Session 18: Mixed Integer Programming (MIP) Solutions

1. Geometry of Linear Programming

Example 1: Solve the following LP graphically without the use of a computer.

$$\begin{aligned}
&\text{Maximize} && 2X+Y \\
&\text{subject to:} && \\
&\text{(Material 1)} && X+2Y \leq 6 \\
&\text{(Material 2)} && 3Y \leq 6 \\
&\text{(Non-negativity)} && X, Y \geq 0
\end{aligned}$$

Q1: Solve the following LP graphically (find optimal solution and objective value). Which constraints are binding at the optimal solution? For each constraint, determine the sign of the its shadow price. (The sign means whether it is positive, negative, or zero.)

$$\begin{aligned}
\text{Maximize} &\quad 10X+20Y \\
\text{subject to:} &\quad \\
\text{(Material 1)} &\quad 4X+Y \leq 60 \\
\text{(Material 2)} &\quad 2Y \leq 48 \\
\text{(Labor)} &\quad X+Y \leq 30 \\
\text{(Non-negativity)} &\quad X, Y \geq 0
\end{aligned}$$

Q2: How does the optimal solution changes if both X and Y have to be integer multiples of 10?

2. Introduction to MIPs

Example 2: Suppose that in the production planning example of Q1, there is an additional fixed cost of 90 for using any amount material 2. If we pay this cost, then we have 48 units of material 2 at our disposal, otherwise we have no material 2. What is the optimal profit and corresponding production plan?

Decision Variables:

- Let X and Y be the amount of each product to produce. (integer)
- Let M_2 be a binary variable corresponding to whether we use material 2 at all.

Objective and Constraints:

$$\begin{aligned}
\text{Maximize} &\quad 10X+20Y-90M_2 \\
\text{subject to:} &\quad \\
\text{(Material 1)} &\quad 4X+Y \leq 60 \\
\text{(Material 2)} &\quad 2Y \leq 48M_2 \\
\text{(Labor)} &\quad X+Y \leq 30 \\
\text{(Non-negativity)} &\quad X, Y \geq 0 \\
\text{(Binary)} &\quad M_2 \in \{0,1\}
\end{aligned}$$

[1]:
```
from gurobipy import Model, GRB
mod=Model()

X=mod.addVar()
```

```
Y=mod.addVar()
M2=mod.addVar(vtype=GRB.BINARY)

mod.setObjective(20*X+10*Y-90*M2,sense=GRB.MAXIMIZE)
mod.addConstr(4*X+Y <=60)
mod.addConstr(2*Y<=48*M2)
mod.addConstr(X+Y<=30)

mod.optimize()
```

```
Academic license - for non-commercial use only
Optimize a model with 3 rows, 3 columns and 6 nonzeros
Variable types: 2 continuous, 1 integer (1 binary)
Coefficient statistics:
  Matrix range     [1e+00, 5e+01]
  Objective range  [1e+01, 9e+01]
  Bounds range     [1e+00, 1e+00]
  RHS range        [3e+01, 6e+01]
Found heuristic solution: objective 300.0000000
Presolve time: 0.00s
Presolved: 3 rows, 3 columns, 7 nonzeros
Variable types: 2 continuous, 1 integer (1 binary)

Root relaxation: objective 3.100000e+02, 2 iterations, 0.00 seconds

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

*    0     0               0     310.0000000  310.00000  0.00%     -    0s

Explored 0 nodes (2 simplex iterations) in 0.02 seconds
Thread count was 4 (of 4 available processors)

Solution count 2: 310 300

Optimal solution found (tolerance 1.00e-04)
Best objective 3.100000000000e+02, best bound 3.100000000000e+02, gap 0.0000%
```

```
[2]: print('Optimal objective:', mod.objval)
     print('\tX:',X.x)
     print('\tY:',Y.x)
     print('\tM2:',M2.x)
```

```
Optimal objective: 310.0
        X: 10.0
        Y: 20.0
        M2: 1.0
```

Q3: Consider the LP as in Q1 but now X and Y have to be integer multiples of 10. Formulate this as a MIP and solve it using Gurobi.

Decision Variables:

- Let X and Y be the amount of each product to produce. (integer)
- Let I_1 and I_2 be integers.

Objective and Constraints:

$$\begin{aligned}
\text{Maximize} \quad & 10X + 20Y \\
\text{subject to:} \quad & \\
\text{(Material 1)} \quad & 4X + Y \leq 60 \\
\text{(Material 2)} \quad & 2Y \leq 48 \\
\text{(Labor)} \quad & X + Y \leq 30 \\
\text{(X multiple of 10)} \quad & X = 10I_1 \\
\text{(Y multiple of 10)} \quad & Y = 10I_2 \\
\text{(Non-negativity)} \quad & X, Y \geq 0 \\
\text{(Integer)} \quad & I_1, I_2 \in \mathbb{Z}
\end{aligned}$$

[3]:
```
from gurobipy import Model, GRB

mod=Model()

X=mod.addVar()
Y=mod.addVar()
I1=mod.addVar(vtype=GRB.INTEGER)
I2=mod.addVar(vtype=GRB.INTEGER)

mod.setObjective(20*X+10*Y,sense=GRB.MAXIMIZE)
mod.addConstr(4*X+Y <=60)
mod.addConstr(2*Y<=48)
mod.addConstr(X+Y<=30)
mod.addConstr(X==10*I1)
mod.addConstr(Y==10*I2)

mod.setParam('outputflag',False)
mod.optimize()

print('Optimal objective:', mod.objval)
print('\tX:',X.x)
print('\tY:',Y.x)
```

```
Optimal objective: 400.0
        X: 10.0
        Y: 20.0
```

Q4: Consider the production planning problem from Q1 with the following modifications:

- All production quantities must be integers.
- You have the additional ability to purchase more units of Material 1 at a cost of 3 per unit. However, you can only purchase them in batches of 9 units.
- Producing any positive amounts of X would incur a fixed cost of 100.
- The number of units produced for Y is either zero or at least 5.

What is the optimal profit and how is it be achieved?
Decision Variables:

- Let X and Y be the amount of each product to produce. (integer)
- Let N_1 be the number of batches of material 1 to purchase. (integer)
- Let Z_1 be 1 if we produce any amount of X, and 0 otherwise. (binary)
- Let Z_2 be 1 if we produce any amount of Y, and 0 otherwise. (binary)

Objective and Constraints:

$$\begin{array}{rc}
\text{Maximize} & 10X + 20Y - 27N_1 - 100Z_1 \\
\text{subject to:} & \\
\text{(Material 1)} & 4X + Y \leq 60 + 9N_1 \\
\text{(Material 2)} & 2Y \leq 48 \\
\text{(Labor)} & X + Y \leq 30 \\
\text{(Whether we produce X)} & X \leq 1000Z_1 \\
\text{(Minimum production size)} & 5Z_2 \leq Y \leq 1000Z_2 \\
\text{(Non-negativity)} & X, Y \geq 0
\end{array}$$

[4]:
```
from gurobipy import Model, GRB
mod=Model()

X=mod.addVar(vtype=GRB.INTEGER)
Y=mod.addVar(vtype=GRB.INTEGER)
N1=mod.addVar(vtype=GRB.INTEGER)
Z1=mod.addVar(vtype=GRB.BINARY)
Z2=mod.addVar(vtype=GRB.BINARY)

mod.setObjective(20*X+10*Y-27*N1-100*Z1,sense=GRB.MAXIMIZE)
mod.addConstr(4*X+Y <=60+9*N1)
mod.addConstr(2*Y<=48)
mod.addConstr(X+Y<=30)
mod.addConstr(X<=1000*Z1)
# Note that each side of this constraint must be coded separately
mod.addConstr(5*Z2 <= Y )
mod.addConstr(Y<=1000*Z2)

mod.setParam('outputflag',False)
mod.optimize()

print('Optimal objective:', mod.objval)
print('\tX:',X.x)
print('\tY:',Y.x)
print('\tN1:',N1.x)
print('\tZ1:',Z1.x)
print('\tZ2:',Z2.x)
```

```
Optimal objective: 315.0
        X: 25.0
        Y: 5.0
        N1: 5.0
```

```
Z1: 1.0
Z2: 1.0
```