Aula 03:

Conceitos Básicos e Modelagem

Programação Linear e Inteira

Túlio Toffolo

www.toffolo.com.br

Departamento de Computação

Universidade Federal de Ouro Preto

Conceitos Básicos e Modelagem

1. Forma Padrão
2. Soluções Básicas
3. Dicas de Modelagem
4. Implementando modelos

Conceitos Básicos e Modelagem

Modelo de exemplo

Em aulas anteriores modelamos o problema de decisão de quantos sacos de soja (x_1) e milho (x_2) comprar:

$$\begin{aligned} \text{maximizar} \quad & 300x_1 + 280x_2 && (1) \\ \text{sujeito a} \quad & 70x_1 + 50x_2 \leq 350 && (2) \\ & 50x_1 + 80x_2 \leq 400 && (3) \\ & x_1 \leq 4 && (4) \\ & x_1, x_2 \geq 0 && (5) \end{aligned}$$

Notações utilizadas

Um modelo de PL pode ser escrito como:

$$\begin{aligned} \text{maximizar} \quad & \mathbf{c}^t\mathbf{x} \\ \text{sujeito a} \quad & \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ & \mathbf{x} \geq 0 \end{aligned}$$

No exemplo anterior teríamos:

$$\mathbf{A} = \begin{bmatrix} 70 & 50 \\ 50 & 80 \\ 1 & 0 \end{bmatrix}$$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 350 \\ 400 \\ 4 \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} 300 \\ 280 \end{bmatrix}$$

Forma Padrão

Um Programa Linear (PL) está na *forma padrão* se:

1. todas as restrições são de igualdade;
2. todas as variáveis são não negativas (≥ 0).

$$\begin{aligned} \text{minimizar} \quad & \mathbf{c}^t\mathbf{x} \\ \text{sujeito a} \quad & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq 0 \end{aligned}$$

- Opcional: transformar o PL em um PL de minimização
  - $\max \mathbf{c}^t\mathbf{x}$ equivale a $\min -\mathbf{c}^t\mathbf{x}$

Forma Padrão

O modelo a seguir está na forma padrão?

$$
\begin{array}{rllll}
\text{Minimize:} & 3x_1 + & 2,5x_2 & & \\
\text{Sujeito a:} & 8x_1 + & 4x_2 & \geq & 32 \\
& 6x_1 + & 6x_2 & \geq & 36 \\
& x_1 & & \geq & 0 \\
& & x_2 & \geq & 0
\end{array}
$$

Não! Mas pode ser facilmente convertido para a **forma padrão**.

Forma Padrão

É possível converter qualquer PL para a forma padrão:

1. Restrições de $\leq$ e $\geq$ se tornam restrições de $=$ por meio da introdução de variáveis de folga.
   - Variáveis de folga indicam *falta* ou *excesso*.

2. Variáveis que podem ser negativas são substituídas por duas variáveis, uma indicando a *parte positiva* e outra a parte *negativa* da mesma.

3. Problemas de maximização podem ser convertidos em problemas de minimização: $\max f(x) \rightarrow \min -f(x)$.

Convertendo restrições...

$$3x_1 + 2x_2 \geq 6$$
$$\Downarrow$$
$$3x_1 + 2x_2 = 6 + s_1$$
$$\Downarrow$$
$$3x_1 + 2x_2 - s_1 = 6$$
$$\vdots$$
$$50x_1 + 35x_3 \leq 80$$
$$\Downarrow$$
$$50x_1 + 35x_3 = 80 - s_3$$
$$\Downarrow$$
$$50x_1 + 35x_3 + s_3 = 80$$

Convertendo variáveis...

Seja x_1 uma variável **livre** que pode assumir valores negativos.

- Substituímos x_1 por duas variáveis, x_1^+ e x_1^-
  - x_1^+ representa a parte positiva de x_1
  - x_1^- representa a parte negativa de x_1
- Fazemos $x_1 = x_1^+ - x_1^-$, com
  - $x_1^+ \geq 0$ e $x_1^- \geq 0$

Suponha que x_1 estava na seguinte restrição :

$$3x_1 + 2x_2 \geq 6$$
$$\Downarrow$$
$$3x_1^+ - 3x_1^- + 2x_2 \geq 6$$

Forma Padrão

As restrições de um PL pode ser escrito como:

$$\mathbf{Ax} = \mathbf{b}$$

Teremos, assim:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

Exemplo 1

Problema da dieta:

$$\begin{aligned}
\text{min.} \quad & 3x_1 + 2,5x_2 \\
\text{s.a.} \quad & 8x_1 + 4x_2 \geq 32 \\
& 6x_1 + 6x_2 \geq 36 \\
& x_1, \quad x_2 \geq 0
\end{aligned}$$

Na forma padrão:

$$\begin{aligned}
\text{min.} \quad & 3x_1 + 2,5x_2 \\
\text{s.a.} \quad & 8x_1 + 4x_2 - s_1 = 32 \\
& 6x_1 + 6x_2 - s_2 = 36 \\
& x_1, \quad x_2, \quad s_1, \quad s_2 \geq 0
\end{aligned}$$

Exemplo 2

$$
\begin{array}{rl}
\text{max.} & 300x_1 + 280x_2 \\
\text{s.a.} & 70x_1 + 50x_2 \leq 350 \\
& 50x_1 + 80x_2 \leq 400 \\
& x_1 \leq 4 \\
& x_1, \quad x_2 \geq 0
\end{array}
$$

Na forma padrão:

$$
\begin{array}{rl}
\text{min.} & -300x_1 - 280x_2 \\
\text{s.a.} & 70x_1 + 50x_2 + s_1 = 350 \\
& 50x_1 + 80x_2 + s_2 = 400 \\
& x_1 + s_3 = 4 \\
& x_1, \quad x_2, \quad s_1, \quad s_2, \quad s_3 \geq 0
\end{array}
$$

Forma Padrão

Exercício

Coloque na forma padrão:

$$\begin{array}{rrcl} min. & 3x_1+x_2 & & \\ s.a. & x_1 \quad\quad & \geq & 3 \\ & x_1+x_2 & \leq & 4 \\ & 2x_1-x_2 & = & 3 \\ & x_1 \quad\quad & \geq & 0 \\ & x_2 & \in & \mathbb{R} \ \ \text{(variável livre)} \end{array}$$

Conceitos Básicos e Modelagem

Variáveis Básicas

Definição

Em um sistema de equações com n variáveis e m restrições definimos como **solução básica** uma solução onde temos:

- m — variáveis para as quais o sistema é resolvido, essas são chamadas **Variáveis Básicas (*VB*)**
- $n - m$ — o restante das variáveis permanece fixada em zero - as **Variáveis Não-Básicas (*VNB*)**

Exemplo

$$\begin{array}{cccc} x_1 & +x_2 & & = & 3 \\ & -x_2 & +x_3 & = & -1 \end{array}$$

Verifique as soluções para $VB = \{x_1, x_2\}$ e $VB = \{x_2, x_3\}$

Variáveis Básicas

Todo conjunto de VB permite a obtenção de uma solução básica ?

$$\begin{array}{rrrcl} x_1 & +2x_2 & +x_3 & = & 1 \\ 2x_1 & +4x_2 & +x_3 & = & 3 \end{array}$$

- Tente VB $= \{x_1, x_2\}$....

Pontos Extremos e Soluções Básicas Factíveis

Definição

Qualquer *solução básica* onde todas as variáveis são *não negativas* é uma ***Solução Básica Factível* - SBF**.

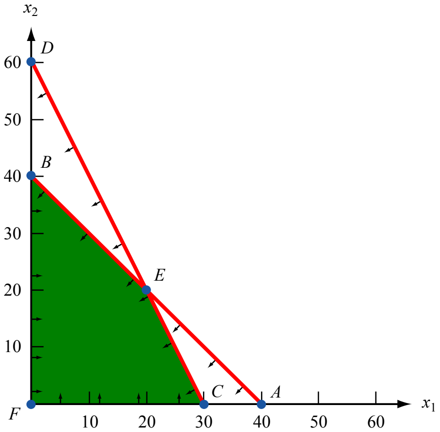
Teorema

Um ponto na região factível de um PL é um ***Ponto Extremo*** se e somente se é uma ***Solução Básica Factível*** para o PL.

$$\begin{array}{rl} max. & 4x_1+3x_2 \\ s.a. & x_1 \quad +x_2 \leq 40 \\ & 2x_1 \quad +x_2 \leq 60 \\ & x_1, x_2 \geq \quad 0 \end{array}$$

$$\Downarrow$$

$$\begin{array}{rl} max. & 4x_1+3x_2 \\ s.a. & x_1 \quad +x_2 + x_3 \qquad = 40 \\ & 2x_1 \quad +x_2 \qquad +x_4 = 60 \\ & x_1, x_2, x_3, x_4 \geq \quad 0 \end{array}$$



Ex.: VB $= \{x_1, x_3\}$ corresponde a qual ponto?
Qual o valor de x_2 e x_4 na SBF desse ponto?

PLs Degenerados

Degeneração

Eventualmente, mais de um Conjunto de Variáveis Básicas pode corresponder a um mesmo Ponto Extremo. Nesse caso dizemos que o Programa Linear é Degenerado.

O impacto de soluções degeneradas na resolução dos PLs será discutido posteriormente.

Direção Ilimitada

Definição

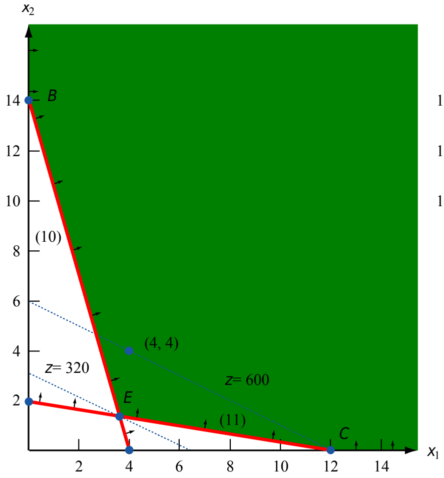
Em um PL com região factível S e restrições $Ax = b$, $x \geq 0$, dizemos que d é uma Direção Ilimitada se para qualquer solução $x \in S$ e qualquer $c \geq 0$:

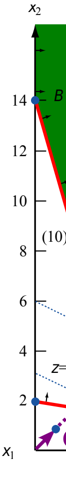
$$x + c\, d \in S$$

Direção Ilimitada

$$
\begin{array}{lllll}
\min & 50x_1 + 100x_2 & & & \\
s.a. & 7x_1 + & 2x_2 - s_1 & & = 28 \\
 & 2x_1 + & 12x_2 & - s_2 & = 24 \\
 & x_1, & x_2, \ s_1, & s_2 & \geq 0
\end{array}
$$

$$
\mathbf{d} = \begin{bmatrix} 1 \\ 1 \\ 9 \\ 14 \end{bmatrix}
$$





Pontos Extremos e Soluções Factíveis

Teorema:
Considere um PL na forma padrão com SBFs:

$$b_1, b_2, \ldots, b_k$$

Qualquer ponto x na região factível pode ser escrito como:

$$x = d + \sum_{i=1}^{k} \sigma_i b_i$$

em que $d = 0$ ou é a direção ilimitada e $\sum_{i=1}^{k} \sigma_i = 1$.

Combinações Convexas de SBFs (PL Limitado)

$$\begin{array}{rcccc} max: & 4x_1 & +3x_2 & & \\ s.a.: & x_1 & +x_2 & \leq & 40 \\ & 2x_1 & +x_2 & \leq & 60 \\ & & x_1, x_2 & \geq & 0 \end{array}$$

Ponto H (24,12) não é SBF.

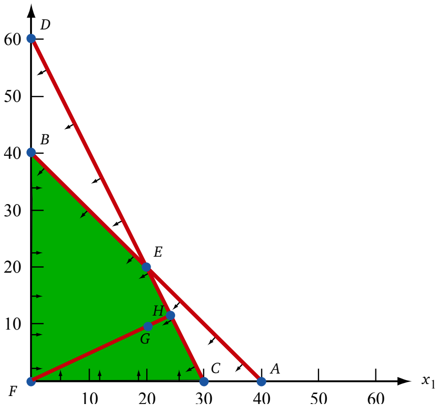
Pode ser escrito como a combinação convexa de E e C:

$$H = 0{,}6\,E + 0{,}4\,C$$

Ponto G também não é BFS.

Pode ser escrito como :

$$G = \tfrac{1}{6}F + \tfrac{5}{6}H$$
$$\Downarrow$$
$$G = \tfrac{1}{6}F + \tfrac{5}{6}(0,6E + 0,4C)$$



Combinações Convexas de SBFs (PL Ilimitado)

$$\begin{array}{lllll} 7x_1 & +2x_2 & -x_3 & & = 28 \\ 2x_1 & +12x_2 & & -x_4 & = 24 \end{array}$$

Descrevendo F em termos de SBFs:

Direção Ilimitada:

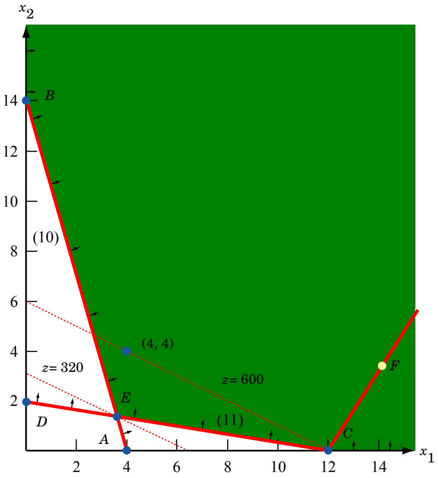
Inclinação para ir de C a F: $\frac{4-0}{14-12}$

$$\mathbf{d} = \begin{bmatrix} 2 \\ 4 \\ 22 \\ 52 \end{bmatrix}$$

$$\mathbf{b_1} = \begin{bmatrix} 12 \\ 0 \\ 56 \\ 0 \end{bmatrix} \qquad \mathbf{x} = \begin{bmatrix} 14 \\ 4 \\ 78 \\ 52 \end{bmatrix}$$

Ou seja, temos:

$$\mathbf{x} = \mathbf{d} + \mathbf{b_1}$$



Conceitos Básicos e Modelagem

1. Forma Padrão
2. Soluções Básicas
3. Dicas de Modelagem
   - Exemplo 1: Fábrica de brinquedos
   - Exemplo 2: Transporte
4. Implementando modelos

Passo-a-passo para modelar um problema:

1. Elabore um **esquema** do problema.
2. Encontre e escreva uma **solução** qualquer para o problema.
3. Olhando para a solução, defina as **variáveis** de decisão.
4. Observando as variáveis de decisão, defina a **função objetivo**, ou seja, o que deve ser maximizado/minimizado.
5. Finalmente, defina as **restrições** do problema.

Exemplo 1: Fábrica de brinquedos

Fonte: [1] Luís Herique Rodrigues *et* al. (2014), *Pesquisa Operacional - Programação Linear Passo a Passo*, Editora Unisinos.

Uma pequena oficina de brinquedos produz dois tipos de brinquedos: caminhão de madeira e boneca de pano. O lucro do caminhão é de R$ 10,00 por unidade e da boneca de pano é de R$ 8,00 por unidade. São necessárias seis pessoas para fazer um lote de dez caminhões por dia e quatro pessoas para fazer um lote de 14 bonecas por dia. Há 18 pessoas disponíveis para produzir os itens, podendo ser alocadas em qualquer um dos dois, em qualquer etapa. Devido à demanda existente é necessário fazer ao menos um lote de caminhões e um lote de bonecas por dia.

Formule um modelo de Programação Linear que maximize a lucratividade diária.

Exemplo 1 - Esquema

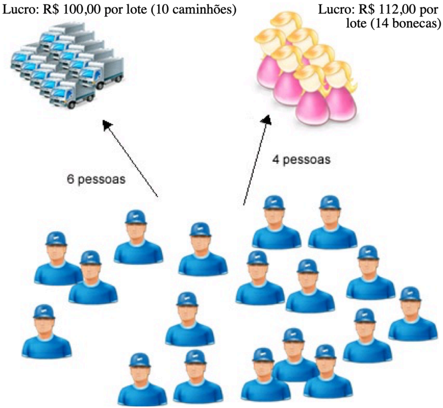


Figura: Esquema (imagem extraída de Rodrigues *et.* al [1])

Exemplo 1 - Solução

Qual seria uma solução para o problema?

- Produzir **1 lote de caminhões** e **1 lote de bonecas**?
  - Utilizaríamos neste caso 10 pessoas (6 + 4)
  - Lucraríamos R$ 212,00 (R$ 100,00 + R$ 112,00)
  - Solução parece ok... não utilizamos mais de 18 pessoas e atendemos à demanda mínima de fazer ao menos um lote de caminhões e um lote de bonecas.
  - **Solução viável!**

Exemplo 1 - Variáveis

Solução: **1 lote de caminhões** e **1 lote de bonecas**!

Qual decisão foi tomada?

- Número de lotes de caminhões: x_1
- Número de lotes de bonecas: x_2

Exemplo 1 - Função objetivo

O que estamos minimizando/maximizando?

- Maximizando o lucro!
- Ou seja: $\max 100x_1 + 112x_2$

Exemplo 1 - Restrições

Quais são as restrições?

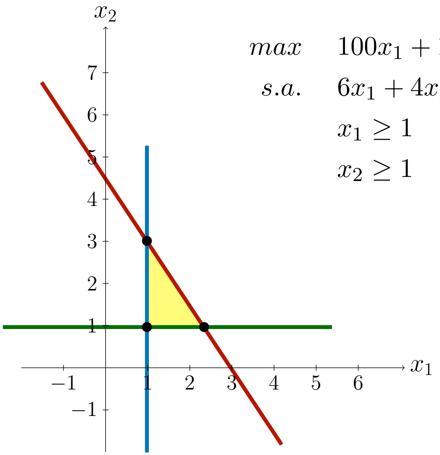
1. Há 18 pessoas disponíveis:
   - $6x_1 + 4x_2 \leq 18$
2. Devemos produzir no mínimo 1 lote de caminhões:
   - $x_1 \geq 1$
3. Devemos produzir no mínimo 1 lote de bonecas:
   - $x_2 \geq 1$

Exemplo 1 - Modelo

Logo, devemos resolver o modelo de programação linear a seguir:

$$
\begin{aligned}
max \quad & 100x_1 + 112x_2 \\
s.a. \quad & 6x_1 + 4x_2 \leq 18 \\
& x_1 \geq 1 \\
& x_2 \geq 1
\end{aligned}
$$

O modelo pode ser facilmente resolvido pelo **método gráfico**:



$$
\begin{aligned}
max \quad & 100x_1 + 112x_2 \\
s.a. \quad & 6x_1 + 4x_2 \leq 18 \\
& x_1 \geq 1 \\
& x_2 \geq 1
\end{aligned}
$$

Exemplo 2: Transporte

Fonte: [1] Luís Herique Rodrigues *et* al. (2014), *Pesquisa Operacional - Programação Linear Passo a Passo*, Editora Unisinos.

Você possui três fábricas localizadas em regiões geográficas distintas, e precisa saber quanto deve produzir e transportar para quatro diferentes mercados a um custo mínimo. As informações do custo de transporte unitário entre as fábricas e os mercados estão no quadro a seguir.

Custo de transporte		Mercados				Capacidade Produtiva
		1	2	3	4	
Fábricas	A	$ 0,90/un	$ 1,00/un	$ 1,80/un	$ 1,05/un	22.500 un
	B	$ 2,10/un	$ 0,80/un	$ 0,70/un	$ 1,15/un	21.000 un
	C	$ 1,10/un	$ 1,00/un	$ 1,20/un	$ 1,50/un	19.500 un
Demanda mínima		10.000un	15.000un	11.000un	10.000un	

Exemplo 2 - Esquema

Deve-se definir quantos produtos de cada fábrica devem ser enviados a cada mercado, considerando capacidade e demanda mínima...

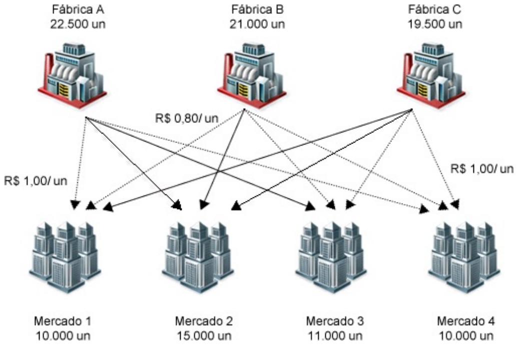


Figura: Esquema (imagem extraída de Rodrigues *et.* al [1])

Exemplo 2 - Solução

Uma solução deve indicar quantas unidades de produto devem ser enviadas de cada fábrica para cada mercado, respeitando as capacidades produtivas e atendendo as demandas mínimas de cada mercado.

Eis uma solução possível:

Unidades enviadas		*Para o mercado...*			
		1	*2*	*3*	*4*
Da Fábrica...	*A*	5000	5000	5000	5000
	B	5000	5000	5000	5000
	C	4000	5000	4000	4000

Exemplo 2 - Variáveis

Estamos decidindo quanto enviar de cada fábrica para cada mercado...

- Seja F o conjunto de fábricas
- Seja M o conjunto de mercados

Variáveis:

- $x_{i,j}$: quantidade a enviar da fábrica $i \in F$ para o mercado $j \in M$.

Exemplo 2 - Função objetivo

Devemos **minimizar** o custo de transporte...

- Seja $c_{i,j}$ o custo de enviar uma unidade do produto produzido na fábrica $i \in F$ para o mercado $j \in M$

Função objetivo:

- $\min \sum_{i \in F} \sum_{j \in M} c_{i,j} x_{i,j}$

Exemplo 2 - Restrições

Custo de transporte		Mercados				Capacidade Produtiva
		1	2	3	4	
Fábricas	A	$ 0,90/un	$ 1,00/un	$ 1,80/un	$ 1,05/un	22.500 un
	B	$ 2,10/un	$ 0,80/un	$ 0,70/un	$ 1,15/un	21.000 un
	C	$ 1,10/un	$ 1,00/un	$ 1,20/un	$ 1,50/un	19.500 un
Demanda mínima		10.000un	15.000un	11.000un	10.000un	

Seja p_i a capacidade produtiva da fábrica $i \in F$

- $\forall i \in F : \sum_{j \in M} x_{i,j} \leq p_i$

Seja d_j a demanda mínima do mercado $j \in M$

- $\forall j \in M : \sum_{i \in F} x_{i,j} \geq d_j$

Exemplo 2 - Modelo

Eis o modelo de programação linear:

$$
\begin{aligned}
min \quad & \sum_{i \in F} \sum_{j \in M} c_{i,j} x_{i,j} & \\
s.a. \quad & \sum_{j \in M} x_{i,j} \leq p_i & \forall i \in F \\
& \sum_{i \in F} x_{i,j} \geq d_j & \forall j \in M \\
& x_{i,j} \geq 0 & \forall i \in F, \forall j \in M
\end{aligned}
$$

Conceitos Básicos e Modelagem

Python-MIP

Vamos utilizar o Python-MIP para implementar os dois modelos vistos nesta aula.

Requisitos:

- Linguagem **Python 3.7** ou mais recente
- Framework **Python-MIP**, desenvolvido por Túlio Toffolo e Haroldo Santos (projeto apoiado pela COIN-OR)

Mais informações: *http://www.python-mip.com*

Exemplo 1:

```python
from mip.model import *

model = Model("Exemplo 1", MAXIMIZE)

# criando variáveis
x1 = model.add_var()
x2 = model.add_var()

# criando a função objetivo
model += 100*x1 + 112*x2

# adicionando as restrições
model += 6*x1 + 4*x2 <= 18
model += x1 >= 1
model += x2 >= 1

# resolvendo o modelo
model.optimize()

# imprimindo a solução
print(f"x1 = {x1.x}, x2 = {x2.x}")
```

Exemplo 1 (usando add_constr):

```python
from mip.model import *

model = Model("Exemplo 1", MAXIMIZE)

# criando variáveis
x1 = model.add_var()
x2 = model.add_var()

# criando a função objetivo
model.objective = 100*x1 + 112*x2

# adicionando as restrições
model.add_constr(6*x1 + 4*x2 <= 18)
model.add_constr(x1 >= 1)
model.add_constr(x2 >= 1)

# resolvendo o modelo
model.optimize()

# imprimindo a solução
print(f"x1 = {x1.x}, x2 = {x2.x}")
```

Exemplo 2:

```python
from mip.model import *

model = Model("Exemplo 2")

# dados de entrada
F = [ 0, 1, 2 ]
M = [ 0, 1, 2, 3 ]
c = [ [ 0.90, 1.00, 1.80, 1.05 ],
      [ 2.10, 0.80, 0.70, 1.15 ],
      [ 1.10, 1.00, 1.20, 1.50 ]  ]
p = [ 22500, 21000, 19500 ]
d = [ 10000, 15000, 11000, 10000 ]

# criando variáveis
x = [ [ model.add_var() for j in M ] for i in F ]

# criando a função objetivo
model += xsum(c[i][j]*x[i][j] for i in F for j in M)
```

Exemplo 2 (continuação):

```python
# adicionando as restrições
for i in F:
    model += xsum(x[i][j] for j in M) <= p[i]
for j in M:
    model += xsum(x[i][j] for i in F) >= d[j]

# resolvendo o modelo
model.optimize()

# imprimindo a solução
print("\n----------------------------------")
print(f"Solução ótima com custo: {model.objective_value}\n")
for i in F:
    for j in M:
        print(f"x({i},{j}) = {x[i][j].x}")
print("----------------------------------\n")
```

```
tuliotoffolo — -bash
Using Python-MIP package version 1.3.2
gurobi version 8.0 found
Academic license - for non-commercial use only
Optimize a model with 7 rows, 12 columns and 24 nonzeros
Coefficient statistics:
  Matrix range     [1e+00, 1e+00]
  Objective range  [7e-01, 2e+00]
  Bounds range     [0e+00, 0e+00]
  RHS range        [1e+04, 2e+04]
Presolve time: 0.00s
Presolved: 7 rows, 12 columns, 24 nonzeros

Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    0.0000000e+00   4.600000e+04   0.000000e+00      0s
       6    4.0200000e+04   0.000000e+00   0.000000e+00      0s

Solved in 6 iterations and 0.00 seconds
Optimal objective  4.020000000e+04

-----------------------------------
Solução ótima com custo: 40200.0

x(0,0) = 10000.0
x(0,1) = 2500.0
x(0,2) = 0.0
x(0,3) = 10000.0
x(1,0) = 0.0
x(1,1) = 10000.0
x(1,2) = 11000.0
x(1,3) = 0.0
x(2,0) = 0.0
x(2,1) = 2500.0
x(2,2) = 0.0
x(2,3) = 0.0
-----------------------------------
```

Perguntas?