

# Assignment 6 - Exercise 2a

Bruno Kiyoshi Ynumaru - 201805995

```
In [1]: prob_statement = """A empresa Schwabe utiliza ouro e prata para produzir dois tipos
ouro, 3g de prata e 1 hora de trabalho para confecção. O colar 2 requer 3g de ouro
de trabalho. Cada colar 1 vende por $400, e cada colar 2 vende por $500. Todos os co
são vendidos. Em estoque a empresa tem 100g de ouro e 120g de prata, e dispõe de 7
obra. Ouro extra pode ser comprado por um custo de $100/g. Encomendas já realizadas
sejam produzidos pelo menos 20 colares 1 e 25 colares 2. Formule um modelo de progra
para maximizar o lucro da empresa.

.
"""
def fix_statement(str_statement):
    list_replacements = [(".", "ç"),
                          ("´a", "á"),
                          ("´e", "é"),
                          ("´i", "í"),
                          ("´o", "ó"),
                          ("´u", "ú"),
                          ("~a", "ã"),
                          ("~o", "õ"),
                          ("$", "\\$")]
    for replacement in list_replacements:
        str_statement = str_statement.replace(replacement[0], replacement[1])
    return str_statement
prob_statement = fix_statement(prob_statement)
print(prob_statement)
```

A empresa Schwabe utiliza ouro e prata para produzir dois tipos de colares. O colar 1 requer 2g de ouro, 3g de prata e 1 hora de trabalho para confecção. O colar 2 requer 3g de ouro, 2g de prata e 2 horas de trabalho. Cada colar 1 vende por \\\$400, e cada colar 2 vende por \\\$500. Todos os colares produzidos são vendidos. Em estoque a empresa tem 100g de ouro e 120g de prata, e dispõe de 70 horas de mão de obra. Ouro extra pode ser comprado por um custo de \\\$100/g. Encomendas já realizadas s requerem que sejam produzidos pelo menos 20 colares 1 e 25 colares 2. Formule um modelo de programação linear para maximizar o lucro da empresa.

A empresa Schwabe utiliza ouro e prata para produzir dois tipos de colares. O colar 1 requer 2g de ouro, 3g de prata e 1 hora de trabalho para confecção. O colar 2 requer 3g de ouro, 2g de prata e 2 horas de trabalho. Cada colar 1 vende por \\$400, e cada colar 2 vende por \\$500. Todos os colares produzidos são vendidos. Em estoque a empresa tem 100g de ouro e 120g de prata, e dispõe de 70 horas de mão de obra. Ouro extra pode ser comprado por um custo de \\$100/g. Encomendas já realizadas requerem que sejam produzidos pelo menos 20 colares 1 e 25 colares 2. Formule um modelo de programação linear para maximizar o lucro da empresa. .

```
In [1]: import gurobipy as gp
from gurobipy import GRB, Model
```

```
In [2]: # Create a new model
m = Model("Schwabe")
```

Restricted license - for non-production use only - expires 2022-01-13

```
In [3]: # Create variables
x1 = m.addVar(lb=0, vtype=GRB.CONTINUOUS, name="no. collars type 1")
x2 = m.addVar(lb=0, vtype=GRB.CONTINUOUS, name="no. collars type 2")
g = m.addVar(lb=0, vtype=GRB.CONTINUOUS, name="grams of purchased gold")
```

```
In [4]: m.setObjective(x1 * 400 + x2 * 500 - g * 100,
                      GRB.MAXIMIZE)
```

```
In [5]: # Add constraints
m.addConstr(x1 * 2 + x2 * 3 <= 100 + g, 'Gold constraint')
m.addConstr(x1 * 3 + x2 * 2 <= 120, 'Silver constraint')
m.addConstr(x1 * 1 + x2 * 2 <= 70, 'MH constraint')
m.addConstr(x1 >= 20, 'Min production of collar type 1')
m.addConstr(x2 >= 25, 'Min production of collar type 2')

m.optimize()
```

Gurobi Optimizer version 9.1.1 build v9.1.1rc0 (win64)  
 Thread count: 4 physical cores, 8 logical processors, using up to 8 threads  
 Optimize a model with 5 rows, 3 columns and 9 nonzeros  
 Model fingerprint: 0x9b865689  
 Coefficient statistics:  
   Matrix range       [1e+00, 3e+00]  
   Objective range    [1e+02, 5e+02]  
   Bounds range       [0e+00, 0e+00]  
   RHS range          [2e+01, 1e+02]  
 Presolve removed 5 rows and 3 columns  
 Presolve time: 0.00s  
 Presolve: All rows and columns removed  

Iteration	Objective	Primal Inf.	Dual Inf.	Time
0	1.9000000e+04	0.000000e+00	0.000000e+00	0s

  
 Solved in 0 iterations and 0.01 seconds  
 Optimal objective 1.900000000e+04

```
In [6]: for v in m.getVars():
        print(f'{v.varName}, {v.x}')

        print(f'Obj: {m.objVal}')
```

```
no. collars type 1, 20.0
no. collars type 2, 25.0
grams of purchased gold, 15.0
Obj: 19000.0
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```