!pip install -q amplpy

from amplpy import AMPL, tools

ampl = tools.ampl\_notebook(

modules=["cplex"],

license\_uuid="906b5cb9-52ea-4345-9735-e9d2687b1b40")

%%writefile gardeners.mod

# Decision Variables

var x1 >= 0; # pounds of chemical C-30 used

var x2 >= 0; # pounds of chemical C-92 used

var x3 >= 0; # pounds of chemical D-21 used

var x4 >= 0; # pounds of chemical E-11 used

# Objective Function: Minimize the total cost

minimize TotalCost:

0.12\*x1 + 0.09\*x2 + 0.11\*x3 + 0.04\*x4;

# Constraints

subject to E11\_percentage:

x4 >= 0.15 \* 50; # Chemical E-11 comprises at least 15% of the blend

subject to C92\_C30\_percentage:

x1 + x2 >= 0.45 \* 50; # C-92 and C-30 constitute at least 45% of the blend

subject to D21\_C92\_percentage:

x3 + x2 <= 0.30 \* 50; # D-21 and C-92 together constitute no more than 30% of the blend

subject to Package\_Weight:

x1 + x2 + x3 + x4 = 50; # Golding-Grow is packaged and sold in 50-pound bags

%%ampl\_eval

reset;

model gardeners.mod

option solver cplex;

solve;

display x1, x2, x3, x4, TotalCost;

%%writefile scheduling.mod

var f1 >= 0;

var f2 >= 0;

var f3 >= 0;

var p1 >= 0;

var p2 >= 0;

var p3 >= 0;

var p4 >= 0;

#Objective Function

minimize TotalCost:

(14 \* 8 \* (f1 + f2 + f3)) + (12 \* 4 \*(p1 + p2 + p3 + p4));

#Constraints

subject to 8am\_noon:

f1 + p1 >= 4;

subject to noon\_4p:

f1 +f2 + p2 >= 8;

subject to 4p\_8p:

f2 + f3 + p3 >= 10;

subject to 8p\_midnight:

f3 + p4 >= 6;

subject to full\_PartRatioShift1:

f1 >= p1;

subject to full\_PartRatioShift2:

f1 + f2 >= p2;

subject to full\_PartRatioShift3:

f2 + f3 >= p3;

subject to full\_PartRatioShift4:

f3 >= p4;

%%ampl\_eval

reset;

model scheduling.mod

option solver cplex;

solve;

display TotalCost, f1, f2, f3, p1, p2, p3, p4;

%%writefile demands.mod

# Decision Variables

var m1 >= 0; # Number of units produced in month 1

var m2 >= 0; # Number of units produced in month 2

var m3 >= 0; # Number of units produced in month 3

var m4 >= 0; # Number of units produced in month 4

var I1 >= 0; # Inventory at the end of month 1

var I2 >= 0; # Inventory at the end of month 2

var I3 >= 0; # Inventory at the end of month 3

var I4 >= 0; # Inventory at the end of month 4

# Objective Function: Minimize the total net cost

minimize TotalCost:

(5 \* m1 + 8 \* m2 + 4 \* m3 + 7 \* m4) + (2 \* I1 + 2 \* I2 + 2 \* I3 + 2 \* I4);

# Constraints

subject to EndingInventory1:

I1 = 0 + m1 - 50;

subject to EndingInventory2:

I2 = I1 + m2 - 65;

subject to EndingInventory3:

I3 = I2 + m3 - 100;

subject to EndingInventory4:

I4 = I3 + m4 - 70;

%%ampl\_eval

reset;

model demands.mod

option solver cplex;

solve;

display m1, m2, m3, m4, I1, I2, I3, I4, TotalCost;