Problema 1: PROGRAMAÇÃO DA PRODUÇÃO MULTI-PERÍODO

Importacao bibliotecas

In [ ]:
```
from pulp import *
```

Função Objetivo: Min [Producao]*[Custo Producao] + [Estoque]*[Custo Estoque]

Restrições (para cada mês i):

[Produção_i] + [Estoque_i] >= [Demanda_i]

[Estoque_i] = [Produção_i-1] + [Estoque_i-1] - [Demanda_i-1]

[Estoque_0] = 0

In [ ]:
```
problem = LpProblem("ProblemaProducao", LpMinimize)

custo_producao_A = [ 120,  126,  129,  140,  135,  138,  133,  130,  130,
custo_estoque_A  = [   5,    5,    7,    8,    7,    6,    5,    4,    4,
demanda_A        = [1000, 1500, 2000, 3500, 2500, 3200, 2800, 4000, 2000,

N = len(demanda_A)
meses = list(range(N))

producao_A = LpVariable.dicts("producao_A", meses, lowBound=0,  upBound=3
estoque_A  = LpVariable.dicts("estoque_A", meses, lowBound=0,  upBound=10


#Objective function
problem += lpSum([producao_A[i]*custo_producao_A[i] + custo_estoque_A[i]*

#Constraints
problem += estoque_A[0] == 0

for i in meses:       problem += producao_A[i] + estoque_A[i] >= demanda_
for i in range(1, N): problem += estoque_A[i] == producao_A[i-1] + estoqu

#Solve
problem.solve(PULP_CBC_CMD(msg=0))

#Resultados
for v in problem.variables():
    print(v.name, "=", v.varValue)

print("FO =", value(problem.objective))

print("Current Status =", LpStatus[problem.status])
```

```
estoque_A_0 = 0.0
estoque_A_1 = 1000.0
estoque_A_2 = 0.0
estoque_A_3 = 1000.0
estoque_A_4 = 500.0
estoque_A_5 = 1000.0
estoque_A_6 = 800.0
estoque_A_7 = 1000.0
estoque_A_8 = 0.0
estoque_A_9 = -0.0
producao_A_0 = 2000.0
producao_A_1 = 500.0
producao_A_2 = 3000.0
producao_A_3 = 3000.0
producao_A_4 = 3000.0
producao_A_5 = 3000.0
producao_A_6 = 3000.0
producao_A_7 = 3000.0
producao_A_8 = 2000.0
producao_A_9 = 2500.0
FO = 3328500.0
Current Status = Optimal
```

In [ ]:
```
custo_producao_B = [ 82, 90, 92, 87, 85, 95, 91, 88, 85, 90] #R$/unidade
custo_estoque_B  = [5, 4, 6, 3, 2, 8, 4, 4, 3, 2] #R$/unidade
demanda_B        = [600, 950, 900, 800, 1200, 1000, 1300, 1500, 1100, 100

N = len(demanda_A)
meses = list(range(N))

producao_B = LpVariable.dicts("producao_B", meses, lowBound=0,  upBound=1
estoque_B  = LpVariable.dicts("estoque_B", meses, lowBound=0,  upBound=10


#Objective function
problem += lpSum([producao_A[i]*custo_producao_A[i] + custo_estoque_A[i]*
                  producao_B[i]*custo_producao_B[i] + custo_estoque_B[i]*
                  for i in meses])

#Constraints
problem += estoque_B[0] == 0

for i in meses:       problem += producao_B[i] + estoque_B[i] >= demanda_
for i in range(1, N): problem += estoque_B[i] == producao_B[i-1] + estoqu
```

In [ ]:
```
#Solve
problem.solve(PULP_CBC_CMD(msg=0))

#Resultados
for v in problem.variables():
    print(v.name, "=", v.varValue)

print("FO =", value(problem.objective))

print("Current Status =", LpStatus[problem.status])
```

```
estoque_A_0 = 0.0
estoque_A_1 = 1000.0
estoque_A_2 = 0.0
estoque_A_3 = 1000.0
estoque_A_4 = 500.0
estoque_A_5 = 1000.0
estoque_A_6 = 800.0
estoque_A_7 = 1000.0
estoque_A_8 = 0.0
estoque_A_9 = -0.0
estoque_B_0 = 0.0
estoque_B_1 = 900.0
estoque_B_2 = 0.0
estoque_B_3 = 0.0
estoque_B_4 = 0.0
estoque_B_5 = 300.0
estoque_B_6 = 0.0
estoque_B_7 = 0.0
estoque_B_8 = 0.0
estoque_B_9 = 400.0
producao_A_0 = 2000.0
producao_A_1 = 500.0
producao_A_2 = 3000.0
producao_A_3 = 3000.0
producao_A_4 = 3000.0
producao_A_5 = 3000.0
producao_A_6 = 3000.0
producao_A_7 = 3000.0
producao_A_8 = 2000.0
producao_A_9 = 2500.0
producao_B_0 = 1500.0
producao_B_1 = 50.0
producao_B_2 = 900.0
producao_B_3 = 800.0
producao_B_4 = 1500.0
producao_B_5 = 700.0
producao_B_6 = 1300.0
producao_B_7 = 1500.0
producao_B_8 = 1500.0
producao_B_9 = 600.0
FO = 4241000.0
Current Status = Optimal
```

O plano de produção de A não mudou

In [ ]:
```python
for i in meses:
    problem += producao_A[i] + producao_B[i] <= 4000
    problem += estoque_A[i] + estoque_B[i] <= 1800
```

In [ ]:
```python
#Solve
problem.solve(PULP_CBC_CMD(msg=0))

#Resultados
for v in problem.variables():
    print(v.name, "=", v.varValue)

print("FO =", value(problem.objective))

print("Current Status =", LpStatus[problem.status])
```

```
estoque_A_0 = 0.0
estoque_A_1 = 900.0
estoque_A_2 = 500.0
estoque_A_3 = 1000.0
estoque_A_4 = 500.0
estoque_A_5 = 1000.0
estoque_A_6 = 800.0
estoque_A_7 = 1000.0
estoque_A_8 = 0.0
estoque_A_9 = 0.0
estoque_B_0 = 0.0
estoque_B_1 = 900.0
estoque_B_2 = 200.0
estoque_B_3 = 800.0
estoque_B_4 = 1000.0
estoque_B_5 = 800.0
estoque_B_6 = 800.0
estoque_B_7 = 500.0
estoque_B_8 = 0.0
estoque_B_9 = 400.0
producao_A_0 = 1900.0
producao_A_1 = 1100.0
producao_A_2 = 2500.0
producao_A_3 = 3000.0
producao_A_4 = 3000.0
producao_A_5 = 3000.0
producao_A_6 = 3000.0
producao_A_7 = 3000.0
producao_A_8 = 2000.0
producao_A_9 = 2500.0
producao_B_0 = 1500.0
producao_B_1 = 250.0
producao_B_2 = 1500.0
producao_B_3 = 1000.0
producao_B_4 = 1000.0
producao_B_5 = 1000.0
producao_B_6 = 1000.0
producao_B_7 = 1000.0
producao_B_8 = 1500.0
producao_B_9 = 600.0
FO = 4263200.0
Current Status = Optimal
```

Problema 2: PROGRAMAÇÃO DE PESSOAL

Importação Bibliotecas

In [ ]:
```
from pulp import *
```

In [ ]:
```
problem = LpProblem("ProblemaProducao", LpMinimize)

empregados_requeridos = [ 17,  13,  15,  19,  14,  16,  11] #n funcionari

N = len(empregados_requeridos)
dias = list(range(N))

funcionarios = LpVariable.dicts("funcionarios", dias, lowBound=0, cat="In

#Objective function
problem += lpSum([funcionarios[i] for i in dias])

#Constraints
for d in dias:
    problem += lpSum([funcionarios[(N-i+d)%N] for i in range(5)]) >= empr
```

In [ ]:
```
#Solve
result = problem.solve(PULP_CBC_CMD(msg=0))
```

In [ ]:
```
#Resultados
for v in problem.variables():
    print(v.name, "=", v.varValue)

print("FO =", value(problem.objective))

print("Current Status =", LpStatus[problem.status])
```

```
funcionarios_0 = 2.0
funcionarios_1 = 6.0
funcionarios_2 = 0.0
funcionarios_3 = 7.0
funcionarios_4 = 0.0
funcionarios_5 = 3.0
funcionarios_6 = 5.0
FO = 23.0
Current Status = Optimal
```

Problema 3: MIX DE PRODUÇÃO

Importação Bibliotecas

In [ ]:
```
from pulp import *
```

In [ ]:
```
problem = LpProblem("ProblemaMixProducao", LpMaximize)

combustivel_octanagem   = [  92,   95,  100]
combustivel_preco_venda = [ 800,  850,  900] #R$/m3
combustivel_demanda     = [ 120,   80,   40] #m3

mistura_octanagem   = [  90,  100,  110]
mistura_custo       = [ 380,  420,  450] #R$/m3
mistura_disp        = [ 120,  100,   70] #m3


combustiveis = list(range(len(combustivel_octanagem)))
misturas     = list(range(len(mistura_octanagem)))


combustivel_custo    = LpVariable.dicts("combustivel_custo", combustiveis
combustivel_producao = LpVariable.dicts("combustivel_producao", combustiv
mistura_producao     = LpVariable.dicts("mistura_producao", [(c,m) for c


#Objective function
problem += lpSum([combustivel_producao[c]*combustivel_preco_venda[c]-comb

#Constraints
for c in combustiveis:
    problem += combustivel_producao[c] <= combustivel_demanda[c]
    problem += combustivel_custo[c] == lpSum([mistura_producao[c,m]*mistu
    problem += combustivel_octanagem[c]*lpSum([mistura_producao[c,m] for
    problem += lpSum([mistura_producao[c,m] for m in misturas]) == combus


for m in misturas:
    problem += lpSum([mistura_producao[(c,m)] for c in combustiveis]) <=
```

In [ ]:
```
#Solve
result = problem.solve(PULP_CBC_CMD(msg=0))
```

In [ ]:
```
#Resultados
for v in problem.variables():
    print(v.name, "=", v.varValue)

print("FO =", value(problem.objective))

print("Current Status =", LpStatus[problem.status])
```

```
combustivel_custo_0 = 38800.0
combustivel_custo_1 = 32000.0
combustivel_custo_2 = 16800.0
combustivel_producao_0 = 100.0
combustivel_producao_1 = 80.0
combustivel_producao_2 = 40.0
mistura_producao_(0,_0) = 80.0
mistura_producao_(0,_1) = 20.0
mistura_producao_(0,_2) = 0.0
mistura_producao_(1,_0) = 40.0
mistura_producao_(1,_1) = 40.0
mistura_producao_(1,_2) = 0.0
mistura_producao_(2,_0) = 0.0
mistura_producao_(2,_1) = 40.0
mistura_producao_(2,_2) = 0.0
FO = 96400.0
Current Status = Optimal
```

Problema 4: CUSTO FIXO

Importação Bibliotecas

In [ ]:
```
from pulp import *
```

In [ ]:
```
problem = LpProblem("ProblemaMixProducao", LpMaximize)

produto_custo_fixo = [200, 150, 100]
produto_MO         = [  3,   2,   6] #h/unidade
produto_materia    = [  4,   3,   5] #unidade/produto
produto_custo      = [  6,   4,   8] #$/produto
produto_venda      = [ 12,   9,  15] #$/produto
produto_qtd_max    = [ 60,  80,  40] #$/produto

horas_MO = 300 #horas
materia  = 320 #unidades

produtos = list(range(len(produto_custo)))

produto_producao = LpVariable.dicts("produto_producao", produtos, lowBoun

#Objective function
problem += lpSum([produto_producao[p]*(produto_venda[p]-produto_custo[p])

#Constraints
for p in produtos:
    problem += produto_producao[p] <= produto_qtd_max[p]

problem += lpSum([produto_producao[p]*produto_materia[p] for p in produto
problem += lpSum([produto_producao[p]*produto_MO[p] for p in produtos]) <
```

In [ ]:
```
#Solve
result = problem.solve(PULP_CBC_CMD(msg=0))
```

In [ ]:
```
#Resultados
for v in problem.variables():
    print(v.name, "=", v.varValue)

print("FO =", value(problem.objective))

print("Current Status =", LpStatus[problem.status])
```

```
produto_producao_0 = 20.0
produto_producao_1 = 80.0
produto_producao_2 = 0.0
FO = 70.0
Current Status = Optimal
```