Name: Eric Haddad

Student ID: 40202855

Date: 12/03/2024

Import Packages

In [1]:
```
import pyomo.environ as pyo
from pyomo.opt import SolverFactory
```

Model Definition

In [2]:
```
# Instantiate a concrete model
m = pyo.ConcreteModel(name = "Pencil Production")
```

Parameters

In [3]:
```
# Profit per unit ($/unit)
m.pr = pyo.Param(initialize = 1.9) # Profit per unit for Red pencils
m.pb = pyo.Param(initialize = 1.1) # Profit per unit for Blue pencils
m.pw = pyo.Param(initialize = 2.8) # Profit per unit for White pencils

pr = m.pr
pb = m.pb
pw = m.pw

# Capacity of Plant
m.c1 = pyo.Param(initialize = 390) # Time Capacity (min/day) for Machine 1
m.c2 = pyo.Param(initialize = 348) # Time Capacity (hours/day) for Machine 2
m.c3 = pyo.Param(initialize = 150) # Capacity to process White pencils (units/day) for Machine 1
m.c4 = pyo.Param(initialize = 220) # Capacity to process White pencils (units/day) for Machine 2

c1 = m.c1
c2 = m.c2
c3 = m.c3
c4 = m.c4

# Processes Time for each Machine
# Process 1
m.t11 = pyo.Param(initialize = 0.8) # Coefficient of x1 (Red) in Processing Time 1 equation
m.t21 = pyo.Param(initialize = 1.3) # Coefficient of x2 (Blue) in Processing Time 1 equation
m.t31 = pyo.Param(initialize = 1.1) # Coefficient of x3 (White) in Processing Time 1 equation

# Process 2
m.t12 = pyo.Param(initialize = 1.8) # Coefficient of x1 (Red) in Processing Time 2 equation
m.t22 = pyo.Param(initialize = 0.5) # Coefficient of x2 (Blue) in Processing Time 2 equation
m.t32 = pyo.Param(initialize = 1.3) # Coefficient of x3 (White) in Processing Time 2 equation

t11 = m.t11
t21 = m.t21
t31 = m.t31
t12 = m.t12
t22 = m.t22
t32 = m.t32
```

Decision Variables

In [4]:
```
# Setting up the variables x1, x2 and x3
m.x1 = pyo.Var(domain = pyo.NonNegativeReals) # x1: Variable for Red pencils
m.x2 = pyo.Var(domain = pyo.NonNegativeReals) # x2: Variable for Blue pencils
m.x3 = pyo.Var(domain = pyo.NonNegativeReals) # x3: Variable for White pencils

x1 = m.x1
x2 = m.x2
x3 = m.x3
```

Objective Function

$$maxZ = 1.9x_1 + 1.1x_2 + 2.8x_3$$

In [5]:
```
m.obj = pyo.Objective(expr = pr*x1 + pb*x2 + pw*x3, sense = pyo.maximize)
```

Constraints

Constraint 1: The time capacity for machine 1 cannot be exceeded:

$$0.8x_1 + 1.3x_2 + 1.1x_3 <= 6.5$$

In [6]:
```
m.cons1 = pyo.Constraint(expr = t11*x1 + t21*x2 + t31*x3 <= c1, doc = "Time Capacity (Machine 1)")
```

Constraint 2: The time capacity for machine 2 cannot be exceeded:

$$1.8x_1 + 0.5x_2 + 1.3x_3 <= 5.8$$

In [7]:
```
m.cons2 = pyo.Constraint(expr = t12*x1 + t22*x2 +t32*x3 <= c2, doc = "Time Capacity (Machine 2)")
```

Constraint 3: The capacity to process white pencils for machine 1 cannot be exceeded:

$$x_3 <= 150$$

In [8]:
```
m.cons3 = pyo.Constraint(expr = x3 <= c3, doc = "Processing Capacity for White Pencils (Machine 1)")
```

Constraint 4: The capacity to process white pencils for machine 2 cannot be exceeded:

$$x_3 <= 220$$

In [9]:
```
m.cons4 = pyo.Constraint(expr = x3 <= c4, doc = "Processing Capacity for White Pencils (Machine 2)")
```

The non-negativity constraint is already in the decision variables definition.

Therefore we're done with setting up the objective function and the constraints.

Solving and Obtaining Results

In [10]:
```
# Creating a dual suffix component
# Instance the dual model, in which suffix data will be imported from the primal solver
m.dual = pyo.Suffix(direction = pyo.Suffix.IMPORT)
```

In [11]:
```
solver = SolverFactory('gurobi')
results = solver.solve(m)
```

In [12]:
```
# Print the solution of the model
print(results)
```

```
Problem:
- Name: x1
  Lower bound: 664.8556701030927
  Upper bound: 664.8556701030927
  Number of objectives: 1
  Number of constraints: 4
  Number of variables: 3
  Number of binary variables: 0
  Number of integer variables: 0
  Number of continuous variables: 3
  Number of nonzeros: 8
  Sense: maximize
Solver:
- Status: ok
  Return code: 0
  Message: Model was solved to optimality (subject to tolerances), and an optimal solution is available.
  Termination condition: optimal
  Termination message: Model was solved to optimality (subject to tolerances), and an optimal solution is availa
ble.
  Wall time: 0.0007519721984863281
  Error rc: 0
  Time: 0.45563483238220215
Solution:
- number of solutions: 0
  number of solutions displayed: 0
```

Questions & Answers

- Question 1:

What is the optimal production plan and what is the profit for that solution?

If needed, round up to 2 decimals.

In [13]:
```
# Optimal Production Plan rounded up to 2 decimals
optimal_x1 = pyo.value(m.x1)
optimal_x2 = pyo.value(m.x2)
optimal_x3 = pyo.value(m.x3)
optimal_profit = pyo.value(m.obj)

print("Optimal Production Plan and Profit:")
print("Production Plan:")
print("x1 =", round(optimal_x1, 2))
print("x2 =", round(optimal_x2, 2))
print("x3 =", round(optimal_x3, 2))
print("Profit: $", round(optimal_profit, 2))

print("The company should produce {:.1f} batches of Red pencils, {:.1f} of Blue pencils, and {:.1f} of White pe
```

```
Optimal Production Plan and Profit:
Production Plan:
x1 = 44.54
x2 = 145.67
x3 = 150.0
Profit: $ 664.86
The company should produce 44.5 batches of Red pencils, 145.7 of Blue pencils, and 150.0 of White pencils for a
total profit of $664.86 per day.
```

- Question 2:

An increase in the demand for blue pencils raised its profit by 30%.

Please answer the questions of point 1 once again.

If needed, round up to 2 decimals.

In [14]:
```
# Increase in the demand for blue pencils raised its profit by 30%
m.obj.deactivate()
m.newobj = pyo.Objective(expr = pr*x1 + 1.3*pb*x2 + pw*x3, sense = pyo.maximize)

# Solve the updated model
results2 = solver.solve(m)

# Retrieve and print the updated solution
updated_optimal_x1 = pyo.value(m.x1)
updated_optimal_x2 = pyo.value(m.x2)
updated_optimal_x3 = pyo.value(m.x3)
updated_optimal_profit = pyo.value(m.newobj)
```

```
print("Optimal Production Plan and Profit with Increased Demand for Blue Pencils:")
print("Production Plan:")
print("x1 =", round(updated_optimal_x1, 2))
print("x2 =", round(updated_optimal_x2, 2))
print("x3 =", round(updated_optimal_x3, 2))
print("Profit: $", round(updated_optimal_profit, 2))
print("""The company should produce {:.1f} batches of Red pencils, {:.1f} of Blue pencils, and {:.1f} of White p
  for a total profit of ${:,.2f} per day.""".
      format(pyo.value(x1), pyo.value(x2), pyo.value(x3), pyo.value(m.newobj)))
```

```
Optimal Production Plan and Profit with Increased Demand for Blue Pencils:
Production Plan:
x1 = 44.54
x2 = 145.67
x3 = 150.0
Profit: $ 712.93
The company should produce 44.5 batches of Red pencils, 145.7 of Blue pencils, and 150.0 of White pencils
  for a total profit of $712.93 per day.
```

- Question 3:

With the profit in point 1 for blue pencils, calculate the dual variables.

Please explain the meaning of the results.

In [15]:
```
# Showing the dual variables
m.obj.activate()
print("Duals:")
print("-------------------------------------------------------------------------")
for c in m.component_objects(pyo.Constraint, active = True):
    print("'{}' constraint: {:,.2f}".format(c.doc, m.dual[c]))

duals = []
for c in m.component_objects(pyo.Constraint, active = True):
    duals.append(m.dual[c])
print("-------------------------------------------------------------------------")
print("""Meaning of the Results:
The dual variables represent the shadow prices or marginal values of the resources (constraints) in the problem
If constraint 1 increases by 1 unit, the profit would increase by """, round(duals[0], 2))
```

```
Duals:
-------------------------------------------------------------------------
'Time Capacity (Machine 1)' constraint: 0.84
'Time Capacity (Machine 2)' constraint: 0.68
'Processing Capacity for White Pencils (Machine 1)' constraint: 0.99
'Processing Capacity for White Pencils (Machine 2)' constraint: 0.00
-------------------------------------------------------------------------
Meaning of the Results:
The dual variables represent the shadow prices or marginal values of the resources (constraints) in the problem.
If constraint 1 increases by 1 unit, the profit would increase by  0.84
```

- Question 4:

Since green is the best color in the world, the company wants to see if it is a good idea to include that pencil in their production.

Assuming that all the capacities remain the same and that the parameters are as follows, please explain what would be your advice to the company's CEO. For blue pencils, use the profit of point 1.

Adding Green pencils to the model

In [16]:
```
m.obj.deactivate()
m.newobj.deactivate()
# Define the parameters
m.pg = pyo.Param(initialize = 0.4)

pg = m.pg

# Define the decision variable
m.x4 = pyo.Var(domain = pyo.NonNegativeReals)

x4 = m.x4

# Define the objective function
m.obj4 = pyo.Objective(expr = pr*x1 + pb*x2 + pw*x3 + pg*x4, sense = pyo.maximize)

# Processes Time for each Machine
# Process 1
m.t41 = pyo.Param(initialize = 0.7) # Coefficient of x4 (Green) in Processing Time 1 equation
```

```
m.t42 = pyo.Param(initialize = 1.2) # Coefficient of x4 (Green) in Processing Time 2 equation

t41 = m.t41
t42 = m.t42

# Define the new constraints
m.cons5 = pyo.Constraint(expr = t41*x4 <= 390, doc = 'Time Capacity (Machine 2) for Green pencils')
m.cons6 = pyo.Constraint(expr = t42*x4 <= 348, doc = 'Time Capacity (Machine 2) for Green pencils')

# Solve the model
results3 = solver.solve(m)


print("Duals:")
print("----------------------------------------------------------------------------")
for c in m.component_objects(pyo.Constraint, active = True):
    print("'{}' constraint: {:,.2f}".format(c.doc, m.dual[c]))

duals = []
for c in m.component_objects(pyo.Constraint, active = True):
    duals.append(m.dual[c])
```

```
Duals:
----------------------------------------------------------------------------
'Time Capacity (Machine 1)' constraint: 0.53
'Time Capacity (Machine 2)' constraint: 0.82
'Processing Capacity for White Pencils (Machine 1)' constraint: 1.15
'Processing Capacity for White Pencils (Machine 2)' constraint: 0.00
'Time Capacity (Machine 2) for Green pencils' constraint: 0.00
'Time Capacity (Machine 2) for Green pencils' constraint: 0.33
```

- After adding green pencils to our model, we received positive dual variables. This indicates that the addition of green pencils has increased the value of the objective function, while still satisfying all constraints.