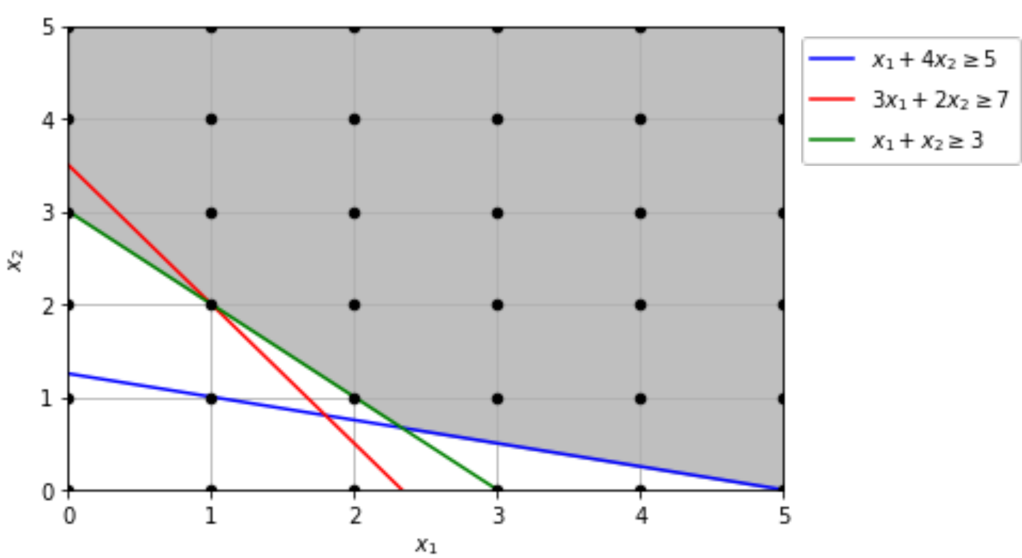
$x_1 - x_4 \leq 1$

$x_1 - (3x_1 + 2x_2 - 7) \leq 1 \implies x_1 + x_2 \geq 3$

Add the above constraint to the original LP:

$$\begin{gathered} \min 4x_1 + 5x_2 \\ x_1 + 4x_2 \geq 5 \\ 3x_1 + 2x_2 \geq 7 \\ x_1 + x_2 \geq 3 \\ x_1, x_2 \in \mathbb{Z}_+ \end{gathered}$$

The new feasible region is the set of integral points within the shaded region.



Put the IP in standard form, relax the integer constraints, and solve over the larger region using Gurobi:

$$\begin{gathered}\min 4x_1 + 5x_2\\ x_1 + 4x_2 - x_3 = 5\\ 3x_1 + 2x_2 - x_4 = 7\\ x_1 + x_2 - x_5 = 3\\ x_1, x_2, x_3, x_4, x_5 \geq 0\end{gathered}$$

```
Variable Values:
X[1] 2.3333333333333335
X[2] 0.6666666666666665
X[3] 0.0
X[4] 1.333333333333333
X[5] 0.0

Objective Value: 12.666666666666666
```

LP in Canonical Form for $B = \{1, 2, 4\}$:

min $\begin{bmatrix} 0 & 0 & \frac{1}{3} & 0 & \frac{11}{3} \end{bmatrix} x + \dfrac{38}{3}$

$\begin{bmatrix} 1 & 0 & \frac{1}{3} & 0 & -\frac{4}{3} \\ 0 & 1 & -\frac{1}{3} & 0 & \frac{1}{3} \\ 0 & 0 & \frac{1}{3} & 1 & -\frac{10}{3} \end{bmatrix} x = \begin{bmatrix} \frac{7}{3} \\ \frac{2}{3} \\ \frac{4}{3} \end{bmatrix}$

$x \geq 0$

Solution is not integral, so pick x_2 and take the floor:

$$x_2 + \lfloor -\frac{1}{3} \rfloor x_3 + \lfloor \frac{1}{3} \rfloor x_5 \leq \frac{2}{3}$$

Given that x_i 's are integers:

$$x_2 + \lfloor -\frac{1}{3} \rfloor x_3 + \lfloor \frac{1}{3} \rfloor x_5 \leq \lfloor \frac{2}{3} \rfloor$$

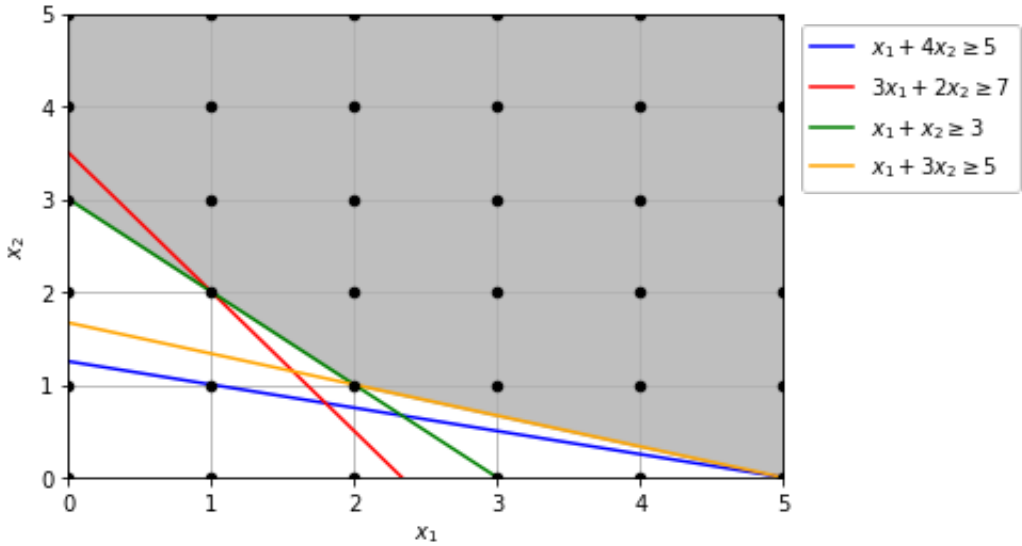
$$x_2 - x_3 \leq 0$$

$$x_2 - (x_1 + 4x_2 - 5) \leq 0 \implies x_1 + 3x_2 \geq 5$$

Add the above constraint to the original LP:

$$\begin{gathered} \min 4x_1 + 5x_2 \\ x_1 + 4x_2 \geq 5 \\ 3x_1 + 2x_2 \geq 7 \\ x_1 + x_2 \geq 3 \\ x_1 + 3x_2 \geq 5 \\ x_1, x_2 \in \mathbb{Z}_+ \end{gathered}$$

The new feasible region is the set of integral points within the shaded region.



Put the IP in standard form, relax the integer constraints, and solve over the larger region using Gurobi:

$$\min 4x_1 + 5x_2$$
$$x_1 + 4x_2 - x_3 = 5$$
$$3x_1 + 2x_2 - x_4 = 7$$
$$x_1 + x_2 - x_5 = 3$$
$$x_1 + 3x_2 - x_6 = 5$$
$$x_1, x_2, x_3, x_4, x_5, x_6 \geq 0$$

```
Variable Values:
X[1] 2.0
X[2] 1.0
X[3] 1.0
X[4] 1.0
X[5] 0.0
X[6] 0.0

Objective Value: 13.0
```

LP in Canonical Form for $B = \{1, 2, 3, 4\}$:

min $\begin{bmatrix} 0 & 0 & 0 & 0 & -\frac{7}{2} & \frac{1}{2} \end{bmatrix} x + 13$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & \frac{3}{2} & \frac{1}{2} \\ 0 & 1 & 0 & 0 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & 0 & 1 & 0 & -\frac{1}{2} & -\frac{3}{2} \\ 0 & 0 & 0 & 1 & \frac{7}{2} & \frac{1}{2} \end{bmatrix} x = \begin{bmatrix} 2 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$x \geq 0$

The solution is integral and is also the solution to the original LP.

$X = (2, 1)$ and the objective value is 13.

Verification in Gurobi:

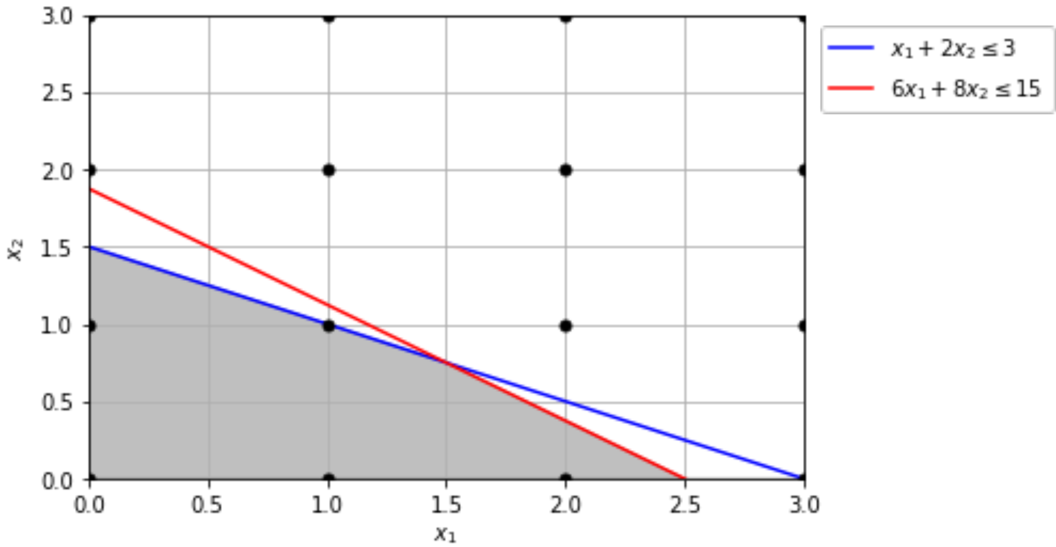
```
Optimize a model with 2 rows, 2 columns and 4 nonzeros
Variable types: 0 continuous, 2 integer (0 binary)
Coefficient statistics:
  Matrix range     [1e+00, 4e+00]
  Objective range  [4e+00, 5e+00]
  Bounds range     [0e+00, 0e+00]
  RHS range        [5e+00, 7e+00]
Found heuristic solution: objective 20.0000000
Presolve time: 0.00s
Presolved: 2 rows, 2 columns, 4 nonzeros
Variable types: 0 continuous, 2 integer (0 binary)

Root relaxation: objective 1.166667e+01, 1 iterations, 0.00 seconds

    Nodes    |    Current Node    |     Objective Bounds      |
Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/N
ode Time

     0     0   11.66667    0    1   20.00000   11.66667  41.7%     -
0s
H    0     0                     13.0000000   11.66667  10.3%     -
0s
     0     0 infeasible    0        13.00000   13.00000  0.00%     -
0s

Explored 1 nodes (1 simplex iterations) in 0.03 seconds
Thread count was 8 (of 8 available processors)

Solution count 2: 13 20

Optimal solution found (tolerance 1.00e-04)
Best objective 1.300000000000e+01, best bound 1.300000000000e+01, ga
p 0.0000%


Variable Values:
X[1] 2.0
X[2] 1.0

Objective Value: 13.0
```

3

Original IP: F

$$\max\ 2x_1 + 3x_2$$
$$x_1 + 2x_2 \leq 3$$
$$6x_1 + 8x_2 \leq 15$$
$$x_1, x_2 \in \mathbb{Z}_+$$

The feasible region is the set of integral points within the shaded region.



Put the IP in standard form, relax the integer constraints, and solve over the larger region using Gurobi:

$$\max\ 2x_1 + 3x_2$$
$$x_1 + 2x_2 + x_3 = 3$$
$$6x_1 + 8x_2 + x_4 = 15$$
$$x_1, x_2, x_3, x_4 \geq 0$$

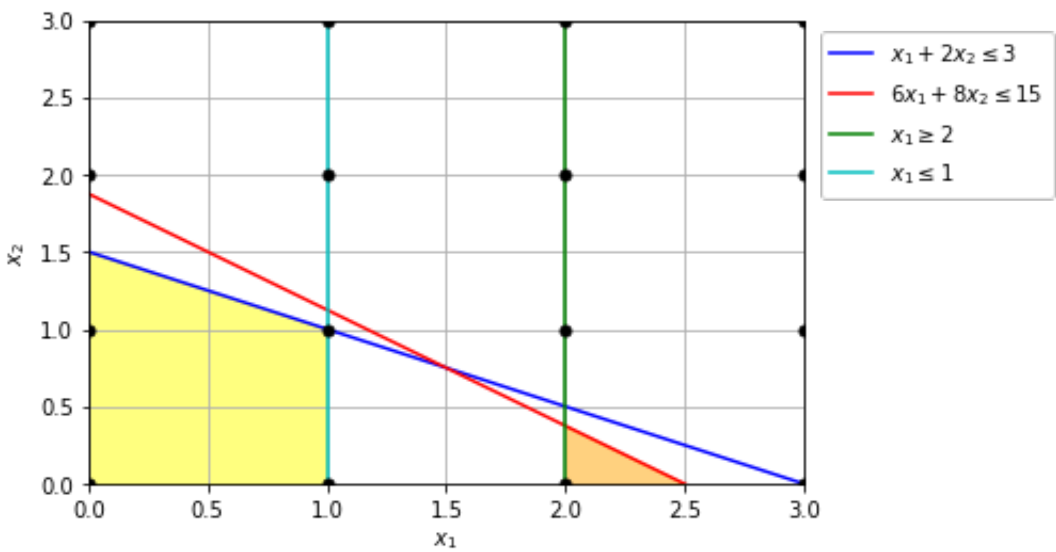
```
Variable Values:
X[1] 1.5
X[2] 0.75
X[3] 0.0
X[4] 0.0

Objective Value: 5.25
```

Set the upper bound or best known feasible solution to 5.25.
LP optimal solution: $X = (1.5, 0.75)$
Branch for x_1:

$$\begin{array}{rr} F_1 & F_2 \\ \max\ 2x_1 + 3x_2 & \max\ 2x_1 + 3x_2 \\ x_1 + 2x_2 \leq 3 & x_1 + 2x_2 \leq 3 \\ 6x_1 + 8x_2 \leq 15 & 6x_1 + 8x_2 \leq 15 \\ x_1 \leq 1 & x_1 \geq 2 \\ x_1, x_2 \in \mathbb{Z}_+ & x_1, x_2 \in \mathbb{Z}_+ \end{array}$$

The feasible region for F_1 is the set of integral points within the yellow shaded region.

The feasible region for F_2 is the set of integral points within the orange shaded region.



LP relaxation in standard form:

$$
\begin{array}{rr}
F_1 \text{ linear} & F_2 \text{ linear} \\
\max 2x_1 + 3x_2 & \max 2x_1 + 3x_2 \\
x_1 + 2x_2 + x_3 = 3 & x_1 + 2x_2 + x_3 = 3 \\
6x_1 + 8x_2 + x_4 = 15 & 6x_1 + 8x_2 + x_4 = 15 \\
x_1 + x_5 = 1 & x_1 - x_5 = 2 \\
x_1, x_2, x_3, x_4, x_5 \geq 0 & x_1, x_2, x_3, x_4, x_5 \geq 0
\end{array}
$$

Solve F_1 LP with Gurobi:

```
Variable Values:
X[1] 1.0
X[2] 1.0
X[3] 0.0
X[4] 1.0
X[5] 0.0

Objective Value: 5.0
```

Optimal solution for $F_1: X = (1, 1)$
Can update the upper bound to: 5

Solve F_2 LP with Gurobi:

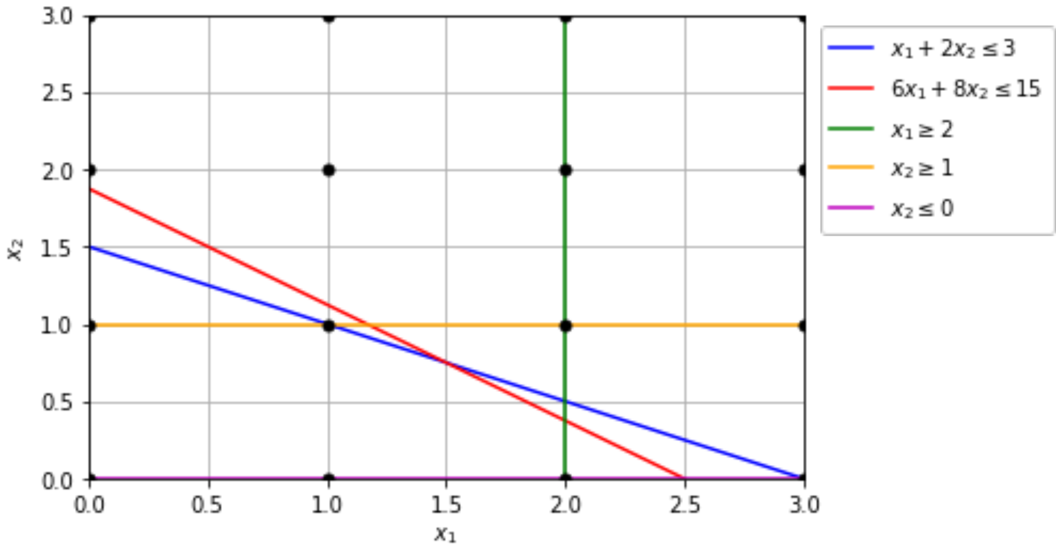
```
Variable Values:
X[1] 2.0
X[2] 0.375
X[3] 0.25
X[4] 0.0
X[5] 0.0

Objective Value: 5.125
```

Need to branch again off of F_2: $X = (2, 0.375)$
Branch for x_2:

$$\begin{array}{rr} F_3 & F_4 \\ \max\ 2x_1 + 3x_2 & \max\ 2x_1 + 3x_2 \\ x_1 + 2x_2 \leq 3 & x_1 + 2x_2 \leq 3 \\ 6x_1 + 8x_2 \leq 15 & 6x_1 + 8x_2 \leq 15 \\ x_1 \geq 2 & x_1 \geq 2 \\ x_2 \geq 1 & x_2 \leq 0 \\ x_1, x_2 \in \mathbb{Z}_+ & x_1, x_2 \in \mathbb{Z}_+ \end{array}$$

There is no feasible region for F_3.

The feasible region for F_4 is the set of integral points that lie on the magenta line between 2 and 2.5 inclusive.



LP relaxation in standard form:

$$
\begin{array}{rr}
F_3 \text{ linear} & F_4 \text{ linear} \\
\max 2x_1 + 3x_2 & \max 2x_1 + 3x_2 \\
x_1 + 2x_2 + x_3 = 3 & x_1 + 2x_2 + x_3 = 3 \\
6x_1 + 8x_2 + x_4 = 15 & 6x_1 + 8x_2 + x_4 = 15 \\
x_1 - x_5 = 2 & x_1 - x_5 = 2 \\
x_2 - x_6 = 1 & x_2 + x_6 = 0 \\
x_1, x_2, x_3, x_4, x_5 \geq 0 & x_1, x_2, x_3, x_4, x_5 \geq 0
\end{array}
$$

Solve F_3 LP with Gurobi:

```
Infeasible or unbounded model
```

F_3 is infeasible so this branch is terminated.

Solve F_4 LP with Gurobi:

```
Variable Values:
X[1] 2.5
X[2] 0.0
X[3] 0.5
X[4] 0.0
X[5] 0.5
X[6] 0.0

Objective Value: 5.0
```

Objective value for F_4 is 5. This is equal to the upper bound. Therefore, the F_4 branch is terminated as there is no feasible solution better than 5.

All branches are terminated so therefore $X = (1, 1)$ is the optimal solution and the optimal objective value is 5.

Verification in Gurobi:

```
Optimize a model with 2 rows, 2 columns and 4 nonzeros
Variable types: 0 continuous, 2 integer (0 binary)
Coefficient statistics:
  Matrix range     [1e+00, 8e+00]
  Objective range  [2e+00, 3e+00]
  Bounds range     [0e+00, 0e+00]
  RHS range        [3e+00, 2e+01]
Found heuristic solution: objective 4.0000000
Presolve removed 2 rows and 2 columns
Presolve time: 0.00s
Presolve: All rows and columns removed

Explored 0 nodes (0 simplex iterations) in 0.01 seconds
Thread count was 1 (of 8 available processors)

Solution count 2: 5 4

Optimal solution found (tolerance 1.00e-04)
Best objective 5.000000000000e+00, best bound 5.000000000000e+00, ga
p 0.0000%
Variable Values:
X[1] 1.0
X[2] 1.0

Objective Value: 5.0
```

4 (a)

500 courses
28 slots for finals
exams for students taking multiple courses cannot be at same time
given enrollment data

s_{ij}: binary parameter with:

$$s_{ij} = \begin{cases} 1 & \text{student } i \text{ enrolled in course } j \qquad \forall i, \forall j = 1, 2, \ldots, 500 \\ 0 & \text{otherwise} \end{cases}$$

x_{jk}: binary variable with:

$$x_{jk} = \begin{cases} 1 & \text{exam for course } j \text{ is scheduled for slot } k \qquad \forall j = 1, 2, \ldots, 500 \\ 0 & \text{otherwise} \qquad \forall k = 1, 2, \ldots, 28 \end{cases}$$

$$\min\ ?$$
$$\sum_{j,k} s_{ij} x_{jk} <= 1$$
$$x_{jk} \in \{0, 1\}$$

4 (b)

s_{ij}: binary parameter with:

$$s_{ij} = \begin{cases} 1 & \text{student } i \text{ enrolled in course } j \qquad \forall i, \forall j = 1, 2, \ldots, 500 \\ 0 & \text{otherwise} \end{cases}$$

x_{jk}: binary variable with:

$$x_{jk} = \begin{cases} 1 & \text{exam for course } j \text{ is scheduled for slot } k \qquad \forall j = 1, 2, \ldots, 500 \\ 0 & \text{otherwise} \qquad \forall k = 1, 2, \ldots, 28 \end{cases}$$

$$\min \sum_{i} \left\{ \sum_{j,k} s_{ij} x_{jk} - 1 \right\}$$
$$x_{jk} \in \{0, 1\}$$

5

x_{ij}: quantity of product that goes from factory i to customer j
y_i: binary variable with:

$$y_i = \begin{cases} 1 & \text{factory } i \text{ is used} \qquad \forall i = 1, 2, \ldots, k \\ 0 & \text{otherwise} \end{cases}$$

d_j: demand of customer j
f_i: operating cost of factory i
M_i: maximum number of units factory i can make
c_{ij}: cost of delivering 1 unit from factory i to customer j
$k = 10$ new factories
$n = 15$ customers

$$\min \left\{ \sum_{i=1}^{k} \sum_{j=1}^{n} c_{ij} x_{ij} + \sum_{i=1}^{k} y_i f_i \right\}$$

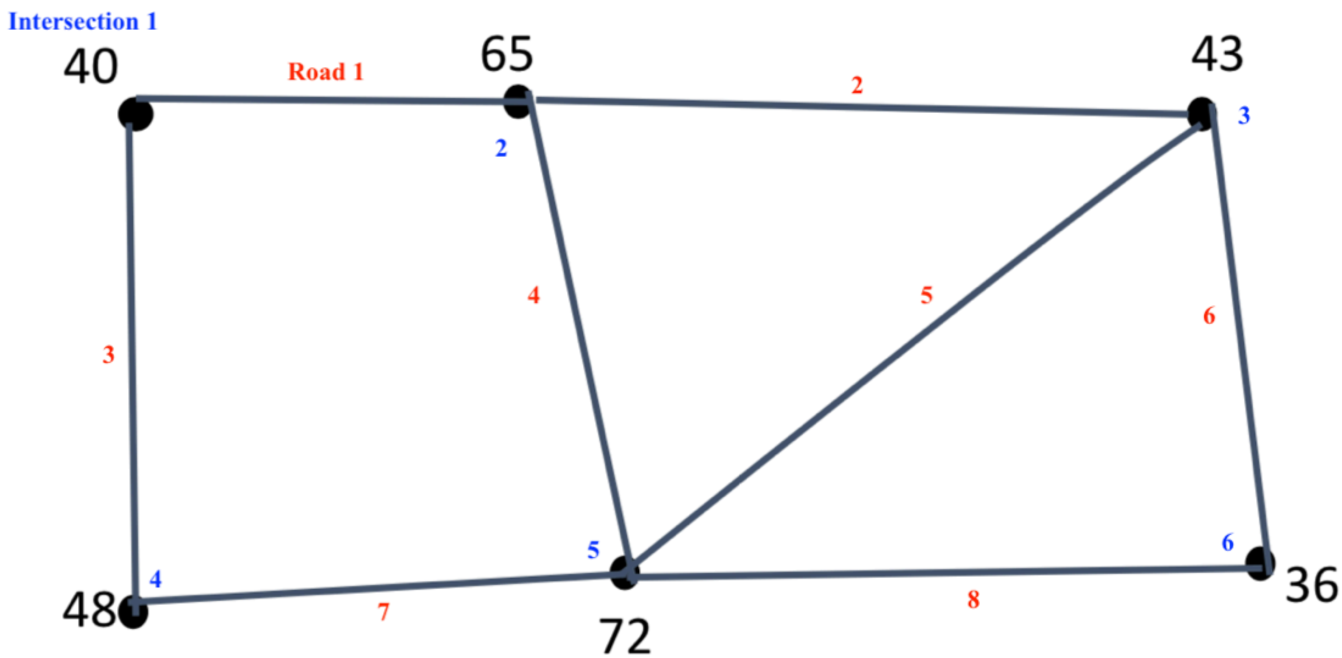
$$\sum_{i=1}^{n} x_{ij} = d_j \forall j$$

$$\sum_{j=1}^{k} x_{ij} \leq M_i y_i \forall i$$

$$x_{ij} \geq 0, \forall i, j$$

$$y_i \in \{0, 1\}$$

6 (a)



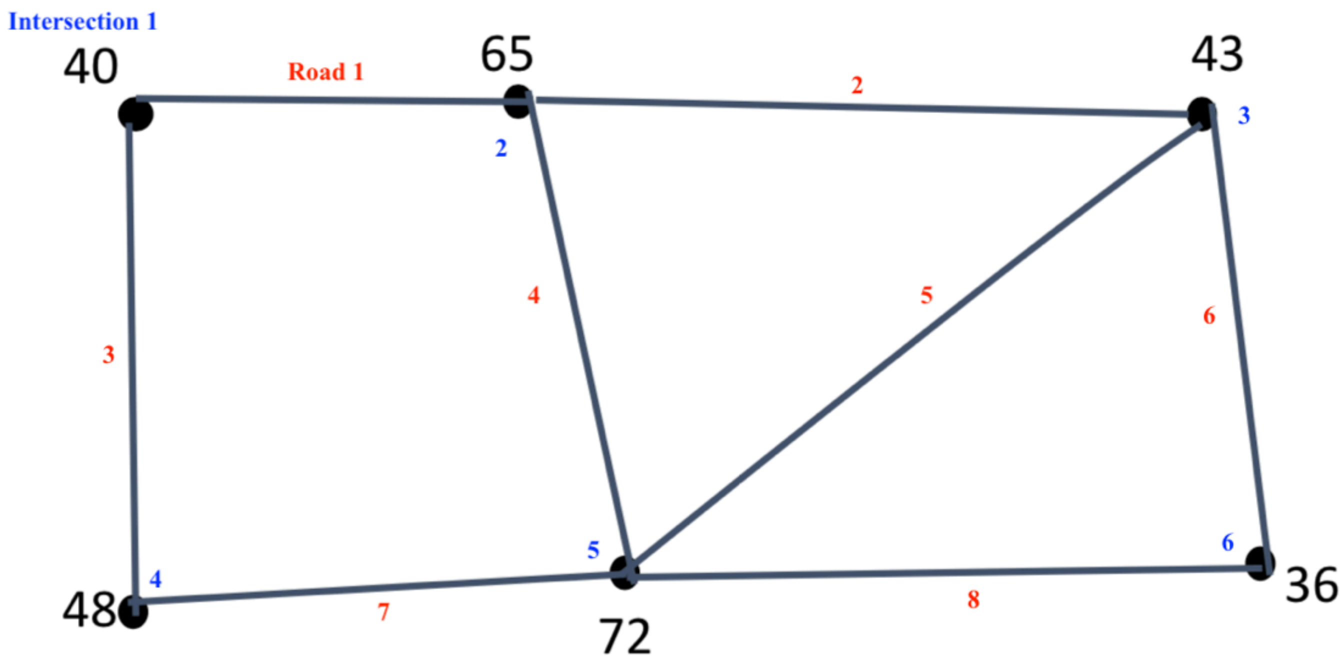
$$x_i = \begin{cases} 1 & \text{if device installed at intersection } i \\ 0 & \text{otherwise} \end{cases} \qquad \forall i = 1, \dots, 6$$

$$\begin{gathered} \min 40x_1 + 65x_2 + 43x_3 + 48x_4 + 72x_5 + 36x_6 \\ x_1 + x_2 \geq 1 \\ x_2 + x_3 \geq 1 \\ x_1 + x_4 \geq 1 \\ x_2 + x_5 \geq 1 \\ x_3 + x_5 \geq 1 \\ x_3 + x_6 \geq 1 \\ x_4 + x_5 \geq 1 \\ x_5 + x_6 \geq 1 \\ x_i \in \{0, 1\} \end{gathered}$$

In [24]:
```
c = np.array([40,65,43,48,72,36])
r = np.array([[1,2], [2,3], [1,4], [2,5], [3,5], [3,6], [4,5], [5,6]])

m = Model()

X = m.addVars([1,2,3,4,5,6], vtype = GRB.BINARY, name = 'x')

for j in r:
    m.addConstr(X[j[0]] + X[j[1]] >= 1)

m.setObjective(np.dot(X.values(),c),GRB.MINIMIZE)

m.optimize()

printOptimal(m)
```

```
Optimize a model with 8 rows, 6 columns and 16 nonzeros
Variable types: 0 continuous, 6 integer (6 binary)
Coefficient statistics:
  Matrix range     [1e+00, 1e+00]
  Objective range  [4e+01, 7e+01]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 1e+00]
Found heuristic solution: objective 221.0000000
Presolve removed 4 rows and 1 columns
Presolve time: 0.00s
Presolved: 4 rows, 5 columns, 9 nonzeros
Variable types: 0 continuous, 5 integer (5 binary)

Root relaxation: objective 1.550000e+02, 3 iterations, 0.00 seconds

    Nodes    |    Current Node    |     Objective Bounds      |
Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/N
ode Time

*    0     0               0     155.0000000  155.00000  0.00%     -
0s

Explored 0 nodes (3 simplex iterations) in 0.01 seconds
Thread count was 8 (of 8 available processors)

Solution count 2: 155 221

Optimal solution found (tolerance 1.00e-04)
Best objective 1.550000000000e+02, best bound 1.550000000000e+02, ga
p 0.0000%
Variable Values:
x[1] 1.0
x[2] 0.0
x[3] 1.0
x[4] -0.0
x[5] 1.0
x[6] 0.0

Objective Value: 155.0
```

6 (b)



$$x_i = \begin{cases} 1 & \text{if device installed at intersection } i \\ 0 & \text{otherwise} \end{cases} \quad \forall i = 1, \ldots, 6$$

$$\begin{gathered} \max\ 2x_1 + 3x_2 + 3x_3 + 2x_4 + 4x_5 + 2x_6 \\ x_1 + x_2 \leq 1 \\ x_2 + x_3 \leq 1 \\ x_1 + x_4 \leq 1 \\ x_2 + x_5 \leq 1 \\ x_3 + x_5 \leq 1 \\ x_3 + x_6 \leq 1 \\ x_4 + x_5 \leq 1 \\ x_5 + x_6 \leq 1 \\ x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \leq 2 \\ x_i \in \{0, 1\} \end{gathered}$$

In [25]:
```
r = np.array([[1,2], [2,3], [1,4], [2,5], [3,5], [3,6], [4,5], [5,6]])
c = np.array([2,3,3,2,4,2])

m = Model()

X = m.addVars([1,2,3,4,5,6], vtype = GRB.BINARY, name = 'x')

m.addConstr(sum(X.values()) <= 2)

for j in r:
    m.addConstr(X[j[0]] + X[j[1]] <= 1)

m.setObjective(np.dot(X.values(),c),GRB.MAXIMIZE)

m.optimize()

printOptimal(m)
```

```
Optimize a model with 9 rows, 6 columns and 22 nonzeros
Variable types: 0 continuous, 6 integer (6 binary)
Coefficient statistics:
  Matrix range     [1e+00, 1e+00]
  Objective range  [2e+00, 4e+00]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 2e+00]
Found heuristic solution: objective 5.0000000
Presolve removed 3 rows and 0 columns
Presolve time: 0.00s
Presolved: 6 rows, 6 columns, 18 nonzeros
Variable types: 0 continuous, 6 integer (6 binary)

Root relaxation: objective 6.000000e+00, 3 iterations, 0.00 seconds

    Nodes    |    Current Node    |     Objective Bounds      |
Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/N
ode Time

*    0     0               0       6.0000000    6.00000  0.00%     -
0s

Explored 0 nodes (3 simplex iterations) in 0.01 seconds
Thread count was 8 (of 8 available processors)

Solution count 2: 6 5

Optimal solution found (tolerance 1.00e-04)
Best objective 6.000000000000e+00, best bound 6.000000000000e+00, ga
p 0.0000%
Variable Values:
x[1] 1.0
x[2] 0.0
x[3] -0.0
x[4] -0.0
x[5] 1.0
x[6] 0.0

Objective Value: 6.0
```