1ª Lista de Exercícios ¶

Aluno: Gustavo Schlieper Tessitore

In [1]:

```
import pandas as pd
import numpy as np
np.set_printoptions(precision=3, suppress=True)
import warnings
warnings.filterwarnings('ignore')


# Funções auxiliares
def describe_problem(problem):
    for variable in problem.variables():
        print(variable.name, "=", round(variable.varValue, 2))
    print(problem.objective.name, "=", round(problem.objective.value(), 2))
```

In [2]:

```python
def Simplex(T, rotulos=[], base=[]):
    """
    Função para calcular o Tableau Simplex apresentando a sua evolução ao longo das iterações.

    Argumentos de entrada:
        T: numpy array representando o Tableau inicial

    Argumentos de Saída
        T: a tabela na última iteração do algoritmo

    Programador: Prof. Dr. Rooney Coelho
    """

    print('Tableu Simplex (inicial):')
    if rotulos == [] and base == []:
        print(T)
    else:
        print( pd.DataFrame(T, columns=rotulos, index=base ) )

    menor = -1
    it = 0
    while menor < 0:
        # Inicialização dos parâmetros (sobreescrever)
        pivo_linha = -1
        pivo_coluna = -1
        pivo = 0
        menor = T[0,:-1].min()

        if menor >=0:
            print('\nNenhum dos coeficientes da linha z associados com as variáveis não básicas
 é negativo assim essa tabela é ótima!')
            break
        else:
            it += 1
            print(f'\nIteração: {it}')
            # pega o menor elemento da primeira linha (função objetivo)
            pivo_coluna = T[0,:-1].argmin()

            aux = np.zeros(len(T)-1)
            i = 0
            for a,b in zip(T[1:,-1],T[1:,pivo_coluna]):
```

```
                aux[i] = a/b
                i+=1

            val = aux[aux>0].min()
            pivo_linha = np.argwhere(aux==val).item() + 1 # Soma um para a mesma referencia incl
uindo objetivo
            pivo = T[pivo_linha, pivo_coluna]
            print(f'Linha do pivô: {pivo_linha}, Coluna do pivô: {pivo_coluna}, Elemento pivô:
{pivo}\n')

            # Nova linha do pivô = linha do pivô atual / elemento do pivô
            T[pivo_linha] = T[pivo_linha]/pivo

            # Todas as outras linhas, incluindo z
            # Nova linha=(Linha atual)-(seu coeficiente de coluna do pivô)×(Nova linha do pivô)

            nova_linha_pivo = T[pivo_linha]

            for i in range(len(T)):
                if i != pivo_linha:
                    T[i] -= T[i,pivo_coluna]*nova_linha_pivo

            print('Tableu Simplex:')
            if rotulos == [] and base == []:
                print(T)
            else:
                base[pivo_linha] = rotulos[pivo_coluna]
                print( pd.DataFrame(T, columns=rotulos, index=base ) )
    return T
```

Execicio 1

Maximizar

$$Z = 3x_1 + 6x_2$$

Sujeito a:

$$9x_1 + 8x_2 \leq 72$$

$$x_2 \leq 6$$

$$-5x_1 + 4x_2 \leq 20$$

$$2x_1 - 4x_2 \leq 20$$

$$x_1, x_2 \geq 0$$

Implementando em python usando pulp

In [3]:

```
from pulp import LpVariable, LpProblem, LpMaximize

problem1 = LpProblem("Execicio1", LpMaximize)

x1 = LpVariable("x1", lowBound=0, cat='Continuous')
x2 = LpVariable("x2", lowBound=0, cat='Continuous')

# Função objetivo
problem1 += 3 * x1 + 6 * x2
# Restrições
problem1 += 9 * x1 + 8 * x2 <= 72
problem1 += x2 <= 6
problem1 += -5 * x1 + 4 * x2 <= 20
problem1 += 2 * x1 - 4 * x2 <= 20

problem1.solve()

describe_problem(problem1)
```

```
x1 = 2.67
x2 = 6.0
OBJ = 44.0
```

Execicio 2

Maximizar

$$Z = 10x_1 + 2x_2$$

Sujeito a:

$$10x_1 + 4x_2 \leq 40$$

$$8x_1 + 2x_2 \geq 0$$

$$x_2 \leq 6$$

$$x_1 - 3x_2 \leq 0$$

$$x_1, x_2 \geq 0$$

Solução

In [4]:

```
problem2 = LpProblem("Execicio2", LpMaximize)

x1 = LpVariable("x1", lowBound=0, cat='Continuous')
x2 = LpVariable("x2", lowBound=0, cat='Continuous')

# Função objetivo
problem2 += 10 * x1 + 2 * x2

# Restrições
problem2 += 10 * x1 + 4 * x2 <= 40
problem2 += 8 * x1 + 2 * x2 >= 0
problem2 += x2 <= 6
problem2 += x1 - 3 * x2 <= 0

problem2.solve()

describe_problem(problem2)
```

```
x1 = 3.53
x2 = 1.18
OBJ = 37.65
```

Execicio 3

Uma pequena manufatura produz dois modelos, Standard e Luxo, de um certo produto. Cada unidade do modelo Standard requer 2 horas de lixação e 1 hora de polimento. Cada unidade do modelo Luxo exige 2 horas de lixação e 3 horas de polimento. A fábrica dispõe de 2 lixadoras e 3 polidoras, cada qual trabalhando 40 horas semanais. As margens de lucro são 24 e 32, respectivamente, para cada unidade Standard e Luxo. Não existem restrições de demanda para nenhum dos modelos. Elabore um modelo de programação linear que nos permita calcular a produção semanal que maximize a margem total de lucro do fabricante.

Definindo a função objetivo e as restrições:

maximixar

$$Z = 24x_1 + 32x_2$$

Sujeito a:

$$2x_1 + 2x_2 \leq 80$$

$$x_1 + 3x_2 \leq 120$$

$$x_1, x_2 \geq 0$$

Implementando em python

In [5]:

```
problem3 = LpProblem("Execicio3", LpMaximize)

x1 = LpVariable("x1", lowBound=0, cat='Continuous')
x2 = LpVariable("x2", lowBound=0, cat='Continuous')

# Função Objetivo
problem3 += 24 * x1 + 32 * x2

# Restrições
problem3 += 2 * x1 + 2 * x2 <= 80
problem3 += x1 + 3 * x2 <= 120

problem3.solve()

describe_problem(problem3)
```

```
x1 = 0.0
x2 = 40.0
OBJ = 1280.0
```

Execicio 4

Um fazendeiro dispõe de 400ha cultiváveis com milho, trigo ou soja. Cada hectare de milho exige 200 reais para preparação do terreno e 10 homens-dias de trabalho, e gera um lucro de 600 reais. Um hectare de trigo implica custos de 240 reais para preparação do terreno e 16 homens-dias de trabalho, e dá um lucro de 700 reais. Analogamente, um hectare de soja exige 140 reais e 12 homens-dias, e dá um lucro de 550 reais. O fazendeiro dispõe de 80.000 reais para cobrir os custos de trabalho e 6.000 homens-dias de mão-de-obra. Elabore um modelo de programação linear de modo a calcular a alocação de terra para os vários tipos de cultura com o objetivo de maximizar o lucro total.

maximizar

$$Z = 600x_1 + 700x_2 + 550x_3$$

Sujeito a:

$$200x_1 + 240x_2 + 140x_3 \leq 80000$$

$$10x_1 + 16x_2 + 12x_3 \leq 6000$$

$$x_1 + x_2 + x_3 \leq 400$$

$$x_1, x_2, x_3 \geq 0$$

In [6]:

```
problem4 = LpProblem("Execicio4", LpMaximize)

x1 = LpVariable("x1", lowBound=0, cat='Continuous')
x2 = LpVariable("x2", lowBound=0, cat='Continuous')
x3 = LpVariable("x3", lowBound=0, cat='Continuous')

# Função objetivo
problem4 += 600 * x1 + 700 * x2 + 550 * x3

# Restrições
problem4 += 200 * x1 + 240 * x2 + 140 * x3 <= 80000
problem4 += 10 * x1 + 16 * x2 + 12 * x3 <= 6000
problem4 += x1 + x2 + x3 <= 400

problem4.solve()

describe_problem(problem4)
```

```
x1 = 0.0
x2 = 240.0
x3 = 160.0
OBJ = 256000.0
```

Execicio 5

Encontre

$$x_1, x_2, x_3$$

pelo método Simplex.

Maximizar

$$Z = 8x_1 + 10x_2 + 6x_3$$

Sujeito a:

$$x_1 + x_3 \leq 400$$

$$4x_1 + 4x_2 + 2x_3 \leq 1200$$

$$3x_1 + 3x_2 \leq 600$$

$$x_1, x_2, x_3 \geq 0$$

Adapatando para gerar igualdades:

$$Z - 8x_1 - 10x_2 - 6x_3 = 0$$

$$x_1 + x_3 + f_1 = 400$$

$$4x_1 + 4x_2 + 2x_3 + f_2 = 1200$$

$$3x_1 + 3x_2 + f_3 = 600$$

In [7]:

```
T = np.array([
    [1,-8,-10,-6,0,0,0,0],
    [0,1,0,1,1,0,0,400],
    [0,4,4,2,0,1,0,1200],
    [0,3,3,0,0,0,1,600]
], dtype=float)

rotulos = ['z', 'x1', 'x2', 'x3', 'f1', 'f2', 'f3', 'LD' ]
base = ['z', 'f1', 'f2', 'f3']

Simplex(T, rotulos, base);
```

```
Tableu Simplex (inicial):
      z   x1    x2   x3   f1   f2   f3      LD
z   1.0 -8.0 -10.0 -6.0  0.0  0.0  0.0     0.0
f1  0.0  1.0   0.0  1.0  1.0  0.0  0.0   400.0
f2  0.0  4.0   4.0  2.0  0.0  1.0  0.0  1200.0
f3  0.0  3.0   3.0  0.0  0.0  0.0  1.0   600.0

Iteração: 1
Linha do pivô: 3, Coluna do pivô: 2, Elemento pivô: 3.0

Tableu Simplex:
      z   x1   x2   x3   f1   f2        f3      LD
z   1.0  2.0  0.0 -6.0  0.0  0.0  3.333333  2000.0
f1  0.0  1.0  0.0  1.0  1.0  0.0  0.000000   400.0
f2  0.0  0.0  0.0  2.0  0.0  1.0 -1.333333   400.0
x2  0.0  1.0  1.0  0.0  0.0  0.0  0.333333   200.0

Iteração: 2
Linha do pivô: 2, Coluna do pivô: 3, Elemento pivô: 2.0

Tableu Simplex:
      z   x1   x2   x3   f1   f2        f3      LD
z   1.0  2.0  0.0  0.0  0.0  3.0 -0.666667  3200.0
f1  0.0  1.0  0.0  0.0  1.0 -0.5  0.666667   200.0
x3  0.0  0.0  0.0  1.0  0.0  0.5 -0.666667   200.0
x2  0.0  1.0  1.0  0.0  0.0  0.0  0.333333   200.0

Iteração: 3
Linha do pivô: 1, Coluna do pivô: 6, Elemento pivô: 0.6666666666666666

Tableu Simplex:
      z   x1   x2   x3   f1    f2   f3      LD
z   1.0  3.0  0.0  0.0  1.0  2.50  0.0  3400.0
f3  0.0  1.5  0.0  0.0  1.5 -0.75  1.0   300.0
x3  0.0  1.0  0.0  1.0  1.0  0.00  0.0   400.0
x2  0.0  0.5  1.0  0.0 -0.5  0.25  0.0   100.0

Nenhum dos coeficientes da linha z associados com as variáveis não básicas é negativo assim e
ssa tabela é ótima!
```

Execicio 6

Resolva o seguinte problema pelo método das duas fases

Minimizar

$$Z = 4x_1 + 2x_2$$

Sujeito a:

$$4x_1 + 3x_2 \geq 6$$

$$6x_1 + 2x_2 = 6$$

$$2x_1 + 4x_2 \leq 6$$

$$x_1, x_2 \geq 0$$

$$r = R_1 + R_2$$

$$4x_1 + 3x_2 - x_3 + R_1 = 6$$

$$6x_1 + 2x_2 + R_2 = 6$$

$$2x_1 + 4x_2 + x_4 = 6$$

$$x_1, x_2, x_3, x_4, R_1, R_2 \geq 0$$

1ª Fase:

Phase 1 (Iter 1							
Basic	x1	x2	Sx3	Rx4	Rx5	sx6	Solution
z (min)	10,00	5,00	-1,00	0,00	0,00	0,00	12,00
Rx4	4,00	3,00	-1,00	1,00	0,00	0,00	6,00
Rx5	6,00	2,00	0,00	0,00	1,00	0,00	6,00
sx6	2,00	4,00	0,00	0,00	0,00	1,00	6,00
Lower Bound	0,00	0,00					
Upper Bound	infinity	infinity					
Unrestr'd (y/n)?	n	n					

Phase 1 (Iter 2							
Basic	x1	x2	Sx3	Rx4	Rx5	sx6	Solution
z (min)	0,00	1,67	-1,00	0,00	-1,67	0,00	2,00
Rx4	0,00	1,67	-1,00	1,00	-0,67	0,00	2,00
x1	1,00	0,33	0,00	0,00	0,17	0,00	1,00
sx6	0,00	3,33	0,00	0,00	-0,33	1,00	4,00
Lower Bound	0,00	0,00					
Upper Bound	infinity	infinity					
Unrestr'd (y/n)?	n	n					

Phase 1 (Iter 3							
Basic	x1	x2	Sx3	Rx4	Rx5	sx6	Solution
z (min)	0,00	0,00	0,00	-1,00	-1,00	0,00	0,00
x2	0,00	1,00	-0,60	0,60	-0,40	0,00	1,20
x1	1,00	0,00	0,20	-0,20	0,30	0,00	0,60
sx6	0,00	0,00	2,00	-2,00	1,00	1,00	0,00
Lower Bound	0,00	0,00					
Upper Bound	infinity	infinity					
Unrestr'd (y/n)?	n	n					

2ª Fase:

Phase 2 (Iter 4							
Basic	x1	x2	Sx3	Rx4	Rx5	sx6	Solution
z (min)	0,00	0,00	-0,40	blocked	blocked	0,00	4,80
x2	0,00	1,00	-0,60	0,60	-0,40	0,00	1,20
x1	1,00	0,00	0,20	-0,20	0,30	0,00	0,60
sx6	0,00	0,00	2,00	-2,00	1,00	1,00	0,00
Lower Bound	0,00	0,00					
Upper Bound	infinity	infinity					
Unrestr'd (y/n)?	n	n					

In [8]:

```
# Pelo Pulp
from pulp import LpMinimize

problem6 = LpProblem("Execicio6", LpMinimize)

x1 = LpVariable("x1", lowBound=0, cat='Continuous')
x2 = LpVariable("x2", lowBound=0, cat='Continuous')

# Função objetivo
problem6 += 4 * x1 + 2 * x2

# Restrições
problem6 += 4 * x1 + 3 * x2 >= 6
problem6 += 6 * x1 + 2 * x2 == 6
problem6 += 2 * x1 + 4 * x2 <= 6

problem6.solve()

describe_problem(problem6)
```

```
x1 = 0.6
x2 = 1.2
OBJ = 4.8
```

Execicio 7

Encontre o problema Dual

Minimizar

$$Z = 10x_1 + 6x_2 - 2y_3$$

Sujeito a:

$$2x_1 + 2x_2 + 2x_3 = 10$$

$$-2x_1 - x_2 - 3y_3 \geq -10$$

$$2x_1 - 4x_2 + 2x_3 \geq 4$$

$$x_1, x_2, x_3 \geq 0$$

Dual Maximizar

$$D = 10y_1 - 10y_2 + 4y_3$$

Sujeito a:

$$2y_1 - 2y_2 + 2y_3 \leq 10$$

$$2y_1 - y_2 - 4y_3 \leq 6$$

$$2y_1 - 3y_2 + 2y_3 \leq -2$$

$$y_2, y_3 \geq 0$$

Execicio 8

Um produtor de embutidos deseja fabricar duas linhas de salsichas • Econômica (>40% de carne de porco) • Premium(>60% de carne de porco) De acordo com a legislação, o máximo de amido que pode ter na salsicha é de 25%. O produtor já fechou contrato com um açougue e comprou 23 kg de carne de porco, que devem ser totalmente usados na produção. A demanda é de 350 salsichas econômicas e 500 salsichas premium. Cada unidade de salsicha produzida pesa 50g. Qual é a mistura mais econômica possível para a produção das salsichas econômica e premium?

Ingrediente	Custo(R$/kg)	Disponibilidade(kg)
Carne de porco	29,03	30
Farinha de trigo	16,53	20
Amido	12,50	17

Função objetivo Minimizar

$$Z = 29,03x_1 + 16,53x_2 + 12,50x_3 + 29,03x_4 + 16,53x_5 + 12,50x_6$$

Sujeito a:

$$x_1 \geq 7$$

$$x_4 \geq 15$$

$$x_1 + x_4 \geq 23$$

$$x_1 + x_4 \leq 30$$

$$x_3 \leq 4,375$$

$$x_6 \leq 6,25$$

$$x_3 + x_6 \leq 17$$

$$x_2 + x_5 \leq 20$$

$$x_1 + x_2 + x_3 = 17,5$$

$$x_4 + x_5 + x_6 = 25$$

$$x_1, x_2, x_3, x_4, x_5, x_6 \geq 0$$

In [9]:

```
problem8 = LpProblem("Execicio8", LpMinimize)

x1 = LpVariable("x1", lowBound=0, cat='Continuous')
x2 = LpVariable("x2", lowBound=0, cat='Continuous')
x3 = LpVariable("x3", lowBound=0, cat='Continuous')
x4 = LpVariable("x4", lowBound=0, cat='Continuous')
x5 = LpVariable("x5", lowBound=0, cat='Continuous')
x6 = LpVariable("x6", lowBound=0, cat='Continuous')

# Função objetivo
problem8 += 29.03 * x1 + 16.53 * x2 + 12.5 * x3 + 29.03 * x4 + 16.53 * x5 + 12.5 * x6

# Restrições
problem8 += x1 >= 7
problem8 += x4 >= 15
problem8 += x1 + x4 >= 23
problem8 += x1 + x4 <= 30
problem8 += x3 <= 4.375
problem8 += x6 <= 6.25
problem8 += x3 + x6 <= 17
problem8 += x2 + x5 <= 20
problem8 += x1 + x2 + x3 == 17.5
problem8 += x4 + x5 + x6 == 25

problem8.solve()

describe_problem(problem8)
```

```
x1 = 8.0
x2 = 5.12
x3 = 4.38
x4 = 15.0
x5 = 3.75
x6 = 6.25
OBJ = 947.21
```

Execicio 9

Um agricultor tem uma fazenda com 200 km², onde planeja cultivar trigo, arroz e milho. A produção esperada é de 1.800 kg por km² plantado de trigo, 2.100 kg por km² plantado de arroz e 2.900 kg por km² plantado de milho. Ele tem condições de armazenar no máximo 700.000 kg de qualquer um dos produtos. Sabendo que o trigo dá um lucro de R1,20 por kg, o arroz R0,60 e o milho R0,28, determine quantos km² de cada produto devem ser plantados para maximizar o lucro do agricultor

Função objetivo Maximizar

$$Z = 1,2 \cdot 1800x_1 + 0,6 \cdot 2100x_2 + 0,28 \cdot 2900x_3$$

Sujeito a:

$$x_1 + x_2 + x_3 \leq 200$$

$$1800x_1 + 2100x_2 + 2900x_3 \leq 700000$$

$$x_1, x_2, x_3 \geq 0$$

In [10]:

```
problem9 = LpProblem("Execicio9", LpMaximize)

x1 = LpVariable("x1", lowBound=0, cat='Continuous')
x2 = LpVariable("x2", lowBound=0, cat='Continuous')
x3 = LpVariable("x3", lowBound=0, cat='Continuous')

# Função objetivo
problem9 += 1.2 * 1800 * x1 + 0.6 * 2100 * x2 + 0.28 * 2900 * x3

# Restrições
problem9 += x1 + x2 + x3 <= 200
problem9 += 1800 * x1 + 2100 * x2 + 2900 * x3 <= 700000

problem9.solve()

describe_problem(problem9)
```

```
x1 = 200.0
x2 = 0.0
x3 = 0.0
OBJ = 432000.0
```

Execicio 10

Uma companhia produz três tipos de fertilizantes, a partir da mistura de ingredientes à base de nitrato, fosfato e potássio e de um componente inerte, conforme mostra o Quadro 1, que apresenta também os preços de venda dos fertilizantes. Dados sobre disponibilidade e custos dos ingredientes são apresentados no Quadro 2. O custo de mistura, empacotamento e promoção de vendas é estimado em 300 reais por tonelada para quaisquer produtos. A companhia tem contrato de longo prazo para fornecimento mensal de 6.500t do fertilizante A. Elabore o modelo de programação linear de modo a propor a programação da produção para o próximo mês, com o objetivo de maximizar o lucro.

Quadro 1

TIPO DE FERTILIZANTE	NITRATO	FOSFORO	POTÁSSIO	INERTE	PREÇO DE MERCADO
A	5	10	5	80	800
B	5	10	10	75	960
C	10	10	10	70	1.100

Quadro 2

INGREDIENTE	DISPONIBILIDADE	CUSTO
NITRATO	1.200	3.000
FOSFORO	2.000	1.000
POTÁSSIO	1.400	1.800
INERTE	Ilimitada	200

Função objetivo Maximizar

$$Z = 0x_1 + 80x_2 + 80x_3$$

Sujeito a:

$$x_1 \geq 6500$$

$$0,05x_1 + 0,05x_2 + 0,1x_3 \leq 1200$$

$$0,1x_1 + 0,1x_2 + 0,1x_3 \leq 2000$$

$$0,05x_1 + 0,1x_2 + 0,1x_3 \leq 1400$$

$$x_1, x_2, x_3 \geq 0$$

In [11]:

```
problem10 = LpProblem("Execicio10", LpMaximize)

x1 = LpVariable("x1", lowBound=6500, cat='Continuous')
x2 = LpVariable("x2", lowBound=0, cat='Continuous')
x3 = LpVariable("x3", lowBound=0, cat='Continuous')

# Função objetivo
problem10 += 0 * x1 + 80 * x2 + 80 * x3

# Restrições
problem10 += 0.05 * x1 + 0.05 * x2 + 0.1 * x3 <= 1200
problem10 += 0.1 * x1 + 0.1 * x2 + 0.1 * x3 <= 2000
problem10 += 0.05 * x1 + 0.1 * x2 + 0.1 * x3 <= 1400

problem10.solve()

describe_problem(problem10)
```

```
x1 = 6500.0
x2 = 4000.0
x3 = 6750.0
OBJ = 860000.0
```

Execicio 11

Uma indústria fabrica dois tipos de produtos, A e B, e assinou um contrato para o fornecimento de 30.000 produtos tipo A e 15.000 tipo B. A indústria tem 3 setores para a fabricação dos produtos: produção, montagem e teste de qualidade, A tabela a seguir mostra as horas utilizadas para cada produto e a disponibilidade de cadasetor da indústria

	A	B	Horas
Produção	0,2	0,4	10.000
Montagem	0,3	0,5	15.000
Teste de Qualidade	0,1	0,1	5.000

O custo unitário de fabricação dos produtos A e B é de R 55,00 e R 85,00, respectivamente. A indústria tem também a opção de terceirizar a produção desses produtos por R 67,00 e R 95,00, respectivamente. Para minimizar os custos, quantas unidades dos produtos ela deve fabricar e quantas deve terceirizar para honrar o seu contrato?

Função objetivo Minimizar

$$Z = 55x_1 + 85x_2 + 67x_3 + 95x_4$$

Sujeito a:

$$x_1 + x_2 \geq 30.000$$

$$x_3 + x_4 \geq 15.000$$

$$0,2x_1 + 0,4x_3 \leq 10.000$$

$$0,3x_1 + 0,5x_3 \leq 15.000$$

$$0,1x_1 + 0,1x_3 \leq 5.000$$

$$x_1, x_2, x_3, x_4 \geq 0$$

In [12]:

```
problem11 = LpProblem("Execicio11", LpMinimize)

x1 = LpVariable("x1", lowBound=0, cat='Continuous')
x2 = LpVariable("x2", lowBound=0, cat='Continuous')
x3 = LpVariable("x3", lowBound=0, cat='Continuous')
x4 = LpVariable("x4", lowBound=0, cat='Continuous')

# Função objetivo
problem11 += 55 * x1 + 85 * x2 + 67 * x3 + 95 * x4

# Restrições
problem11 += x1 + x2 >= 30000
problem11 += x3 + x4 >= 15000
problem11 += 0.2 * x1 + 0.4 * x3 <= 10000
problem11 += 0.3 * x1 + 0.5 * x3 <= 15000
problem11 += 0.1 * x1 + 0.1 * x3 <= 5000

problem11.solve()

describe_problem(problem11)
```

```
x1 = 30000.0
x2 = 0.0
x3 = 10000.0
x4 = 5000.0
OBJ = 2795000.0
```

Execicio 12

Uma pequena fábrica de laticínios recebe por dia 8.000 litros de leite que são utilizados na fabricação de queijo, doce de leite e ricota. A ricota é subproduto do queijo, já que é feita com o soro que sobra da fabricação deste. Cada 3 quilos de queijo geram soro suficiente para se fazer no máximo 1 kg de ricota. Para fazer 1 kg de queijo, o laticínio gasta 10 litros de leite. Para se fazer 1kg de doce, gastam-se 6 litros de leite. Além dessas, deve-se obedecer a duas outras restrições de mercado

a) A quantidade de doce por dia não deve ultrapassar 200 kg. b) A quantidade de queijo deve ser no mínimo igual a 3 vezes a quantidade de doce.

Produto	Lucro unitário	Mão de obra (min)
Queijo	1,50	3
Doce	2,00	2
Ricota	1,20	1

A fábrica dispõe de 12 empregados que trabalham 8 horas por dia. Em todo o processo, desde o recebimento do leite, a pasteurização, a produção, a embalagem, o armazenamento e o despacho, os produtos requerem a quantidade de mão-de-obra mostrada na tabela acima. A tabela apresenta também os lucros unitários de cada produto

Função objetivo

$$Z = 1,5x_1 + 2x_2 + 1,2x_3$$

Sujeito a:

$$3x_1 + 2x_2 + x_3 \leq 5760$$

$$10x_1 + 6x_2 \leq 8000$$

$$x_2 \leq 200$$

$$x_1 \geq 3x_2$$

$$3x_3 \leq x_1$$

$$x_1, x_2, x_3 \geq 0$$

In [13]:

```
problem12 = LpProblem("Execicio12", LpMaximize)

x1 = LpVariable("x1", lowBound=0, cat='Continuous')
x2 = LpVariable("x2", lowBound=0, upBound=200, cat='Continuous')
x3 = LpVariable("x3", lowBound=0, cat='Continuous')

# Função objetivo
problem12 += 1.5 * x1 + 2 * x2 + 1.2 * x3

# Restrições
problem12 += 3 * x1 + 2 * x2 + x3 <= 5760
problem12 += 10 * x1 + 6 * x2 <= 8000
problem12 += x1 >= 3 * x2
problem12 += 3 * x3 <= x1

problem12.solve()

describe_problem(problem12)
```

```
x1 = 680.0
x2 = 200.0
x3 = 226.67
OBJ = 1692.0
```