Alex Hines, Gerald Jones, Kimon Swanson

Home Work 6: Simplex Algorithm

ISE522 Spg 22

Notebook Links:

1. Simplex Solution
2. Gurobi Implementation and Solution
3. Solution Discussion

Problem Description:

> 1. Solve the following LP using the simplex algorithm. Verify your solution using a solver of your choice.

Max.

$$z = 5x1 + x2$$

s.t.

$$2 \cdot x_1 + x_2 \leq 6$$

$$x_1 - x_2 \leq 0$$

$$x_1, x_2 \geq 0$$

Module imports and data loading

In [11]:
```
from _GUROBI_TOOLS_.GUROBI_MODEL_BUILDING_TOOLS import *
from _NOTE_BOOK_UTILS import *
note_book_title = "_HW_6.ipynb"
```

Simplex Algorithm Solution

display data for problem

Gurobi Implementation and Solution

In [9]:
```
try:
    # instantiate model object
    m = gp.Model("Simplex_Verification")

    ###########################################################################################
    ################################### Variables set up ######################################
    ###########################################################################################
    Xi = m.addVars(2, vtype=GRB.CONTINUOUS, name="X", lb=0)        # X1, X2 creation and >= 0
    Z = m.addVar(vtype=GRB.CONTINUOUS, name="Z", lb=0)             # objective


    ###########################################################################################
    ################################### Objective set up ######################################
    ###########################################################################################
    m.setObjective(Z, GRB.MAXIMIZE)

    ###########################################################################################
    ################################### Constraint set up #####################################
    ###########################################################################################
    m.addConstr(Z == 5*Xi[0] + Xi[1])                                  # Optimization Expression
    m.addConstr(2*Xi[0] + Xi[1] <= 6)
    m.addConstr(Xi[0] - Xi[1] <= 0)


    ###########################################################################################
    ################################### SOLVE:OPTIMIZE ########################################
    ###########################################################################################


    m.optimize()

    ###########################################################################################
    ################################### Display Results #######################################
    ###########################################################################################
    displayDecisionVars(m, end_sentinel="6")

    print("\n-------------Does it make sense?----------------------")
    print('Obj: {:.2f}'.format(m.ObjVal))


# catch some math errors
except gp.GurobiError as e:
    print('Error code ' + str(e.errno) + ': ' + str(e))

except AttributeError:
    print('Encountered an attribute error')
```

```
Gurobi Optimizer version 9.5.0 build v9.5.0rc5 (win64)
Thread count: 6 physical cores, 12 logical processors, using up to 12 threads
Optimize a model with 3 rows, 3 columns and 7 nonzeros
Model fingerprint: 0x88501388
Coefficient statistics:
  Matrix range     [1e+00, 5e+00]
  Objective range  [1e+00, 1e+00]
  Bounds range     [0e+00, 0e+00]
  RHS range        [6e+00, 6e+00]
Presolve removed 3 rows and 3 columns
Presolve time: 0.00s
Presolve: All rows and columns removed
Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    1.2000000e+01   0.000000e+00   0.000000e+00      0s

Solved in 0 iterations and 0.01 seconds (0.00 work units)
Optimal objective  1.200000000e+01
X[0] 2.00000
X[1] 2.00000
Z 12.00000
-------------------------------------



-------------Does it make sense?----------------------
Obj: 12.00
```

Solution Discussion

> The solution produced by Gurobi Agrees with hour Simplex derives solution. Accroding to both **X1 and X2 should be 2** leading to an optimal value of **z = 12**.

In [ ]:
```
# save the notebook as a pdf
to_PDF(note_book_title)
```