



Universidade Federal de São Carlos  
Departamento de Engenharia de Produção



# Otimização Linear Contínua e Discreta (Tópicos Avançados em PCSP)

PPGEP, UFSCar - Semestre 01/2022  
Prof. Dr. Pedro Munari (munari@dep.ufscar.br)

Tópico 1.2: Introdução à Otimização e Programação Linear

## Objetivos deste tópico

- ▶ Estudar os conceitos básicos de otimização e suas aplicações;
- ▶ Compreender o que é um modelo de otimização e como modelar um problema usando a otimização;
- ▶ Conhecer a programação linear, seus conceitos básicos e alguns modelos clássicos;
- ▶ Aprender a resolver modelos de programação linear por meio de *softwares*, usando o `lp_solve` e o suplemento *Solver* do Excel.

# Otimização

## ▷ Definição

Matematicamente, um problema de otimização é definido como

minimizar  $f(x)$  sujeito a  $x \in \mathcal{X}$ .

# Otimização

## ▷ Definição

Matematicamente, um problema de otimização é definido como

minimizar  $f(x)$  sujeito a  $x \in \mathcal{X}$ .  
(maximizar)

# Otimização

## ▷ Definição

Matematicamente, um problema de otimização é definido como

$$\begin{array}{l} \text{minimizar } f(x) \text{ sujeito a } x \in \mathcal{X}. \\ \text{(maximizar)} \end{array}$$

- ▶  $x$ : variável de decisão;

# Otimização

## ▷ Definição

Matematicamente, um problema de otimização é definido como

$$\begin{array}{l} \text{minimizar } f(x) \text{ sujeito a } x \in \mathcal{X}. \\ \text{(maximizar)} \end{array}$$

- ▶  $x$ : variável de decisão; em geral:  $x = (x_1, x_2, \dots, x_n)$ .

# Otimização

## ▷ Definição

Matematicamente, um problema de otimização é definido como

$$\begin{array}{l} \text{minimizar } f(x) \text{ sujeito a } x \in \mathcal{X}. \\ \text{(maximizar)} \end{array}$$

- ▶  $x$ : variável de decisão; em geral:  $x = (x_1, x_2, \dots, x_n)$ .
- ▶  $\mathcal{X}$ : conjunto factível (domínio, restrições);  
Contém todas as *alternativas* viáveis.

# Otimização

## ▷ Definição

Matematicamente, um problema de otimização é definido como

$$\begin{array}{l} \text{minimizar } f(x) \text{ sujeito a } x \in \mathcal{X}. \\ \text{(maximizar)} \end{array}$$

- ▶  $x$ : variável de decisão; em geral:  $x = (x_1, x_2, \dots, x_n)$ .
- ▶  $\mathcal{X}$ : conjunto factível (domínio, restrições);  
Contém todas as *alternativas* viáveis.
- ▶  $f(x)$ : função objetivo; determina o *critério* de escolha.



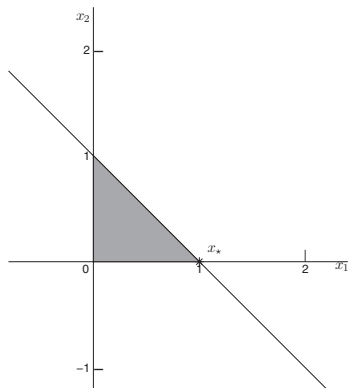
# Otimização

## ▷ Exemplos

$$\max f(x_1, x_2) = 2x_1 + x_2$$

$$\text{s.a } x_1 + x_2 \leq 1$$

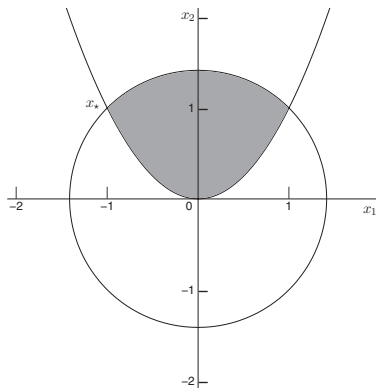
$$x_1, x_2 \geq 0.$$



# Otimização

## ▷ Exemplos

$$\begin{aligned} \min \quad & f(x_1, x_2) = x_1 \\ \text{s.a} \quad & x_1^2 \leq x_2 \\ & x_1^2 + x_2^2 \leq 2. \end{aligned}$$



# Otimização

## ▷ Nomenclatura

Matematicamente, um problema de otimização é definido como

$$\begin{array}{l} \text{minimizar } f(x) \text{ sujeito a } x \in \mathcal{X}. \\ \text{(maximizar)} \end{array}$$

- ▶ Uma solução  $\bar{x}$  é *factível* quando

# Otimização

## ▷ Nomenclatura

Matematicamente, um problema de otimização é definido como

$$\begin{array}{l} \text{minimizar } f(x) \text{ sujeito a } x \in \mathcal{X}. \\ \text{(maximizar)} \end{array}$$

- ▶ Uma solução  $\bar{x}$  é *factível* quando satisfaz todas as restrições do problema, i.e.,  $x \in \mathcal{X}$ .

# Otimização

## ▷ Nomenclatura

Matematicamente, um problema de otimização é definido como

$$\begin{array}{l} \text{minimizar } f(x) \text{ sujeito a } x \in \mathcal{X}. \\ \text{(maximizar)} \end{array}$$

- ▶ Uma solução  $\bar{x}$  é *factível* quando satisfaz todas as restrições do problema, i.e.,  $x \in \mathcal{X}$ . Caso contrário,  $\bar{x}$  é *infactível*.

# Otimização

## ▷ Nomenclatura

Matematicamente, um problema de otimização é definido como

$$\begin{array}{l} \text{minimizar } f(x) \text{ sujeito a } x \in \mathcal{X}. \\ \text{(maximizar)} \end{array}$$

- ▶ Uma solução  $\bar{x}$  é *factível* quando satisfaz todas as restrições do problema, i.e.,  $x \in \mathcal{X}$ . Caso contrário,  $\bar{x}$  é *infactível*.
- ▶ Uma solução  $\bar{x}$  é *ótima* quando

# Otimização

## ▷ Nomenclatura

Matematicamente, um problema de otimização é definido como

$$\begin{array}{l} \text{minimizar } f(x) \text{ sujeito a } x \in \mathcal{X}. \\ \text{(maximizar)} \end{array}$$

- ▶ Uma solução  $\bar{x}$  é *factível* quando satisfaz todas as restrições do problema, i.e.,  $x \in \mathcal{X}$ . Caso contrário,  $\bar{x}$  é *infactível*.
- ▶ Uma solução  $\bar{x}$  é *ótima* quando ela é factível e resulta no melhor valor da função objetivo, i.e.,  $f(\bar{x}) \leq f(x), \forall x \in \mathcal{X}$ .

# Otimização

## ▷ Nomenclatura

Matematicamente, um problema de otimização é definido como

$$\begin{array}{l} \text{minimizar } f(x) \text{ sujeito a } x \in \mathcal{X}. \\ \text{(maximizar)} \end{array}$$

- ▶ Uma solução  $\bar{x}$  é *factível* quando satisfaz todas as restrições do problema, i.e.,  $x \in \mathcal{X}$ . Caso contrário,  $\bar{x}$  é *infactível*.
- ▶ Uma solução  $\bar{x}$  é *ótima* quando ela é factível e resulta no melhor valor da função objetivo, i.e.,  $f(\bar{x}) \leq f(x), \forall x \in \mathcal{X}$ .
- ▶ Se  $\mathcal{X} = \emptyset$ , então



# Otimização

## ▷ Nomenclatura

Matematicamente, um problema de otimização é definido como

$$\begin{array}{l} \text{minimizar } f(x) \text{ sujeito a } x \in \mathcal{X}. \\ \text{(maximizar)} \end{array}$$

- ▶ Uma solução  $\bar{x}$  é *factível* quando satisfaz todas as restrições do problema, i.e.,  $x \in \mathcal{X}$ . Caso contrário,  $\bar{x}$  é *infactível*.
- ▶ Uma solução  $\bar{x}$  é *ótima* quando ela é factível e resulta no melhor valor da função objetivo, i.e.,  $f(\bar{x}) \leq f(x), \forall x \in \mathcal{X}$ .
- ▶ Se  $\mathcal{X} = \emptyset$ , então o problema é *infactível*.

# Otimização

## ▷ Nomenclatura

Matematicamente, um problema de otimização é definido como

$$\begin{array}{l} \text{minimizar } f(x) \text{ sujeito a } x \in \mathcal{X}. \\ \text{(maximizar)} \end{array}$$

- ▶ Uma solução  $\bar{x}$  é *factível* quando satisfaz todas as restrições do problema, i.e.,  $x \in \mathcal{X}$ . Caso contrário,  $\bar{x}$  é *infactível*.
- ▶ Uma solução  $\bar{x}$  é *ótima* quando ela é factível e resulta no melhor valor da função objetivo, i.e.,  $f(\bar{x}) \leq f(x), \forall x \in \mathcal{X}$ .
- ▶ Se  $\mathcal{X} = \emptyset$ , então o problema é *infactível*. Se para toda solução  $\bar{x} \in \mathcal{X}$

# Otimização

## ▷ Nomenclatura

Matematicamente, um problema de otimização é definido como

$$\begin{array}{l} \text{minimizar } f(x) \text{ sujeito a } x \in \mathcal{X}. \\ \text{(maximizar)} \end{array}$$

- ▶ Uma solução  $\bar{x}$  é *factível* quando satisfaz todas as restrições do problema, i.e.,  $x \in \mathcal{X}$ . Caso contrário,  $\bar{x}$  é *infectível*.
- ▶ Uma solução  $\bar{x}$  é *ótima* quando ela é factível e resulta no melhor valor da função objetivo, i.e.,  $f(\bar{x}) \leq f(x), \forall x \in \mathcal{X}$ .
- ▶ Se  $\mathcal{X} = \emptyset$ , então o problema é *infectível*. Se para toda solução  $\bar{x} \in \mathcal{X}$  existir uma outra solução  $\tilde{x} \in \mathcal{X}$

# Otimização

## ▷ Nomenclatura

Matematicamente, um problema de otimização é definido como

$$\begin{array}{l} \text{minimizar } f(x) \text{ sujeito a } x \in \mathcal{X}. \\ \text{(maximizar)} \end{array}$$

- ▶ Uma solução  $\bar{x}$  é *factível* quando satisfaz todas as restrições do problema, i.e.,  $x \in \mathcal{X}$ . Caso contrário,  $\bar{x}$  é *infectível*.
- ▶ Uma solução  $\bar{x}$  é *ótima* quando ela é factível e resulta no melhor valor da função objetivo, i.e.,  $f(\bar{x}) \leq f(x), \forall x \in \mathcal{X}$ .
- ▶ Se  $\mathcal{X} = \emptyset$ , então o problema é *infectível*. Se para toda solução  $\bar{x} \in \mathcal{X}$  existir uma outra solução  $\tilde{x} \in \mathcal{X}$  tal que  $f(\tilde{x}) < f(\bar{x})$ ,

# Otimização

## ▷ Nomenclatura

Matematicamente, um problema de otimização é definido como

$$\begin{array}{l} \text{minimizar } f(x) \text{ sujeito a } x \in \mathcal{X}. \\ \text{(maximizar)} \end{array}$$

- ▶ Uma solução  $\bar{x}$  é *factível* quando satisfaz todas as restrições do problema, i.e.,  $x \in \mathcal{X}$ . Caso contrário,  $\bar{x}$  é *infectível*.
- ▶ Uma solução  $\bar{x}$  é *ótima* quando ela é factível e resulta no melhor valor da função objetivo, i.e.,  $f(\bar{x}) \leq f(x), \forall x \in \mathcal{X}$ .
- ▶ Se  $\mathcal{X} = \emptyset$ , então o problema é *infectível*. Se para toda solução  $\bar{x} \in \mathcal{X}$  existir uma outra solução  $\tilde{x} \in \mathcal{X}$  tal que  $f(\tilde{x}) < f(\bar{x})$ , então o problema é *ilimitado*.

# Otimização

## ▷ Nomenclatura

Matematicamente, um problema de otimização é definido como

$$\begin{array}{l} \text{minimizar } f(x) \text{ sujeito a } x \in \mathcal{X}. \\ \text{(maximizar)} \end{array}$$

- ▶ Uma solução  $\bar{x}$  é *factível* quando satisfaz todas as restrições do problema, i.e.,  $x \in \mathcal{X}$ . Caso contrário,  $\bar{x}$  é *infectível*.
- ▶ Uma solução  $\bar{x}$  é *ótima* quando ela é factível e resulta no melhor valor da função objetivo, i.e.,  $f(\bar{x}) \leq f(x), \forall x \in \mathcal{X}$ .
- ▶ Se  $\mathcal{X} = \emptyset$ , então o problema é *infectível*. Se para toda solução  $\bar{x} \in \mathcal{X}$  existir uma outra solução  $\tilde{x} \in \mathcal{X}$  tal que  $f(\tilde{x}) < f(\bar{x})$ , então o problema é *ilimitado*. Em ambos os caso, *não* existe solução ótima.

# Otimização

## ▷ Classificação

Os modelos de otimização podem ser classificados como:

- ▶ lineares ou não-lineares;
- ▶ contínuos ou discretos;
- ▶ determinísticos ou estocásticos.

# Modelo de otimização linear x não-linear

- ▶ Um modelo linear é descrito por meio de funções, equações e inequações lineares;



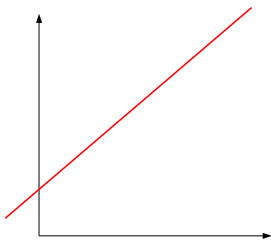
# Modelo de otimização linear x não-linear

- ▶ Um modelo linear é descrito por meio de funções, equações e inequações lineares;
- ▶ Funções lineares são aquelas que variam de forma constante. Podem envolver apenas a soma de variáveis e a multiplicação de variáveis por constantes.

# Modelo de otimização linear x não-linear

## ▷ Funções lineares

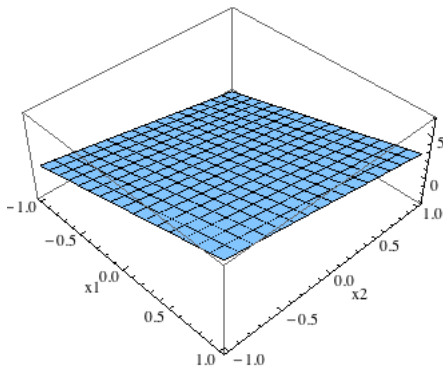
$$f(x) = bx + c$$



# Modelo de otimização linear x não-linear

## ▷ Funções lineares

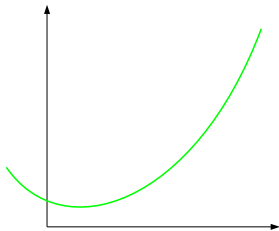
$$f(x_1, x_2) = 3x_1 - 2x_2 + 2$$



# Modelo de otimização linear x não-linear

## ▷ Funções não-lineares

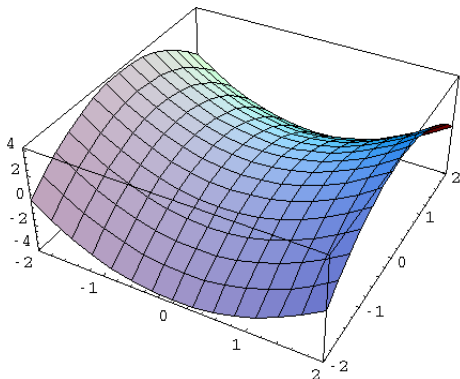
$$f(x) = ax^2 + bx + c$$



# Modelo de otimização linear x não-linear

## ▷ Funções não-lineares

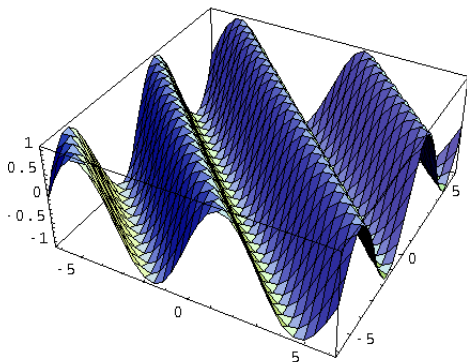
$$f(x_1, x_2) = x_1^2 - x_2^2$$



# Modelo de otimização linear x não-linear

## ▷ Funções não-lineares

$$f(x_1, x_2) = \text{sen}(x_1 + x_2)$$



## Modelo de otimização linear x não-linear

- ▶ Um modelo é contínuo quando seu conjunto factível contém todos os pontos interiores à sua fronteira;

## Modelo de otimização linear x não-linear

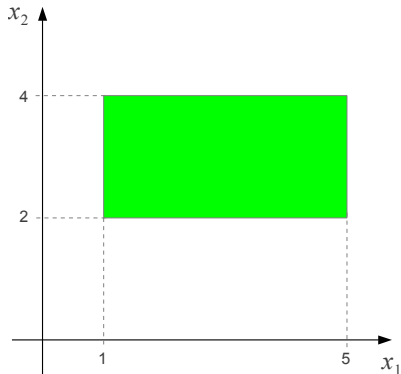
- ▶ Um modelo é contínuo quando seu conjunto factível contém todos os pontos interiores à sua fronteira;
- ▶ Em um conjunto factível discreto, os pontos factíveis são isolados no conjunto.



# Modelo de otimização contínuo x discreto

## ▷ Domínio contínuo

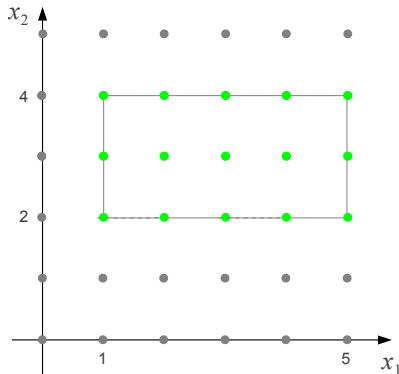
$$\mathcal{X} = \{(x_1, x_2) \in \mathbb{R}^2 \mid 1 \leq x_1 \leq 5, 2 \leq x_2 \leq 4\}$$



# Modelo de otimização contínuo x discreto

## ▷ Domínio discreto

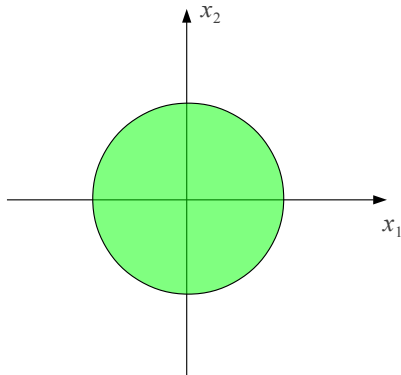
$$\mathcal{X} = \{(x_1, x_2) \in \mathbb{Z}^2 \mid 1 \leq x_1 \leq 5, 2 \leq x_2 \leq 4\}$$



# Modelo de otimização contínuo x discreto

## ▷ Domínio contínuo

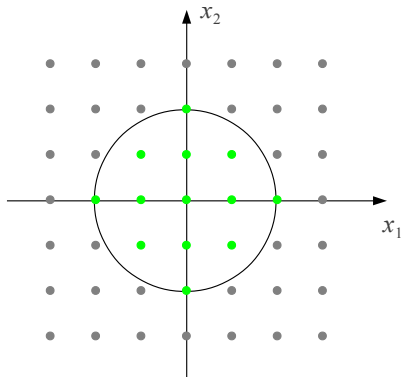
$$\mathcal{X} = \{(x_1, x_2) \in \mathbb{R}^2 \mid x_1^2 + x_2^2 \leq r\}$$



# Modelo de otimização contínuo x discreto

▷ Domínio discreto

$$\mathcal{X} = \{(x_1, x_2) \in \mathbb{Z}^2 \mid x_1^2 + x_2^2 \leq r\}$$



## Modelo de otimização determinístico x estocástico

- ▶ Um modelo é determinístico quando os dados de entrada são conhecidos com certeza. Por exemplo: a demanda de um dado produto para os três próximos meses é 170, 185 e 147 unidades;

## Modelo de otimização determinístico x estocástico

- ▶ Um modelo é determinístico quando os dados de entrada são conhecidos com certeza. Por exemplo: a demanda de um dado produto para os três próximos meses é 170, 185 e 147 unidades;
- ▶ Em um modelo estocástico, há incerteza quanto aos dados de entrada. A demanda para os próximos meses não é conhecida, mas são conhecidos possíveis cenários e suas probabilidades de ocorrência.

# Otimização

## ▷ Nomenclatura

- ▶ Programação Linear: modelo linear contínuo determinístico;
- ▶ Programação Inteira: modelo linear discreto determinístico;
- ▶ Programação Estocástica: modelo linear contínuo estocástico;
- ▶ Programação Não-linear: modelo não-linear contínuo determinístico;
- ▶ ...

# Otimização

## ▷ Métodos de solução

- ▶ Para cada tipo de modelo, existem métodos (algoritmos) que podem ser usados para obter uma solução.



# Otimização

## ▷ Métodos de solução

- ▶ Para cada tipo de modelo, existem métodos (algoritmos) que podem ser usados para obter uma solução.
- ▶ Por exemplo, para obter a solução ótima de um modelo de otimização linear contínuo determinístico (programação linear) podemos usar o *método simplex* ou o *método de pontos interiores*.

# Otimização

## ▷ Métodos de solução

- ▶ Para cada tipo de modelo, existem métodos (algoritmos) que podem ser usados para obter uma solução.
- ▶ Por exemplo, para obter a solução ótima de um modelo de otimização linear contínuo determinístico (programação linear) podemos usar o *método simplex* ou o *método de pontos interiores*.
- ▶ Para modelos de otimização discreta, temos o método branch-and-bound;

# Otimização

## ▷ Métodos de solução

- ▶ Para cada tipo de modelo, existem métodos (algoritmos) que podem ser usados para obter uma solução.
- ▶ Por exemplo, para obter a solução ótima de um modelo de otimização linear contínuo determinístico (programação linear) podemos usar o *método simplex* ou o *método de pontos interiores*.
- ▶ Para modelos de otimização discreta, temos o método branch-and-bound;
- ▶ Alguns modelos possuem uma estrutura especial que pode ser explorada por técnicas de reformulação (Dantzig-Wolfe, Lagrangiana, Benders).

# Programação Linear

## ▷ Modelo

$$\text{minimizar } f(x_1, \dots, x_n) = c_1x_1 + \dots + c_nx_n$$

$$\text{sujeito a } a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \leq b_1$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \geq b_2$$

$$\vdots \quad \quad \quad \vdots \quad \quad \quad \vdots$$

$$a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n = b_m$$

$$x_1, \dots, x_n \geq 0$$

# Programação Linear

## ▷ Modelo

$$\text{minimizar } f(x_1, \dots, x_n) = c_1x_1 + \dots + c_nx_n$$

$$\text{sujeito a } a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \leq b_1$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \geq b_2$$

$$\vdots \qquad \qquad \qquad \vdots \qquad \qquad \qquad \vdots$$

$$a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n = b_m$$

$$x_1, \dots, x_n \geq 0$$

Parâmetros: dados de entrada (números)

# Programação Linear

## ▷ Modelo

$$\text{minimizar } f(x_1, x_2, x_3) = 3x_1 + 7x_2 + 2x_3$$

$$\text{sujeito a } 1x_1 + 4x_2 + 8x_3 \leq 15$$

$$5x_1 + 5x_2 + 3x_3 \geq 9$$

$$12x_1 + 9x_2 + 1x_3 = 4$$

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

# Programação Linear

## ▷ Nomenclatura

- ▶ Modelo de otimização linear contínuo determinístico.

# Programação Linear

## ▷ Nomenclatura

- ▶ Modelo de otimização linear contínuo determinístico.
- ▶ Minimizar ou maximizar uma função linear

$$f(x_1, \dots, x_n) = c_1x_1 + \dots + c_nx_n$$



# Programação Linear

## ▷ Nomenclatura

- ▶ Modelo de otimização linear contínuo determinístico.
- ▶ Minimizar ou maximizar uma função linear

$$f(x_1, \dots, x_n) = c_1x_1 + \dots + c_nx_n$$

- ▶ sujeita a um domínio contínuo, representado por um sistema de equações e/ou inequações lineares, p.ex.:

$$a_{11}x_1 + \dots + a_{1n}x_n = b_1$$

$$a_{21}x_1 + \dots + a_{2n}x_n \geq b_2$$

$$a_{31}x_1 + \dots + a_{3n}x_n \leq b_3$$

# Programação Linear

## ▷ Nomenclatura

- ▶ Modelo de otimização linear contínuo determinístico.
- ▶ Minimizar ou maximizar uma função linear

$$f(x_1, \dots, x_n) = c_1x_1 + \dots + c_nx_n$$

- ▶ sujeita a um domínio contínuo, representado por um sistema de equações e/ou inequações lineares, p.ex.:

$$a_{11}x_1 + \dots + a_{1n}x_n = b_1$$

$$a_{21}x_1 + \dots + a_{2n}x_n \geq b_2$$

$$a_{31}x_1 + \dots + a_{3n}x_n \leq b_3$$

- ▶ Obs.: não é permitido usar '<' nem '>'.

# Programação Linear

## ▷ Nomenclatura

- ▶  $x_1, \dots, x_n \in \mathbb{R}$ : variáveis de decisão

# Programação Linear

## ▷ Nomenclatura

- ▶  $x_1, \dots, x_n \in \mathbb{R}$ : variáveis de decisão
- ▶  $c_1, \dots, c_n \in \mathbb{R}$ : coeficientes de custo

# Programação Linear

## ▷ Nomenclatura

- ▶  $x_1, \dots, x_n \in \mathbb{R}$ : variáveis de decisão
- ▶  $c_1, \dots, c_n \in \mathbb{R}$ : coeficientes de custo
- ▶  $a_{11}x_1 + \dots + a_{1n}x_n = b_1$ : restrição do problema

# Programação Linear

## ▷ Nomenclatura

- ▶  $x_1, \dots, x_n \in \mathbb{R}$ : variáveis de decisão
- ▶  $c_1, \dots, c_n \in \mathbb{R}$ : coeficientes de custo
- ▶  $a_{11}x_1 + \dots + a_{1n}x_n = b_1$ : restrição do problema
- ▶  $b_1, \dots, b_m \in \mathbb{R}$ : coeficientes do vetor de recursos

# Programação Linear

## ▷ Nomenclatura

- ▶  $x_1, \dots, x_n \in \mathbb{R}$ : variáveis de decisão
- ▶  $c_1, \dots, c_n \in \mathbb{R}$ : coeficientes de custo
- ▶  $a_{11}x_1 + \dots + a_{1n}x_n = b_1$ : restrição do problema
- ▶  $b_1, \dots, b_m \in \mathbb{R}$ : coeficientes do vetor de recursos
- ▶  $a_{11}, \dots, a_{mn} \in \mathbb{R}$ : coeficientes da matriz de recursos

# Programação Linear

## ▷ Propriedades

Um modelo de programação linear deve satisfazer:



# Programação Linear

## ▷ Propriedades

Um modelo de programação linear deve satisfazer:

- ▶ **Proporcionalidade:** A contribuição de cada variável é diretamente proporcional ao seu valor; (constante multiplicando variável)

# Programação Linear

## ▷ Propriedades

Um modelo de programação linear deve satisfazer:

- ▶ **Proporcionalidade:** A contribuição de cada variável é diretamente proporcional ao seu valor; (constante multiplicando variável)
- ▶ **Aditividade:** A contribuição total das variáveis é igual à soma direta das contribuições individuais de cada uma; (apenas soma ou subtração entre variáveis)

# Programação Linear

## ▷ Propriedades

Um modelo de programação linear deve satisfazer:

- ▶ **Proporcionalidade:** A contribuição de cada variável é diretamente proporcional ao seu valor; (constante multiplicando variável)
- ▶ **Aditividade:** A contribuição total das variáveis é igual à soma direta das contribuições individuais de cada uma; (apenas soma ou subtração entre variáveis)
- ▶ **Fracionamento:** As variáveis podem assumir valores fracionários;

# Programação Linear

## ▷ Propriedades

Um modelo de programação linear deve satisfazer:

- ▶ **Proporcionalidade:** A contribuição de cada variável é diretamente proporcional ao seu valor; (constante multiplicando variável)
- ▶ **Aditividade:** A contribuição total das variáveis é igual à soma direta das contribuições individuais de cada uma; (apenas soma ou subtração entre variáveis)
- ▶ **Fracionamento:** As variáveis podem assumir valores fracionários;
- ▶ **Certeza:** Todos os dados são determinísticos.

## Exemplo

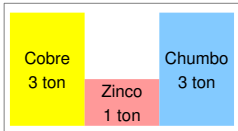
Uma metalúrgica produz dois tipos de ligas metálicas. Cada liga é composta de proporções diferentes de cobre, zinco e chumbo, os quais estão disponíveis em quantidades limitadas em estoque. Deseja-se determinar quanto produzir de cada liga metálica, de modo a maximizar a receita bruta, satisfazendo-se as seguintes composições das ligas e a disponibilidade de matéria-prima em estoque:

Matéria-prima	Liga 1	Liga 2	Estoque
Cobre	50%	30%	3 ton
Zinco	10%	20%	1 ton
Chumbo	40%	50%	3 ton
Preço venda	3 mil	2 mil	(R\$ por ton)

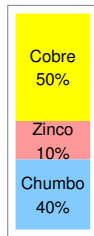
# Exemplo



**Estoque**

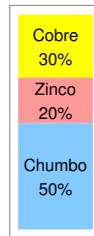


**Liga 1**



R\$ 3 mil/ton

**Liga 2**



R\$ 2 mil/ton

(Composição)

(Preço)

# Exemplo

## ▷ Modelagem

- ▶ Variáveis de decisão?

# Exemplo

## ▷ Modelagem

- ▶ Variáveis de decisão?

$x_i \rightarrow$  qtd a ser produzida da liga  $i$ , para  $i = 1, 2$ .



# Exemplo

## ▷ Modelagem

- ▶ Variáveis de decisão?

$x_i \rightarrow$  qtd a ser produzida da liga  $i$ , para  $i = 1, 2$ .

- ▶ Restrições?

# Exemplo

## ▷ Modelagem

- ▶ Variáveis de decisão?

$x_i \rightarrow$  qtd a ser produzida da liga  $i$ , para  $i = 1, 2$ .

- ▶ Restrições?

Estoque de cobre:

# Exemplo

## ▷ Modelagem

- ▶ Variáveis de decisão?

$x_i \rightarrow$  qtd a ser produzida da liga  $i$ , para  $i = 1, 2$ .

- ▶ Restrições?

Estoque de cobre:  $0,5x_1 + 0,3x_2 \leq 3$

# Exemplo

## ▷ Modelagem

- ▶ Variáveis de decisão?

$x_i \rightarrow$  qtd a ser produzida da liga  $i$ , para  $i = 1, 2$ .

- ▶ Restrições?

Estoque de cobre:  $0,5x_1 + 0,3x_2 \leq 3$

Estoque de zinco:

# Exemplo

## ▷ Modelagem

- ▶ Variáveis de decisão?

$x_i \rightarrow$  qtd a ser produzida da liga  $i$ , para  $i = 1, 2$ .

- ▶ Restrições?

Estoque de cobre:  $0,5x_1 + 0,3x_2 \leq 3$

Estoque de zinco:  $0,1x_1 + 0,2x_2 \leq 1$

# Exemplo

## ▷ Modelagem

- ▶ Variáveis de decisão?

$x_i \rightarrow$  qtd a ser produzida da liga  $i$ , para  $i = 1, 2$ .

- ▶ Restrições?

Estoque de cobre:  $0,5x_1 + 0,3x_2 \leq 3$

Estoque de zinco:  $0,1x_1 + 0,2x_2 \leq 1$

Estoque de chumbo:

# Exemplo

## ▷ Modelagem

- ▶ Variáveis de decisão?

$x_i \rightarrow$  qtd a ser produzida da liga  $i$ , para  $i = 1, 2$ .

- ▶ Restrições?

Estoque de cobre:  $0,5x_1 + 0,3x_2 \leq 3$

Estoque de zinco:  $0,1x_1 + 0,2x_2 \leq 1$

Estoque de chumbo:  $0,4x_1 + 0,5x_2 \leq 3$

# Exemplo

## ▷ Modelagem

- ▶ Variáveis de decisão?

$x_i \rightarrow$  qtd a ser produzida da liga  $i$ , para  $i = 1, 2$ .

- ▶ Restrições?

Estoque de cobre:  $0,5x_1 + 0,3x_2 \leq 3$

Estoque de zinco:  $0,1x_1 + 0,2x_2 \leq 1$

Estoque de chumbo:  $0,4x_1 + 0,5x_2 \leq 3$

Quantidades não-negativas:



# Exemplo

## ▷ Modelagem

- ▶ Variáveis de decisão?

$x_i \rightarrow$  qtd a ser produzida da liga  $i$ , para  $i = 1, 2$ .

- ▶ Restrições?

Estoque de cobre:  $0,5x_1 + 0,3x_2 \leq 3$

Estoque de zinco:  $0,1x_1 + 0,2x_2 \leq 1$

Estoque de chumbo:  $0,4x_1 + 0,5x_2 \leq 3$

Quantidades não-negativas:  $x_1 \geq 0, x_2 \geq 0$

# Exemplo

## ▷ Modelagem

- ▶ Variáveis de decisão?

$x_i \rightarrow$  qtd a ser produzida da liga  $i$ , para  $i = 1, 2$ .

- ▶ Restrições?

Estoque de cobre:  $0,5x_1 + 0,3x_2 \leq 3$

Estoque de zinco:  $0,1x_1 + 0,2x_2 \leq 1$

Estoque de chumbo:  $0,4x_1 + 0,5x_2 \leq 3$

Quantidades não-negativas:  $x_1 \geq 0, x_2 \geq 0$

- ▶ Função objetivo?

# Exemplo

## ▷ Modelagem

- ▶ Variáveis de decisão?

$x_i \rightarrow$  qtd a ser produzida da liga  $i$ , para  $i = 1, 2$ .

- ▶ Restrições?

Estoque de cobre:  $0,5x_1 + 0,3x_2 \leq 3$

Estoque de zinco:  $0,1x_1 + 0,2x_2 \leq 1$

Estoque de chumbo:  $0,4x_1 + 0,5x_2 \leq 3$

Quantidades não-negativas:  $x_1 \geq 0, x_2 \geq 0$

- ▶ Função objetivo?

Maximizar  $3x_1 + 2x_2$

# Exemplo

## ▷ Modelagem

$$\max \quad f(x_1, x_2) = 3x_1 + 2x_2$$

$$\text{s.a} \quad 0,5x_1 + 0,3x_2 \leq 3$$

$$0,1x_1 + 0,2x_2 \leq 1$$

$$0,4x_1 + 0,5x_2 \leq 3$$

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

# Problemas de mix de produção

## ▷ Exemplo

Uma fábrica produz utensílios domésticos feitos de metal. São feitos 3 produtos por meio de 3 operações (estamparia, perfuração e montagem), sendo que cada operação possui um limite máximo de horas disponíveis. A fabricação dos produtos 2 e 3 consome um material que está disponível em quantidades limitadas em estoque. A disponibilidade do material e de horas em cada operação, bem como o quanto cada unidade produzida consome desses recursos são descritos na seguinte tabela:

Setor	Taxa de produção (horas por unidade)			Recurso disponível
	Produto 1	Produto 2	Produto 3	
Estamparia	0,03	0,15	0,10	400 h
Perfuração	0,06	0,12	0,10	400 h
Montagem	0,05	0,10	0,12	500 h
Material	—	2,0	1,2	2000 $m^2$

# Problemas de mix de produção

## ▷ Exemplo

A fábrica fez um levantamento de qual o custo unitário e o preço de venda adequado para cada produto, bem como uma estimativa para o mínimo e máximo de vendas, sendo:

	Produto 1	Produto 2	Produto 3
Preço unitário	10	25	20
Custo unitário	6	15	14
Mínimo vendas	1000	—	100
Máximo vendas	6000	500	1000

Deseja-se determinar quanto fabricar de cada produto, de modo a maximizar o lucro.

# Problemas de mix de produção

## ▷ Modelo com os dados

- ▶ Variáveis de decisão?

# Problemas de mix de produção

## ▷ Modelo com os dados

### ▶ Variáveis de decisão?

$x_i \rightarrow$  qtd a ser fabricada do produto  $i$ , para  $i = 1, 2, 3$ .



# Problemas de mix de produção

## ▷ Modelo com os dados

### ▶ Variáveis de decisão?

$x_i \rightarrow$  qtd a ser fabricada do produto  $i$ , para  $i = 1, 2, 3$ .

### ▶ Restrições?

# Problemas de mix de produção

## ▷ Modelo com os dados

### ▶ Variáveis de decisão?

$x_i \rightarrow$  qtd a ser fabricada do produto  $i$ , para  $i = 1, 2, 3$ .

### ▶ Restrições?

Estamparia:

# Problemas de mix de produção

## ▷ Modelo com os dados

### ▶ Variáveis de decisão?

$x_i \rightarrow$  qtd a ser fabricada do produto  $i$ , para  $i = 1, 2, 3$ .

### ▶ Restrições?

Estamparia:  $0,03x_1 + 0,15x_2 + 0,10x_3 \leq 400$

# Problemas de mix de produção

## ▷ Modelo com os dados

### ▶ Variáveis de decisão?

$x_i \rightarrow$  qtd a ser fabricada do produto  $i$ , para  $i = 1, 2, 3$ .

### ▶ Restrições?

Estamparia:  $0,03x_1 + 0,15x_2 + 0,10x_3 \leq 400$

Perfuração:

# Problemas de mix de produção

## ▷ Modelo com os dados

### ▶ Variáveis de decisão?

$x_i \rightarrow$  qtd a ser fabricada do produto  $i$ , para  $i = 1, 2, 3$ .

### ▶ Restrições?

Estamparia:  $0,03x_1 + 0,15x_2 + 0,10x_3 \leq 400$

Perfuração:  $0,06x_1 + 0,12x_2 + 0,10x_3 \leq 400$

# Problemas de mix de produção

## ▷ Modelo com os dados

### ▶ Variáveis de decisão?

$x_i \rightarrow$  qtd a ser fabricada do produto  $i$ , para  $i = 1, 2, 3$ .

### ▶ Restrições?

Estamparia:  $0,03x_1 + 0,15x_2 + 0,10x_3 \leq 400$

Perfuração:  $0,06x_1 + 0,12x_2 + 0,10x_3 \leq 400$

Montagem:

# Problemas de mix de produção

## ▷ Modelo com os dados

### ▶ Variáveis de decisão?

$x_i \rightarrow$  qtd a ser fabricada do produto  $i$ , para  $i = 1, 2, 3$ .

### ▶ Restrições?

Estamparia:  $0,03x_1 + 0,15x_2 + 0,10x_3 \leq 400$

Perfuração:  $0,06x_1 + 0,12x_2 + 0,10x_3 \leq 400$

Montagem:  $0,05x_1 + 0,10x_2 + 0,12x_3 \leq 500$

# Problemas de mix de produção

## ▷ Modelo com os dados

### ▶ Variáveis de decisão?

$x_i \rightarrow$  qtd a ser fabricada do produto  $i$ , para  $i = 1, 2, 3$ .

### ▶ Restrições?

Estamparia:  $0,03x_1 + 0,15x_2 + 0,10x_3 \leq 400$

Perfuração:  $0,06x_1 + 0,12x_2 + 0,10x_3 \leq 400$

Montagem:  $0,05x_1 + 0,10x_2 + 0,12x_3 \leq 500$

Material:



# Problemas de mix de produção

## ▷ Modelo com os dados

### ▶ Variáveis de decisão?

$x_i \rightarrow$  qtd a ser fabricada do produto  $i$ , para  $i = 1, 2, 3$ .

### ▶ Restrições?

Estamparia:  $0,03x_1 + 0,15x_2 + 0,10x_3 \leq 400$

Perfuração:  $0,06x_1 + 0,12x_2 + 0,10x_3 \leq 400$

Montagem:  $0,05x_1 + 0,10x_2 + 0,12x_3 \leq 500$

Material:  $2,0x_2 + 1,2x_3 \leq 2000$

# Problemas de mix de produção

## ▷ Modelo com os dados

### ▶ Variáveis de decisão?

$x_i \rightarrow$  qtd a ser fabricada do produto  $i$ , para  $i = 1, 2, 3$ .

### ▶ Restrições?

Estamparia:  $0,03x_1 + 0,15x_2 + 0,10x_3 \leq 400$

Perfuração:  $0,06x_1 + 0,12x_2 + 0,10x_3 \leq 400$

Montagem:  $0,05x_1 + 0,10x_2 + 0,12x_3 \leq 500$

Material:  $2,0x_2 + 1,2x_3 \leq 2000$

Quantidades mínimas e máximas:

# Problemas de mix de produção

## ▷ Modelo com os dados

### ▶ Variáveis de decisão?

$x_i \rightarrow$  qtd a ser fabricada do produto  $i$ , para  $i = 1, 2, 3$ .

### ▶ Restrições?

Estamparia:  $0,03x_1 + 0,15x_2 + 0,10x_3 \leq 400$

Perfuração:  $0,06x_1 + 0,12x_2 + 0,10x_3 \leq 400$

Montagem:  $0,05x_1 + 0,10x_2 + 0,12x_3 \leq 500$

Material:  $2,0x_2 + 1,2x_3 \leq 2000$

Quantidades mínimas e máximas:

$1000 \leq x_1 \leq 6000, 0 \leq x_2 \leq 500, 100 \leq x_3 \leq 1000$

# Problemas de mix de produção

## ▷ Modelo com os dados

### ▶ Variáveis de decisão?

$x_i \rightarrow$  qtd a ser fabricada do produto  $i$ , para  $i = 1, 2, 3$ .

### ▶ Restrições?

Estamparia:  $0,03x_1 + 0,15x_2 + 0,10x_3 \leq 400$

Perfuração:  $0,06x_1 + 0,12x_2 + 0,10x_3 \leq 400$

Montagem:  $0,05x_1 + 0,10x_2 + 0,12x_3 \leq 500$

Material:  $2,0x_2 + 1,2x_3 \leq 2000$

Quantidades mínimas e máximas:

$1000 \leq x_1 \leq 6000, 0 \leq x_2 \leq 500, 100 \leq x_3 \leq 1000$

### ▶ Função objetivo?

# Problemas de mix de produção

## ▷ Modelo com os dados

### ▶ Variáveis de decisão?

$x_i \rightarrow$  qtd a ser fabricada do produto  $i$ , para  $i = 1, 2, 3$ .

### ▶ Restrições?

Estamparia:  $0,03x_1 + 0,15x_2 + 0,10x_3 \leq 400$

Perfuração:  $0,06x_1 + 0,12x_2 + 0,10x_3 \leq 400$

Montagem:  $0,05x_1 + 0,10x_2 + 0,12x_3 \leq 500$

Material:  $2,0x_2 + 1,2x_3 \leq 2000$

Quantidades mínimas e máximas:

$1000 \leq x_1 \leq 6000, 0 \leq x_2 \leq 500, 100 \leq x_3 \leq 1000$

### ▶ Função objetivo?

Maximizar  $4x_1 + 10x_2 + 6x_3$

## Problemas de mix de produção

### ▷ Modelo com os dados

$$\max f(x_1, x_2, x_3) = 4x_1 + 10x_2 + 6x_3$$

$$\text{s.a } 0,03x_1 + 0,15x_2 + 0,10x_3 \leq 400$$

$$0,06x_1 + 0,12x_2 + 0,10x_3 \leq 400$$

$$0,05x_1 + 0,10x_2 + 0,12x_3 \leq 500$$

$$2,0x_2 + 1,2x_3 \leq 2000$$

$$1000 \leq x_1 \leq 6000$$

$$0 \leq x_2 \leq 500$$

$$100 \leq x_3 \leq 1000$$

# Problemas de mix de produção

## ▷ Descrição geral

- ▶ Existem vários produtos que a empresa pode produzir num período.  
Determinar **quanto** produzir de cada produto;

# Problemas de mix de produção

## ▷ Descrição geral

- ▶ Existem vários produtos que a empresa pode produzir num período.  
Determinar **quanto** produzir de cada produto;
- ▶ Múltiplos produtos, um período;



# Problemas de mix de produção

## ▷ Descrição geral

- ▶ Existem vários produtos que a empresa pode produzir num período.  
Determinar **quanto** produzir de cada produto;
- ▶ Múltiplos produtos, um período;
- ▶ Restrições:

# Problemas de mix de produção

## ▷ Descrição geral

- ▶ Existem vários produtos que a empresa pode produzir num período.  
Determinar **quanto** produzir de cada produto;
- ▶ Múltiplos produtos, um período;
- ▶ Restrições:
  - ▶ limitações de recursos;

# Problemas de mix de produção

## ▷ Descrição geral

- ▶ Existem vários produtos que a empresa pode produzir num período.  
Determinar **quanto** produzir de cada produto;
- ▶ Múltiplos produtos, um período;
- ▶ Restrições:
  - ▶ limitações de recursos;
  - ▶ previsão de demanda mínima e máxima.

# Problemas de mix de produção

## ▷ Descrição geral

- ▶ Existem vários produtos que a empresa pode produzir num período.  
Determinar **quanto** produzir de cada produto;
- ▶ Múltiplos produtos, um período;
- ▶ Restrições:
  - ▶ limitações de recursos;
  - ▶ previsão de demanda mínima e máxima.
- ▶ Objetivo:

# Problemas de mix de produção

## ▷ Descrição geral

- ▶ Existem vários produtos que a empresa pode produzir num período.  
Determinar **quanto** produzir de cada produto;
- ▶ Múltiplos produtos, um período;
- ▶ Restrições:
  - ▶ limitações de recursos;
  - ▶ previsão de demanda mínima e máxima.
- ▶ Objetivo:
  - ▶ maximizar o lucro.

# Problemas de mix de produção

## ▷ Modelagem

### ▶ Parâmetros:

# Problemas de mix de produção

## ▷ Modelagem

### ▶ Parâmetros:

- ▶ Conjunto de produtos:  $\mathcal{P} = \{1, \dots, n\}$ ;

# Problemas de mix de produção

## ▷ Modelagem

### ▶ Parâmetros:

- ▶ Conjunto de produtos:  $\mathcal{P} = \{1, \dots, n\}$ ;
- ▶ Conjunto de recursos:  $\mathcal{R} = \{1, \dots, K\}$ ;



# Problemas de mix de produção

## ▷ Modelagem

### ▶ Parâmetros:

- ▶ Conjunto de produtos:  $\mathcal{P} = \{1, \dots, n\}$ ;
- ▶ Conjunto de recursos:  $\mathcal{R} = \{1, \dots, K\}$ ;
- ▶  $r_i$ : retorno do produto  $i$ ;

# Problemas de mix de produção

## ▷ Modelagem

### ▶ Parâmetros:

- ▶ Conjunto de produtos:  $\mathcal{P} = \{1, \dots, n\}$ ;
- ▶ Conjunto de recursos:  $\mathcal{R} = \{1, \dots, K\}$ ;
- ▶  $r_i$ : retorno do produto  $i$ ;
- ▶  $c_i$ : custo unitário do produto  $i$ ;

# Problemas de mix de produção

## ▷ Modelagem

### ▶ Parâmetros:

- ▶ Conjunto de produtos:  $\mathcal{P} = \{1, \dots, n\}$ ;
- ▶ Conjunto de recursos:  $\mathcal{R} = \{1, \dots, K\}$ ;
- ▶  $r_i$ : retorno do produto  $i$ ;
- ▶  $c_i$ : custo unitário do produto  $i$ ;
- ▶  $b_k$ : qtd do recurso  $k$ ;

# Problemas de mix de produção

## ▷ Modelagem

### ▶ Parâmetros:

- ▶ Conjunto de produtos:  $\mathcal{P} = \{1, \dots, n\}$ ;
- ▶ Conjunto de recursos:  $\mathcal{R} = \{1, \dots, K\}$ ;
- ▶  $r_i$ : retorno do produto  $i$ ;
- ▶  $c_i$ : custo unitário do produto  $i$ ;
- ▶  $b_k$ : qtd do recurso  $k$ ;
- ▶  $a_{ki}$ : qtd do recurso  $k$  para produzir 1 unidade do produto  $i$ ;

# Problemas de mix de produção

## ▷ Modelagem

### ▶ Parâmetros:

- ▶ Conjunto de produtos:  $\mathcal{P} = \{1, \dots, n\}$ ;
- ▶ Conjunto de recursos:  $\mathcal{R} = \{1, \dots, K\}$ ;
- ▶  $r_i$ : retorno do produto  $i$ ;
- ▶  $c_i$ : custo unitário do produto  $i$ ;
- ▶  $b_k$ : qtd do recurso  $k$ ;
- ▶  $a_{ki}$ : qtd do recurso  $k$  para produzir 1 unidade do produto  $i$ ;
- ▶  $L_i, U_i$ : demandas mínima e máxima do produto  $i$ .

# Problemas de mix de produção

## ▷ Modelagem

- ▶ Variáveis de decisão:

# Problemas de mix de produção

## ▷ Modelagem

- ▶ Variáveis de decisão:

$x_i$ :

# Problemas de mix de produção

## ▷ Modelagem

- ▶ Variáveis de decisão:  
 $x_i$ : qtd a ser produzida do produto  $i$ .
- ▶ Restrições:



# Problemas de mix de produção

## ▷ Modelagem

- ▶ Variáveis de decisão:

$x_i$ : qtd a ser produzida do produto  $i$ .

- ▶ Restrições:

Limitação do recurso  $k$ :

# Problemas de mix de produção

## ▷ Modelagem

- ▶ Variáveis de decisão:

$x_i$ : qtd a ser produzida do produto  $i$ .

- ▶ Restrições:

Limitação do recurso  $k$ : 
$$\sum_{i=1}^n a_{ki}x_i \leq b_k,$$

# Problemas de mix de produção

## ▷ Modelagem

- ▶ Variáveis de decisão:

$x_i$ : qtd a ser produzida do produto  $i$ .

- ▶ Restrições:

Limitação do recurso  $k$ : 
$$\sum_{i=1}^n a_{ki}x_i \leq b_k, \quad k = 1, \dots, K;$$

# Problemas de mix de produção

## ▷ Modelagem

- ▶ Variáveis de decisão:

$x_i$ : qtd a ser produzida do produto  $i$ .

- ▶ Restrições:

Limitação do recurso  $k$ :  $\sum_{i=1}^n a_{ki}x_i \leq b_k, k = 1, \dots, K;$

Demanda do produto  $i$ :

# Problemas de mix de produção

## ▷ Modelagem

- ▶ Variáveis de decisão:

$x_i$ : qtd a ser produzida do produto  $i$ .

- ▶ Restrições:

Limitação do recurso  $k$ :  $\sum_{i=1}^n a_{ki}x_i \leq b_k, k = 1, \dots, K;$

Demanda do produto  $i$ :  $L_i \leq x_i \leq U_i,$

# Problemas de mix de produção

## ▷ Modelagem

- ▶ Variáveis de decisão:

$x_i$ : qtd a ser produzida do produto  $i$ .

- ▶ Restrições:

Limitação do recurso  $k$ :  $\sum_{i=1}^n a_{ki}x_i \leq b_k, k = 1, \dots, K;$

Demanda do produto  $i$ :  $L_i \leq x_i \leq U_i, i = 1, \dots, n.$

# Problemas de mix de produção

## ▷ Modelagem

- ▶ Variáveis de decisão:

$x_i$ : qtd a ser produzida do produto  $i$ .

- ▶ Restrições:

Limitação do recurso  $k$ :  $\sum_{i=1}^n a_{ki}x_i \leq b_k, k = 1, \dots, K;$

Demanda do produto  $i$ :  $L_i \leq x_i \leq U_i, i = 1, \dots, n.$

- ▶ Função objetivo:

# Problemas de mix de produção

## ▷ Modelagem

- ▶ Variáveis de decisão:

$x_i$ : qtd a ser produzida do produto  $i$ .

- ▶ Restrições:

Limitação do recurso  $k$ :  $\sum_{i=1}^n a_{ki}x_i \leq b_k, k = 1, \dots, K;$

Demanda do produto  $i$ :  $L_i \leq x_i \leq U_i, i = 1, \dots, n.$

- ▶ Função objetivo:

Maximizar o lucro:  $\max \sum_{i=1}^n (r_i - c_i)x_i.$



# Problemas de mix de produção

## ▷ Modelo algébrico

$$\begin{aligned} \max \quad & \sum_{i=1}^n (r_i - c_i)x_i \\ \text{s.a} \quad & \sum_{i=1}^n a_{ki}x_i \leq b_k, \quad k = 1, \dots, K, \\ & L_i \leq x_i \leq U_i, \quad i = 1, \dots, n. \end{aligned}$$

## Problemas de mix de produção

▷ Modelo algébrico (usando conjuntos)

$$\begin{aligned} \max \quad & \sum_{i \in \mathcal{P}} (r_i - c_i) x_i \\ \text{s.a} \quad & \sum_{i \in \mathcal{P}} a_{ki} x_i \leq b_k, \quad k \in \mathcal{R}, \\ & L_i \leq x_i \leq U_i, \quad i \in \mathcal{P}. \end{aligned}$$

# Problemas de mix de produção

## ▷ Modelo com os dados

$$\max f(x_1, x_2, x_3) = 4x_1 + 10x_2 + 6x_3$$

$$\text{s.a } 0,03x_1 + 0,15x_2 + 0,10x_3 \leq 400$$

$$0,06x_1 + 0,12x_2 + 0,10x_3 \leq 400$$

$$0,05x_1 + 0,10x_2 + 0,12x_3 \leq 500$$

$$2,0x_2 + 1,2x_3 \leq 2000$$

$$1000 \leq x_1 \leq 6000$$

$$0 \leq x_2 \leq 500$$

$$100 \leq x_3 \leq 1000$$

# Problemas de mistura (*blending*)



**INFORMAÇÃO NUTRICIONAL/INFORMACIÓN NUTRICIONAL**  
Porção de/Porción de 20g (1 unidade/unidad)

Quantidade por porção/ Cantidad por porción		%VD**
Valor energético	64 kcal=269 kJ	3%
Carboidratos/Carbohidratos	11 g	4%
Proteínas	1,4 g	2%
Gorduras totais/Grasas totales	1,5 g	3%
Gorduras saturadas/Grasas saturadas	0,6 g	3%
Gorduras trans/Grasas trans	não contém/ no contiene	**
Gorduras monoinsaturadas/ Grasas monoinsaturadas	0,4 g	**
Gorduras poli-insaturadas/ Grasas poliinsaturadas	0,2 g	**
Colesterol	0 mg	0%
Fibra alimentar/Fibra alimentaria	4,4 g	18%
Sódio/Sodio	27 mg	1%

\*\*% Valores diários de referência com base em uma dieta de 2.000 kcal ou 8.400 kJ. Seus valores diários podem ser maiores ou menores dependendo de suas necessidades energéticas. \*\* VD não estabelecido.  
\*\*% Valores diarios de referencia con base a una dieta de 2.000 kcal u 8.400kJ. Sus valores diarios pueden ser mayores o menores dependiendo de sus necesidades energéticas. \*\* VD no establecido.

**FAZ BEM SABER**

**Serviço Nestlé ao Consumidor**  
0800-7702457 SMS: 25770  
CAIXA POSTAL 21144  
CEP 04602-970  
SÃO PAULO - SP  
FALECOM@NESTLE.COM.BR  
WWW.NESTLE.COM.BR

NUTRITIONAL COMPASS®  
® Marca Registrada de Société des Produits Nestlé S.A.

**Uma Dica Nestlé**  
*Cereal integral é muito mais do que fibra! Além da fibra, o cereal integral conserva todas as partes do grão e seus nutrientes. Inclua no seu dia a dia alimentos com cereais integrais para uma alimentação mais equilibrada.*

DISPLAY  
PAPEL

12152255  
43338509

## Problemas de mistura (*blending*)

### ▷ Exemplo

Uma fábrica de alimentos deseja produzir um novo sabor de barra de cereais. Os requisitos nutricionais exigem que as barras tenham certas quantidades mínimas e máximas de certos nutrientes principais, sendo: no mínimo 22% de fibra e 7% de proteína; e no máximo 55% de carboidrato e 8% de gorduras. Para produzir as barras, a indústria usará como ingredientes, farinha de cereais, mel, soja e banana. As proporções de nutrientes em cada ingrediente, bem como os custos por quilograma de cada ingrediente são apresentados na tabela a seguir:

Nutrientes	Ingredientes				Barra
	Cereais	Mel	Soja	Banana	
Fibra	0,26	0,01	0,25	0,10	0,22
Proteína	0,05	0,05	0,26	0,02	0,07
Carboidrato	0,60	0,75	0,45	0,24	0,55
Gorduras	0,07	–	0,01	0,01	0,08
Custos (R\$/kg)	5,20	6,80	7,10	2,50	

A fábrica deseja determinar em que quantidades os ingredientes devem ser misturados de modo a produzir 1 kg da barra de cereais que satisfaça às restrições nutricionais e tenha custo mínimo.

# Problemas de mistura (*blending*)

## ▷ Descrição geral

- ▶ Misturar diversos ingredientes para produzir um produto com certas especificações de componentes;

# Problemas de mistura (*blending*)

## ▷ Descrição geral

- ▶ Misturar diversos ingredientes para produzir um produto com certas especificações de componentes;
- ▶ Existem várias combinações possíveis que permitem produzir o produto;

# Problemas de mistura (*blending*)

## ▷ Descrição geral

- ▶ Misturar diversos ingredientes para produzir um produto com certas especificações de componentes;
- ▶ Existem várias combinações possíveis que permitem produzir o produto;
- ▶ Exemplos: fabricação de ração, fertilizantes, ligas metálicas, ...



# Problemas de mistura (*blending*)

## ▷ Descrição geral

- ▶ Misturar diversos ingredientes para produzir um produto com certas especificações de componentes;
- ▶ Existem várias combinações possíveis que permitem produzir o produto;
- ▶ Exemplos: fabricação de ração, fertilizantes, ligas metálicas, ...
- ▶ Restrições:

# Problemas de mistura (*blending*)

## ▷ Descrição geral

- ▶ Misturar diversos ingredientes para produzir um produto com certas especificações de componentes;
- ▶ Existem várias combinações possíveis que permitem produzir o produto;
- ▶ Exemplos: fabricação de ração, fertilizantes, ligas metálicas, ...
- ▶ Restrições:
  - ▶ Garantir as especificações do produto.

# Problemas de mistura (*blending*)

## ▷ Descrição geral

- ▶ Misturar diversos ingredientes para produzir um produto com certas especificações de componentes;
- ▶ Existem várias combinações possíveis que permitem produzir o produto;
- ▶ Exemplos: fabricação de ração, fertilizantes, ligas metálicas, ...
- ▶ Restrições:
  - ▶ Garantir as especificações do produto.
- ▶ Objetivos:

# Problemas de mistura (*blending*)

## ▷ Descrição geral

- ▶ Misturar diversos ingredientes para produzir um produto com certas especificações de componentes;
- ▶ Existem várias combinações possíveis que permitem produzir o produto;
- ▶ Exemplos: fabricação de ração, fertilizantes, ligas metálicas, ...
- ▶ Restrições:
  - ▶ Garantir as especificações do produto.
- ▶ Objetivos:
  - ▶ Minimizar o custo de produzir uma dada quantidade da mistura.

# Problemas de mistura

## ▷ Modelagem

▶ Parâmetros:

# Problemas de mistura

## ▷ Modelagem

### ▶ Parâmetros:

- ▶ Ingredientes:  $\mathcal{I} = \{1, \dots, n\}$ ;

# Problemas de mistura

## ▷ Modelagem

### ▶ Parâmetros:

- ▶ Ingredientes:  $\mathcal{I} = \{1, \dots, n\}$ ;
- ▶ Componentes:  $\mathcal{J} = \{1, \dots, m\}$ ;

# Problemas de mistura

## ▷ Modelagem

### ▶ Parâmetros:

- ▶ Ingredientes:  $\mathcal{I} = \{1, \dots, n\}$ ;
- ▶ Componentes:  $\mathcal{J} = \{1, \dots, m\}$ ;
- ▶  $c_i$ : custo do ingrediente  $i \in \mathcal{I}$ ;



# Problemas de mistura

## ▷ Modelagem

### ▶ Parâmetros:

- ▶ Ingredientes:  $\mathcal{I} = \{1, \dots, n\}$ ;
- ▶ Componentes:  $\mathcal{J} = \{1, \dots, m\}$ ;
- ▶  $c_i$ : custo do ingrediente  $i \in \mathcal{I}$ ;
- ▶  $a_{ji}$ : qtd do componente  $j \in \mathcal{J}$  em 1 unid do ingrediente  $i \in \mathcal{I}$ ;

# Problemas de mistura

## ▷ Modelagem

### ▶ Parâmetros:

- ▶ Ingredientes:  $\mathcal{I} = \{1, \dots, n\}$ ;
- ▶ Componentes:  $\mathcal{J} = \{1, \dots, m\}$ ;
- ▶  $c_i$ : custo do ingrediente  $i \in \mathcal{I}$ ;
- ▶  $a_{ji}$ : qtd do componente  $j \in \mathcal{J}$  em 1 unid do ingrediente  $i \in \mathcal{I}$ ;
- ▶  $b_j$ : qtd do componente  $j \in \mathcal{J}$  exigida na mistura;

# Problemas de mistura

## ▷ Modelagem

- ▶ Variáveis de decisão:

# Problemas de mistura

## ▷ Modelagem

### ▶ Variáveis de decisão:

$x_i$ : qtd do ingrediente  $i \in \mathcal{I}$  a ser usado para produzir 1 unid de mistura.

# Problemas de mistura

## ▷ Modelagem

### ▶ Variáveis de decisão:

$x_i$ : qtd do ingrediente  $i \in \mathcal{I}$  a ser usado para produzir 1 unid de mistura.

### ▶ Restrições:

# Problemas de mistura

## ▷ Modelagem

- ▶ Variáveis de decisão:

$x_i$ : qtd do ingrediente  $i \in \mathcal{I}$  a ser usado para produzir 1 unid de mistura.

- ▶ Restrições:

Qtd do componente:

# Problemas de mistura

## ▷ Modelagem

- ▶ Variáveis de decisão:

$x_i$ : qtd do ingrediente  $i \in \mathcal{I}$  a ser usado para produzir 1 unid de mistura.

- ▶ Restrições:

Qtd do componente:  $\sum_{i \in \mathcal{I}} a_{ji} x_i \geq b_j$ ,

# Problemas de mistura

## ▷ Modelagem

- ▶ Variáveis de decisão:

$x_i$ : qtd do ingrediente  $i \in \mathcal{I}$  a ser usado para produzir 1 unid de mistura.

- ▶ Restrições:

Qtd do componente:  $\sum_{i \in \mathcal{I}} a_{ji} x_i \geq b_j, j \in \mathcal{J};$



# Problemas de mistura

## ▷ Modelagem

- ▶ Variáveis de decisão:

$x_i$ : qtd do ingrediente  $i \in \mathcal{I}$  a ser usado para produzir 1 unid de mistura.

- ▶ Restrições:

Qtd do componente:  $\sum_{i \in \mathcal{I}} a_{ji} x_i \geq b_j, j \in \mathcal{J};$

Produzir 1 unid da mistura:

# Problemas de mistura

## ▷ Modelagem

- ▶ Variáveis de decisão:

$x_i$ : qtd do ingrediente  $i \in \mathcal{I}$  a ser usado para produzir 1 unid de mistura.

- ▶ Restrições:

Qtd do componente:  $\sum_{i \in \mathcal{I}} a_{ji} x_i \geq b_j, j \in \mathcal{J};$

Produzir 1 unid da mistura:  $\sum_{i \in \mathcal{I}} x_i = 1;$

# Problemas de mistura

## ▷ Modelagem

- ▶ Variáveis de decisão:

$x_i$ : qtd do ingrediente  $i \in \mathcal{I}$  a ser usado para produzir 1 unid de mistura.

- ▶ Restrições:

Qtd do componente:  $\sum_{i \in \mathcal{I}} a_{ji} x_i \geq b_j, j \in \mathcal{J};$

Produzir 1 unid da mistura:  $\sum_{i \in \mathcal{I}} x_i = 1;$

Não-negatividade:

# Problemas de mistura

## ▷ Modelagem

- ▶ Variáveis de decisão:

$x_i$ : qtd do ingrediente  $i \in \mathcal{I}$  a ser usado para produzir 1 unid de mistura.

- ▶ Restrições:

Qtd do componente:  $\sum_{i \in \mathcal{I}} a_{ji} x_i \geq b_j, j \in \mathcal{J};$

Produzir 1 unid da mistura:  $\sum_{i \in \mathcal{I}} x_i = 1;$

Não-negatividade:  $x_i \geq 0, i \in \mathcal{I}.$

# Problemas de mistura

## ▷ Modelagem

- ▶ Variáveis de decisão:

$x_i$ : qtd do ingrediente  $i \in \mathcal{I}$  a ser usado para produzir 1 unid de mistura.

- ▶ Restrições:

Qtd do componente:  $\sum_{i \in \mathcal{I}} a_{ji} x_i \geq b_j, j \in \mathcal{J};$

Produzir 1 unid da mistura:  $\sum_{i \in \mathcal{I}} x_i = 1;$

Não-negatividade:  $x_i \geq 0, i \in \mathcal{I}.$

- ▶ Função objetivo:

# Problemas de mistura

## ▷ Modelagem

- ▶ Variáveis de decisão:

$x_i$ : qtd do ingrediente  $i \in \mathcal{I}$  a ser usado para produzir 1 unid de mistura.

- ▶ Restrições:

Qtd do componente:  $\sum_{i \in \mathcal{I}} a_{ji} x_i \geq b_j, j \in \mathcal{J};$

Produzir 1 unid da mistura:  $\sum_{i \in \mathcal{I}} x_i = 1;$

Não-negatividade:  $x_i \geq 0, i \in \mathcal{I}.$

- ▶ Função objetivo:

Minimizar o custo:

# Problemas de mistura

## ▷ Modelagem

- ▶ Variáveis de decisão:

$x_i$ : qtd do ingrediente  $i \in \mathcal{I}$  a ser usado para produzir 1 unid de mistura.

- ▶ Restrições:

Qtd do componente:  $\sum_{i \in \mathcal{I}} a_{ji} x_i \geq b_j, j \in \mathcal{J};$

Produzir 1 unid da mistura:  $\sum_{i \in \mathcal{I}} x_i = 1;$

Não-negatividade:  $x_i \geq 0, i \in \mathcal{I}.$

- ▶ Função objetivo:

Minimizar o custo:  $\min \sum_{i \in \mathcal{I}} c_i x_i.$

# Problemas de mistura

## ▷ Modelo algébrico

$$\begin{aligned} \min \quad & \sum_{i \in \mathcal{I}} c_i x_i \\ \text{s.a} \quad & \sum_{i \in \mathcal{I}} a_{ji} x_i \left\{ \begin{array}{l} \leq \\ = \\ \geq \end{array} \right\} b_j, \quad j \in \mathcal{J}, \\ & \sum_{i \in \mathcal{I}} x_i = 1. \\ & x_i \geq 0, \quad i \in \mathcal{I}. \end{aligned}$$



## Problemas de mistura

### ▷ Modelo com dados

$$\begin{aligned} \min \quad & f(x) = 5.2x_1 + 6.8x_2 + 7.1x_3 + 2.5x_4 \\ \text{s.a} \quad & 0,26x_1 + 0,01x_2 + 0,25x_3 + 0,10x_4 \geq 0,22 \\ & 0,05x_1 + 0,05x_2 + 0,26x_3 + 0,02x_4 \geq 0,07 \\ & 0,60x_1 + 0,75x_2 + 0,45x_3 + 0,24x_4 \leq 0,55 \\ & 0,07x_1 + 0,01x_3 + 0,01x_4 \leq 0,08 \\ & x_1 + x_2 + x_3 + x_4 = 1 \\ & x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0. \end{aligned}$$

## Problemas de mistura

### ▷ Modelo com dados

$$\begin{aligned} \min \quad & f(x) = 5.2x_1 + 6.8x_2 + 7.1x_3 + 2.5x_4 \\ \text{s.a} \quad & 0,26x_1 + 0,01x_2 + 0,25x_3 + 0,10x_4 \geq 0,22 \\ & 0,05x_1 + 0,05x_2 + 0,26x_3 + 0,02x_4 \geq 0,07 \\ & 0,60x_1 + 0,75x_2 + 0,45x_3 + 0,24x_4 \leq 0,55 \\ & 0,07x_1 + 0,01x_3 + 0,01x_4 \leq 0,08 \\ & x_1 + x_2 + x_3 + x_4 = 1 \\ & x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0. \end{aligned}$$

*E se quiséssemos produzir  $Q$  kg da mistura?*

## Problemas de transporte

Uma fabricante de bebidas possui dois centros de produção: um em Ribeirão Preto-SP e outro em Cariacica-ES. A empresa deseja planejar qual a melhor forma de atender a demanda para a próxima semana, de mercados consumidores em três capitais: São Paulo, Belo Horizonte e Rio de Janeiro. O custo unitário de se transportar 1 fardo de bebida de cada centro de produção a cada mercado consumidor é dado na tabela a seguir, juntamente com as demandas de cada mercado e as quantidades disponíveis em cada centro de produção:

Custos (R\$/unid)	SP	BH	RJ	Disponível
Ribeirão Preto	3,70	4,30	6,10	1100 unid
Cariacica	9,80	6,90	2,10	1800 unid
Demanda (unid)	960	510	895	

A empresa deseja determinar como atender a demanda de cada mercado consumidor, minimizando os gastos totais com transporte.

# Problemas de transporte

Variáveis de decisão:

# Problemas de transporte

Variáveis de decisão:

- ▶  $x_{ij}$ : qtd a ser transportada da origem  $i$  até o destino  $j$ , com  $i = 1, 2$  e  $j = 1, 2, 3$ .

# Problemas de transporte

Variáveis de decisão:

- ▶  $x_{ij}$ : qtd a ser transportada da origem  $i$  até o destino  $j$ , com  $i = 1, 2$  e  $j = 1, 2, 3$ .

Restrições:

# Problemas de transporte

Variáveis de decisão:

- ▶  $x_{ij}$ : qtd a ser transportada da origem  $i$  até o destino  $j$ , com  $i = 1, 2$  e  $j = 1, 2, 3$ .

Restrições:

- ▶ Origem 1:

# Problemas de transporte

Variáveis de decisão:

- ▶  $x_{ij}$ : qtd a ser transportada da origem  $i$  até o destino  $j$ , com  $i = 1, 2$  e  $j = 1, 2, 3$ .

Restrições:

- ▶ Origem 1:  $x_{11} + x_{12} + x_{13} \leq 1100$ ;



# Problemas de transporte

Variáveis de decisão:

- ▶  $x_{ij}$ : qtd a ser transportada da origem  $i$  até o destino  $j$ , com  $i = 1, 2$  e  $j = 1, 2, 3$ .

Restrições:

- ▶ Origem 1:  $x_{11} + x_{12} + x_{13} \leq 1100$ ;
- ▶ Origem 2:

# Problemas de transporte

Variáveis de decisão:

- ▶  $x_{ij}$ : qtd a ser transportada da origem  $i$  até o destino  $j$ , com  $i = 1, 2$  e  $j = 1, 2, 3$ .

Restrições:

- ▶ Origem 1:  $x_{11} + x_{12} + x_{13} \leq 1100$ ;
- ▶ Origem 2:  $x_{21} + x_{22} + x_{23} \leq 1800$ ;

# Problemas de transporte

Variáveis de decisão:

- ▶  $x_{ij}$ : qtd a ser transportada da origem  $i$  até o destino  $j$ , com  $i = 1, 2$  e  $j = 1, 2, 3$ .

Restrições:

- ▶ Origem 1:  $x_{11} + x_{12} + x_{13} \leq 1100$ ;
- ▶ Origem 2:  $x_{21} + x_{22} + x_{23} \leq 1800$ ;
- ▶ Destino 1:

# Problemas de transporte

Variáveis de decisão:

- ▶  $x_{ij}$ : qtd a ser transportada da origem  $i$  até o destino  $j$ , com  $i = 1, 2$  e  $j = 1, 2, 3$ .

Restrições:

- ▶ Origem 1:  $x_{11} + x_{12} + x_{13} \leq 1100$ ;
- ▶ Origem 2:  $x_{21} + x_{22} + x_{23} \leq 1800$ ;
- ▶ Destino 1:  $x_{11} + x_{21} = 960$ ;

# Problemas de transporte

Variáveis de decisão:

- ▶  $x_{ij}$ : qtd a ser transportada da origem  $i$  até o destino  $j$ , com  $i = 1, 2$  e  $j = 1, 2, 3$ .

Restrições:

- ▶ Origem 1:  $x_{11} + x_{12} + x_{13} \leq 1100$ ;
- ▶ Origem 2:  $x_{21} + x_{22} + x_{23} \leq 1800$ ;
- ▶ Destino 1:  $x_{11} + x_{21} = 960$ ;
- ▶ Destino 2:

# Problemas de transporte

Variáveis de decisão:

- ▶  $x_{ij}$ : qtd a ser transportada da origem  $i$  até o destino  $j$ , com  $i = 1, 2$  e  $j = 1, 2, 3$ .

Restrições:

- ▶ Origem 1:  $x_{11} + x_{12} + x_{13} \leq 1100$ ;
- ▶ Origem 2:  $x_{21} + x_{22} + x_{23} \leq 1800$ ;
- ▶ Destino 1:  $x_{11} + x_{21} = 960$ ;
- ▶ Destino 2:  $x_{12} + x_{22} = 510$ ;

# Problemas de transporte

Variáveis de decisão:

- ▶  $x_{ij}$ : qtd a ser transportada da origem  $i$  até o destino  $j$ , com  $i = 1, 2$  e  $j = 1, 2, 3$ .

Restrições:

- ▶ Origem 1:  $x_{11} + x_{12} + x_{13} \leq 1100$ ;
- ▶ Origem 2:  $x_{21} + x_{22} + x_{23} \leq 1800$ ;
- ▶ Destino 1:  $x_{11} + x_{21} = 960$ ;
- ▶ Destino 2:  $x_{12} + x_{22} = 510$ ;
- ▶ Destino 3:

# Problemas de transporte

Variáveis de decisão:

- ▶  $x_{ij}$ : qtd a ser transportada da origem  $i$  até o destino  $j$ , com  $i = 1, 2$  e  $j = 1, 2, 3$ .

Restrições:

- ▶ Origem 1:  $x_{11} + x_{12} + x_{13} \leq 1100$ ;
- ▶ Origem 2:  $x_{21} + x_{22} + x_{23} \leq 1800$ ;
- ▶ Destino 1:  $x_{11} + x_{21} = 960$ ;
- ▶ Destino 2:  $x_{12} + x_{22} = 510$ ;
- ▶ Destino 3:  $x_{13} + x_{23} = 895$ ;



# Problemas de transporte

Variáveis de decisão:

- ▶  $x_{ij}$ : qtd a ser transportada da origem  $i$  até o destino  $j$ , com  $i = 1, 2$  e  $j = 1, 2, 3$ .

Restrições:

- ▶ Origem 1:  $x_{11} + x_{12} + x_{13} \leq 1100$ ;
- ▶ Origem 2:  $x_{21} + x_{22} + x_{23} \leq 1800$ ;
- ▶ Destino 1:  $x_{11} + x_{21} = 960$ ;
- ▶ Destino 2:  $x_{12} + x_{22} = 510$ ;
- ▶ Destino 3:  $x_{13} + x_{23} = 895$ ;
- ▶ Não-negatividade:

# Problemas de transporte

Variáveis de decisão:

- ▶  $x_{ij}$ : qtd a ser transportada da origem  $i$  até o destino  $j$ , com  $i = 1, 2$  e  $j = 1, 2, 3$ .

Restrições:

- ▶ Origem 1:  $x_{11} + x_{12} + x_{13} \leq 1100$ ;
- ▶ Origem 2:  $x_{21} + x_{22} + x_{23} \leq 1800$ ;
- ▶ Destino 1:  $x_{11} + x_{21} = 960$ ;
- ▶ Destino 2:  $x_{12} + x_{22} = 510$ ;
- ▶ Destino 3:  $x_{13} + x_{23} = 895$ ;
- ▶ Não-negatividade:  $x_{ij} \geq 0$ .

Função objetivo?

# Problemas de transporte

Variáveis de decisão:

- ▶  $x_{ij}$ : qtd a ser transportada da origem  $i$  até o destino  $j$ , com  $i = 1, 2$  e  $j = 1, 2, 3$ .

Restrições:

- ▶ Origem 1:  $x_{11} + x_{12} + x_{13} \leq 1100$ ;
- ▶ Origem 2:  $x_{21} + x_{22} + x_{23} \leq 1800$ ;
- ▶ Destino 1:  $x_{11} + x_{21} = 960$ ;
- ▶ Destino 2:  $x_{12} + x_{22} = 510$ ;
- ▶ Destino 3:  $x_{13} + x_{23} = 895$ ;
- ▶ Não-negatividade:  $x_{ij} \geq 0$ .

Função objetivo?

- ▶ Minimizar custos:

# Problemas de transporte

Variáveis de decisão:

- ▶  $x_{ij}$ : qtd a ser transportada da origem  $i$  até o destino  $j$ , com  $i = 1, 2$  e  $j = 1, 2, 3$ .

Restrições:

- ▶ Origem 1:  $x_{11} + x_{12} + x_{13} \leq 1100$ ;
- ▶ Origem 2:  $x_{21} + x_{22} + x_{23} \leq 1800$ ;
- ▶ Destino 1:  $x_{11} + x_{21} = 960$ ;
- ▶ Destino 2:  $x_{12} + x_{22} = 510$ ;
- ▶ Destino 3:  $x_{13} + x_{23} = 895$ ;
- ▶ Não-negatividade:  $x_{ij} \geq 0$ .

Função objetivo?

- ▶ Minimizar custos:

$$\min 3,7x_{11} + 4,3x_{12} + 6,1x_{13} + 9,8x_{21} + 6,9x_{22} + 2,1x_{23}$$

# Problemas de transporte

## ▷ Modelo

$$\begin{aligned} \min \quad & f(x) = 3,7x_{11} + 4,3x_{12} + 6,1x_{13} \\ & + 9,8x_{21} + 6,9x_{22} + 2,1x_{23} \end{aligned}$$

$$\begin{aligned} \text{s.a} \quad & x_{11} + x_{12} + x_{13} \leq 1100 \\ & x_{21} + x_{22} + x_{23} \leq 1800 \\ & x_{11} + x_{21} = 960 \\ & x_{12} + x_{22} = 510 \\ & x_{13} + x_{23} = 895 \\ & x_{11}, x_{12}, \dots, x_{23} \geq 0. \end{aligned}$$

# Problemas de transporte

## ▷ Modelo algébrico

$$\min \quad f(x) = \sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij}$$

$$\text{s.a} \quad \sum_{j=1}^m x_{ij} \leq a_i, \quad i = 1, \dots, n,$$

$$\sum_{i=1}^n x_{ij} = d_j, \quad j = 1, \dots, m,$$

$$x_{ij} \geq 0, \quad i = 1, \dots, n, \quad j = 1, \dots, m.$$

## Solução de problemas com o uso de *software*

## Solução de problemas com o uso de *software*

- ▶ Existem diversos *softwares* de otimização disponíveis atualmente. Dentre os mais utilizados/conhecidos podemos citar:



## Solução de problemas com o uso de *software*

- ▶ Existem diversos *softwares* de otimização disponíveis atualmente. Dentre os mais utilizados/conhecidos podemos citar:
  - ▶ Excel (Microsoft) – conta com o Solver (nativo) e outros suplementos mais eficientes (OpenSolver e SolverStudio);

## Solução de problemas com o uso de *software*

- ▶ Existem diversos *softwares* de otimização disponíveis atualmente. Dentre os mais utilizados/conhecidos podemos citar:
  - ▶ Excel (Microsoft) – conta com o Solver (nativo) e outros suplementos mais eficientes (OpenSolver e SolverStudio);
  - ▶ `lp_solve` – simples, didático, livre e gratuito;

## Solução de problemas com o uso de *software*

- ▶ Existem diversos *softwares* de otimização disponíveis atualmente. Dentre os mais utilizados/conhecidos podemos citar:
  - ▶ Excel (Microsoft) – conta com o Solver (nativo) e outros suplementos mais eficientes (OpenSolver e SolverStudio);
  - ▶ `lp_solve` – simples, didático, livre e gratuito;
  - ▶ CPLEX (IBM) – comercial (\$\$\$) e um dos mais eficientes;

## Solução de problemas com o uso de *software*

- ▶ Existem diversos *softwares* de otimização disponíveis atualmente. Dentre os mais utilizados/conhecidos podemos citar:
  - ▶ Excel (Microsoft) – conta com o Solver (nativo) e outros suplementos mais eficientes (OpenSolver e SolverStudio);
  - ▶ `lp_solve` – simples, didático, livre e gratuito;
  - ▶ CPLEX (IBM) – comercial (\$\$\$) e um dos mais eficientes;
  - ▶ GLPK – livre, gratuito e pode ser uma boa alternativa aos softwares comerciais.

## Solução de problemas com o uso de *software*

- ▶ Existem diversos *softwares* de otimização disponíveis atualmente. Dentre os mais utilizados/conhecidos podemos citar:
  - ▶ Excel (Microsoft) – conta com o Solver (nativo) e outros suplementos mais eficientes (OpenSolver e SolverStudio);
  - ▶ `lp_solve` – simples, didático, livre e gratuito;
  - ▶ CPLEX (IBM) – comercial (\$\$\$) e um dos mais eficientes;
  - ▶ GLPK – livre, gratuito e pode ser uma boa alternativa aos softwares comerciais.
- ▶ Em geral, os *softwares* usam o **algoritmo simplex** para a resolução dos problemas de Programação Linear e variações do método **branch-and-bound** para a resolução de problemas de Programação Inteira.

## Solução de problemas com o uso de *software*

- ▶ O Solver/Excel tem como principais vantagens:
  - ▶ Estar disponível em quase todo computador;

## Solução de problemas com o uso de *software*

- ▶ O Solver/Excel tem como principais vantagens:
  - ▶ Estar disponível em quase todo computador;
  - ▶ Entrada de dados por uma planilha do Excel.

## Solução de problemas com o uso de *software*

- ▶ O Solver/Excel tem como principais vantagens:
  - ▶ Estar disponível em quase todo computador;
  - ▶ Entrada de dados por uma planilha do Excel.
- ▶ Por outro lado, além limitar o número máximo de variáveis e restrições, pode ser ineficiente até mesmo em problemas relativamente pequenos.



## Solução de problemas com o uso de *software*

- ▶ O Solver/Excel tem como principais vantagens:
  - ▶ Estar disponível em quase todo computador;
  - ▶ Entrada de dados por uma planilha do Excel.
- ▶ Por outro lado, além limitar o número máximo de variáveis e restrições, pode ser ineficiente até mesmo em problemas relativamente pequenos.
- ▶ O `lp_solve` é uma boa alternativa que permite a entrada de dados de forma bastante intuitiva;

## Solução de problemas com o uso de *software*

- ▶ O Solver/Excel tem como principais vantagens:
  - ▶ Estar disponível em quase todo computador;
  - ▶ Entrada de dados por uma planilha do Excel.
- ▶ Por outro lado, além limitar o número máximo de variáveis e restrições, pode ser ineficiente até mesmo em problemas relativamente pequenos.
- ▶ O `lp_solve` é uma boa alternativa que permite a entrada de dados de forma bastante intuitiva;
  - ▶ Tem caráter mais didático, embora possa ser usado na resolução de problemas de pequeno porte.

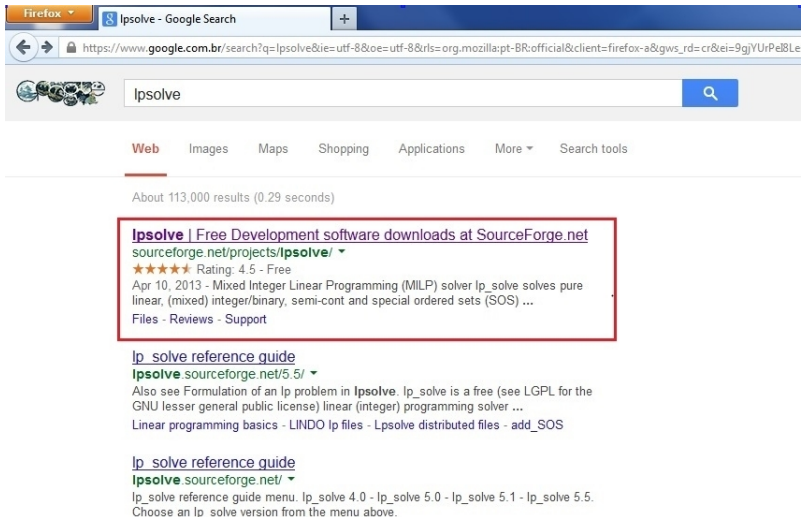
## Solução de problemas com o uso de *software*

- ▶ Na prática, CPLEX e GLPK são mais apropriados na resolução de problemas de grande porte, embora o uso desses softwares seja bem mais complicado e exija um conhecimento mais avançado de PO e linguagem de programação;

## Solução de problemas com o uso de *software*

- ▶ Na prática, CPLEX e GLPK são mais apropriados na resolução de problemas de grande porte, embora o uso desses softwares seja bem mais complicado e exija um conhecimento mais avançado de PO e linguagem de programação;
- ▶ Algumas linguagens de modelagem com interface gráfica podem facilitar o uso desses softwares, por exemplo OPL, GAMS e AMPL.

## Para baixar e instalar o lp\_solve:



Firefox Ipsolve - Google Search

https://www.google.com.br/search?q=Ipsolve&ie=utf-8&oe=utf-8&rls=org.mozilla:pt-BR:official&client=firefox-a&gws\_rd=cr&ei=9gYUrpel8Le

Ipsolve

Web Images Maps Shopping Applications More Search tools

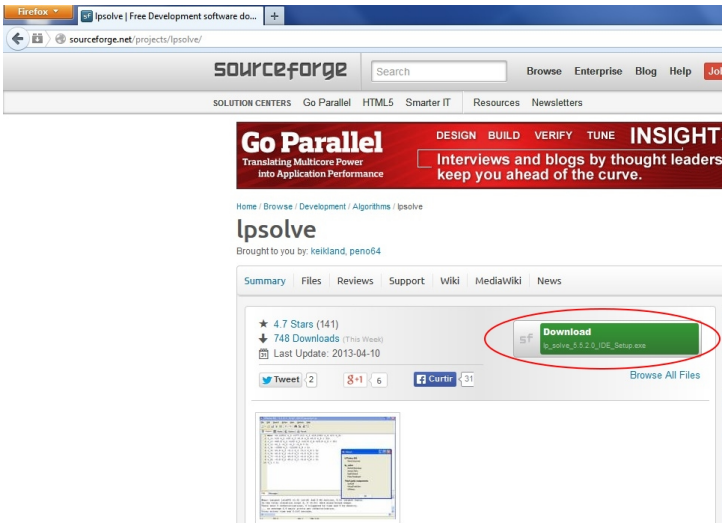
About 113,000 results (0.29 seconds)

**Ipsolve** | Free Development software downloads at SourceForge.net  
sourceforge.net/projects/ipsolve/ ▾  
★★★★★ Rating: 4.5 - Free  
Apr 10, 2013 - Mixed Integer Linear Programming (MILP) solver Ip\_solve solves pure linear, (mixed) integer/binary, semi-cont and special ordered sets (SOS) ...  
Files - Reviews - Support

[Ip\\_solve reference guide](#)  
[ipsolve.sourceforge.net/5.5/](#) ▾  
Also see Formulation of an Ip problem in Ipsolve. Ip\_solve is a free (see LGPL for the GNU lesser general public license) linear (integer) programming solver ...  
Linear programming basics - LINDO Ip files - Lpsolve distributed files - add\_SOS

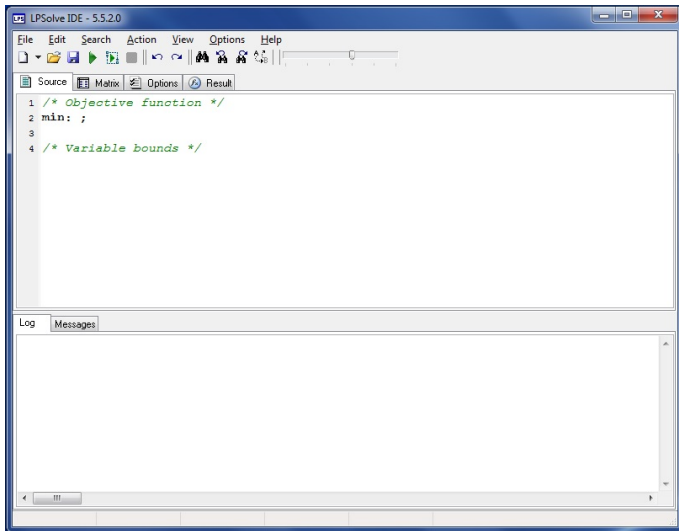
[Ip\\_solve reference guide](#)  
[ipsolve.sourceforge.net/](#) ▾  
Ip\_solve reference guide menu. Ip\_solve 4.0 - Ip\_solve 5.0 - Ip\_solve 5.1 - Ip\_solve 5.5.  
Choose an Ip\_solve version from the menu above.

Para baixar e instalar o lp\_solve:

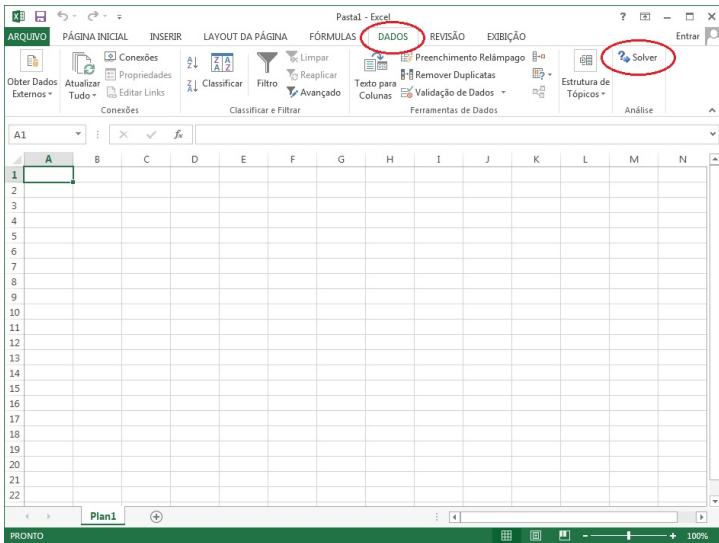


The screenshot shows a web browser window displaying the SourceForge project page for lp\_solve. The browser's address bar shows the URL `sourceforge.net/projects/lpsolve/`. The page header includes the SourceForge logo, a search bar, and navigation links for Browse, Enterprise, Blog, Help, and Jobs. Below the header, there are links for SOLUTION CENTERS, Go Parallel, HTML5, Smarter IT, Resources, and Newsletters. A prominent red banner for "Go Parallel" is visible, with the text "Translating Multicore Power into Application Performance" and "Interviews and blogs by thought leaders keep you ahead of the curve." Below the banner, the breadcrumb trail reads "Home / Browse / Development / Algorithms / lpsolve". The main heading is "lpsolve", followed by "Brought to you by: keikland, peno64". A navigation bar contains links for Summary, Files, Reviews, Support, Wiki, MediaWiki, and News. The "Files" section shows a download button for "lp\_solve\_5.5.2.0\_IDE\_Setup.exe", which is circled in red. Other statistics include 4.7 Stars (141), 748 Downloads (This Week), and a last update date of 2013-04-10. Social media sharing options for Tweet (2), Google+ (6), and Curtir (31) are also present. A "Browse All Files" link is located to the right of the download button. At the bottom, there is a small thumbnail image of the software's installation window.

## Tela inicial lp\_solve:

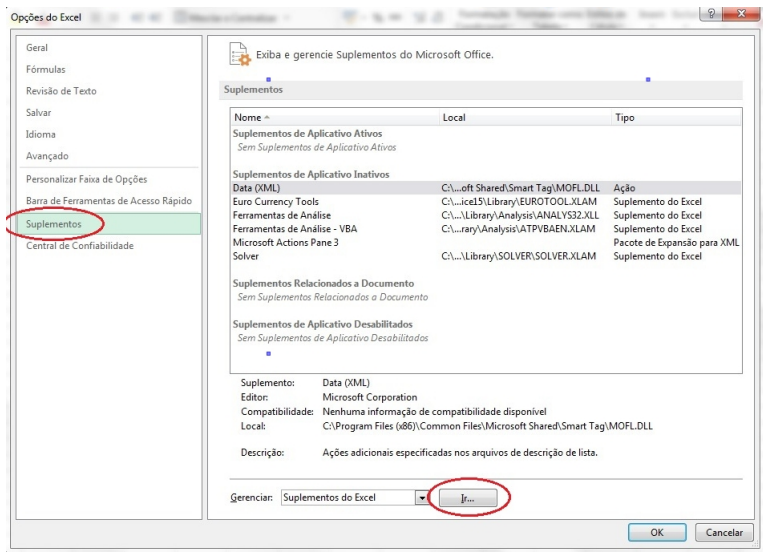


## Para usar o Solver/Excel:

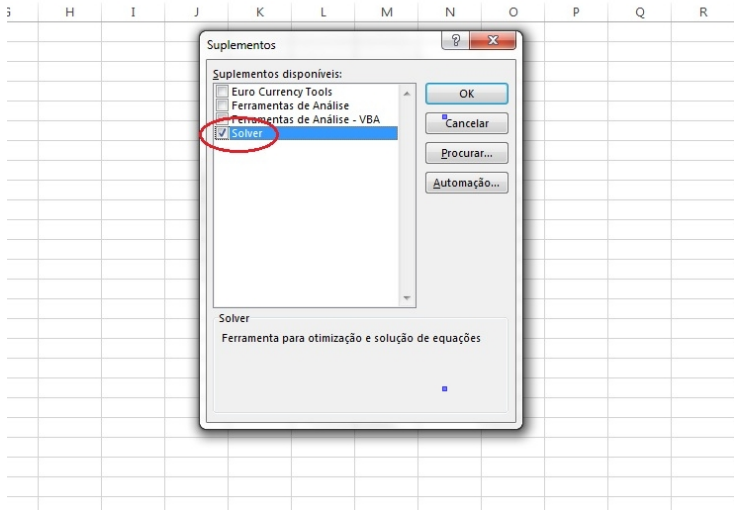




## Caso não encontre, vá em Arquivo &gt;&gt; Opções &gt;&gt; Suplementos &gt;&gt; Ir:



Selecione a opção Solver e clique em OK:



## Solução de problemas com o uso de *software*

- ▶ No canal do YouTube estão disponíveis exemplos de como modelar e resolver pelo `lp_solve` e Solver/Excel alguns dos problemas vistos nesta aula:

<https://www.youtube.com/PedroMunari>

## Linguagem de modelagem algébrica (extra)

- ▶ Para lidar com problemas de grande porte, é importante que o software permita entrar com o modelo usando alguma linguagem de modelagem algébrica.
- ▶ Exemplos: GAMS, AMPL, GMPL, OPL, etc.

## Linguagem de modelagem algébrica (extra)

- ▶ Para lidar com problemas de grande porte, é importante que o software permita entrar com o modelo usando alguma linguagem de modelagem algébrica.
- ▶ Exemplos: GAMS, AMPL, GMPL, OPL, etc.
- ▶ Permitem entrar com o modelo na forma algébrica;

## Linguagem de modelagem algébrica (extra)

- ▶ Para lidar com problemas de grande porte, é importante que o software permita entrar com o modelo usando alguma linguagem de modelagem algébrica.
- ▶ Exemplos: GAMS, AMPL, GMPL, OPL, etc.
- ▶ Permitem entrar com o modelo na forma algébrica;
- ▶ Em geral, devem ser utilizadas dentro de um software de modelagem que tipicamente leva o mesmo nome da linguagem;

## Linguagem de modelagem algébrica (extra)

- ▶ Para lidar com problemas de grande porte, é importante que o software permita entrar com o modelo usando alguma linguagem de modelagem algébrica.
- ▶ Exemplos: GAMS, AMPL, GMPL, OPL, etc.
- ▶ Permitem entrar com o modelo na forma algébrica;
- ▶ Em geral, devem ser utilizadas dentro de um software de modelagem que tipicamente leva o mesmo nome da linguagem;
- ▶ Por exemplo, a linguagem GAMS é usada dentro do software GAMS para modelagem e resolução do problema. O GAMS não resolve o problema, servindo apenas como uma interface que chama outros softwares (p.ex. CPLEX) para a resolução de fato;

## Linguagem de modelagem algébrica (extra)

- ▶ Para lidar com problemas de grande porte, é importante que o software permita entrar com o modelo usando alguma linguagem de modelagem algébrica.
- ▶ Exemplos: GAMS, AMPL, GMPL, OPL, etc.
- ▶ Permitem entrar com o modelo na forma algébrica;
- ▶ Em geral, devem ser utilizadas dentro de um software de modelagem que tipicamente leva o mesmo nome da linguagem;
- ▶ Por exemplo, a linguagem GAMS é usada dentro do software GAMS para modelagem e resolução do problema. O GAMS não resolve o problema, servindo apenas como uma interface que chama outros softwares (p.ex. CPLEX) para a resolução de fato;
- ▶ Algumas podem ser usadas no SolverStudio, dentro do Excel.



# Linguagem de modelagem algébrica

- ▷ GNU Mathematical Programming Language (GMPL)

# Linguagem de modelagem algébrica

## ▷ GNU Mathematical Programming Language (GMPL)

- ▶ GMPL é um linguagem de modelagem algébrica criada pela GNU para interação com o GLPK (GNU Linear Programming Kit);

# Linguagem de modelagem algébrica

## ▷ GNU Mathematical Programming Language (GMPL)

- ▶ GMPL é um linguagem de modelagem algébrica criada pela GNU para interação com o GLPK (GNU Linear Programming Kit);
- ▶ Pode ser usada por meio do software livre Gusek:  
<http://gusek.sourceforge.net/>

# Linguagem de modelagem algébrica

## ▷ GNU Mathematical Programming Language (GMPL)

- ▶ GMPL é um linguagem de modelagem algébrica criada pela GNU para interação com o GLPK (GNU Linear Programming Kit);
- ▶ Pode ser usada por meio do software livre Gusek:  
<http://gusek.sourceforge.net/>
- ▶ Há também uma interface web didática e com exemplos:  
<https://online-optimizer.appspot.com>

# Linguagem de modelagem algébrica

## ▷ GNU Mathematical Programming Language (GMPL)

- ▶ GMPL é um linguagem de modelagem algébrica criada pela GNU para interação com o GLPK (GNU Linear Programming Kit);
- ▶ Pode ser usada por meio do software livre Gusek:  
<http://gusek.sourceforge.net/>
- ▶ Há também uma interface web didática e com exemplos:  
<https://online-optimizer.appspot.com>
- ▶ E pelo suplemento SolverStudio, dentro do Excel;

# Linguagem de modelagem algébrica

## ▷ GNU Mathematical Programming Language (GMPL)

- ▶ GMPL é um linguagem de modelagem algébrica criada pela GNU para interação com o GLPK (GNU Linear Programming Kit);
- ▶ Pode ser usada por meio do software livre Gusek:  
<http://gusek.sourceforge.net/>
- ▶ Há também uma interface web didática e com exemplos:  
<https://online-optimizer.appspot.com>
- ▶ E pelo suplemento SolverStudio, dentro do Excel;
- ▶ É um subconjunto da AMPL, uma linguagem comercial (\$\$\$) bastante utilizada, assim pode também ser usada no software AMPL junto com outros softwares comerciais (p.ex. CPLEX).

# GNU Mathematical Programming Language (GMPL)

## ▷ Exemplo

Uma metalúrgica produz dois tipos de ligas metálicas. Cada liga é composta de proporções diferentes de cobre, zinco e chumbo, os quais estão disponíveis em quantidades limitadas em estoque. Deseja-se determinar quanto produzir de cada liga metálica, de modo a maximizar a receita bruta, satisfazendo-se as seguintes composições das ligas e a disponibilidade de matéria-prima em estoque:

Matéria-prima	Liga 1	Liga 2	Estoque
Cobre	50%	30%	3 ton
Zinco	10%	20%	1 ton
Chumbo	40%	50%	3 ton
Preço venda	3 mil	2 mil	(R\$ por ton)

# GNU Mathematical Programming Language (GMPL)

## ▷ Exemplo

$$\max \quad f(x_1, x_2) = 3x_1 + 2x_2$$

$$\text{s.a} \quad 0,5x_1 + 0,3x_2 \leq 3$$

$$0,1x_1 + 0,2x_2 \leq 1$$

$$0,4x_1 + 0,5x_2 \leq 3$$

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



# GNU Mathematical Programming Language (GMPL)

## ▷ Modelo com dados em GMPL

```
# Variaveis de decisao
var x1 >= 0;
var x2 >= 0;

# Funcao objetivo
maximize Receita: 3 * x1 + 2 * x2;

# Restricoes
Cobre: 0.5 * x1 + 0.3 * x2 <= 3;
Zinco: 0.1 * x1 + 0.2 * x2 <= 1;
Chumbo: 0.4 * x1 + 0.5 * x2 <= 3;

# Resolver o problema
solve;

# Escrever valor otimo e solucao otima
display Receita, x1, x2;
```

# GNU Mathematical Programming Language (GMPL)

## ▷ Gusek

```

F:\Pesquisa Operacional para Eng Prod 2 - 2017\GMPL\ligas_v1.mod - Gusek [2 of 2]
File Edit Search View Tools Options Language Buffers Help
1 diet.mod 2 ligas_v1.mod
1 # Modelo para o problema de ligas metalicas
2
3 # Variaveis de decisao
4 var x1 >= 0;
5 var x2 >= 0;
6
7 # Funcao objetivo
8 maximize Receita: 3 * x1 + 2 * x2;
9
10 # Restricoes
11 Cobre: 0.5 * x1 + 0.3 * x2 <= 3;
12 Zinco: 0.1 * x1 + 0.2 * x2 <= 1;
13 Chumbo: 0.4 * x1 + 0.5 * x2 <= 3;
14
15 # Resolver o problema
16 solve;
17
18 # Escrever valor otimo e solucao otima
19 display Receita, x1, x2;
20
>C:\Users\Pedro\Downloads\gusek_0-2-20\gusek\glpsol.exe --c
GLPSOL: GLPK LP/MIP Solver, v4.60
Parameter(s) specified in the command line:
--cover --clicque --gomory --mir -m ligas_v1.mod
Reading model section from ligas_v1.mod...
ligas_v1.mod:19: warning: unexpected end of file; missing e
19 lines were read
Generating Receita...
Generating Cobre...
Generating Zinco...
Generating Chumbo...
Model has been successfully generated
GLPK Simplex Optimizer, v4.60
4 rows, 2 columns, 8 non-zeros
Preprocessing...
3 rows, 2 columns, 6 non-zeros
Scaling...
A: min|a[ij]| = 1.000e-01 max|a[ij]| = 5.000e-01 ratio =
Problem data seem to be well scaled
Constructing initial basis...
Size of triangular part is 3
* 0: obj = -0.00000000e+00 inf = 0.000e+00 (2)
* 2: obj = 1.846153846e+01 inf = 0.000e+00 (0)
OPTIMAL LP SOLUTION FOUND
Time used: 0.0 secs
Memory used: 0.1 Mb (96621 bytes)
Display statement at line 19
Receita.val = 18.4615384615385
x1.val = 4.61538461538462
x2.val = 2.30769230769231
Model has been successfully processed
>Exit code: 0 Time: 0.218
Cursor: line[20] col[1] Selection: [0]lines [0]chars [INS] [CR+LF] File: ligas_v1.mod, 20 lines

```

# GNU Mathematical Programming Language (GMPL)

## ▷ Exemplo

- ▶ A modelagem anterior explicita os dados diretamente no modelo, semelhante ao que é feito no `lp_solve`;

# GNU Mathematical Programming Language (GMPL)

## ▷ Exemplo

- ▶ A modelagem anterior explicita os dados diretamente no modelo, semelhante ao que é feito no `lp_solve`;
- ▶ Embora o GMPL permita isso, esta **não** é a forma ideal de usar a linguagem;

# GNU Mathematical Programming Language (GMPL)

## ▷ Exemplo

- ▶ A modelagem anterior explicita os dados diretamente no modelo, semelhante ao que é feito no `lp_solve`;
- ▶ Embora o GMPL permita isso, esta **não** é a forma ideal de usar a linguagem;
- ▶ O potencial da GMPL está em permitir a definição do *modelo algébrico*, **sem os dados** de um problema específico;

# GNU Mathematical Programming Language (GMPL)

## ▷ Exemplo

- ▶ A modelagem anterior explicita os dados diretamente no modelo, semelhante ao que é feito no `lp_solve`;
- ▶ Embora o GMPL permita isso, esta **não** é a forma ideal de usar a linguagem;
- ▶ O potencial da GMPL está em permitir a definição do *modelo algébrico*, **sem os dados** de um problema específico;
- ▶ Vamos então definir um modelo algébrico para o exemplo.

# GNU Mathematical Programming Language (GMPL)

▷ Exemplo: modelo algébrico

▶ Ligas:  $L = \{1, \dots, n\}$ ;

# GNU Mathematical Programming Language (GMPL)

## ▷ Exemplo: modelo algébrico

- ▶ Ligas:  $L = \{1, \dots, n\}$ ;
- ▶ Matérias-primas:  $M = \{1, \dots, m\}$ .



# GNU Mathematical Programming Language (GMPL)

## ▷ Exemplo: modelo algébrico

- ▶ Ligas:  $L = \{1, \dots, n\}$ ;
- ▶ Matérias-primas:  $M = \{1, \dots, m\}$ .

$$\begin{aligned} \max \quad & f(x) = \sum_{i \in L} r_i x_i \\ \text{s.a} \quad & \sum_{i \in L} a_{ji} x_i \leq b_j, \quad \forall j \in M \\ & x_i \geq 0, \quad \forall i \in L. \end{aligned}$$

# GNU Mathematical Programming Language (GMPL)

## ▷ Exemplo: modelo algébrico

```
# Conjuntos de indices
set Ligas;
set MPrimas;

# Parametros (dados de entrada)
param receita{Ligas};
param estoque{MPrimas};
param composicao{MPrimas, Ligas};

# Variaveis de decisao
var x{Ligas} >= 0;

# Funcao objetivo
maximize ReceitaTotal: sum{i in Ligas} receita[i] * x[i];

# Restricoes
estoques{j in MPrimas}: sum{i in Ligas} composicao[j,i] * x[i] <= estoque[j];

# Resolver o problema
solve;
```

# GNU Mathematical Programming Language (GMPL)

## ▷ Exemplo: modelo algébrico

```
# Escrever valor otimo e solucao otima
display ReceitaTotal, x;

# Dados de uma instância do problema
data;

set Ligas := 1 2;
set MPrimas := Cobre Zinco Chumbo;

param: receita :=
1 3
2 2;

param: estoque :=
Cobre 3
Zinco 1
Chumbo 3;

param composicao : 1 2 :=
Cobre 0.5 0.3
Zinco 0.1 0.2
Chumbo 0.4 0.5;
end;
```

# GNU Mathematical Programming Language (GMPL)

## ▷ Seções de um arquivo GMPL

- ▶ Definição dos conjuntos de índices, usando o comando `set`:  
`set Ligas;`

# GNU Mathematical Programming Language (GMPL)

## ▷ Seções de um arquivo GMPL

- ▶ Definição dos conjuntos de índices, usando o comando `set`:  
`set Ligas;`
- ▶ Definição de parâmetros, usando o comando `param`:  
`param composicao{MPrimas, Ligas};`

# GNU Mathematical Programming Language (GMPL)

## ▷ Seções de um arquivo GMPL

- ▶ Definição dos conjuntos de índices, usando o comando `set`:  
`set Ligas;`
- ▶ Definição de parâmetros, usando o comando `param`:  
`param composicao{MPrimas, Ligas};`
- ▶ Definição de variáveis, usando o comando `var`:  
`var x{Ligas} >= 0;`  
`var y{Produtos} >= 0, integer;`  
`var w{Veiculos} binary;`

# GNU Mathematical Programming Language (GMPL)

## ▷ Seções de um arquivo GMPL

- ▶ Definição dos conjuntos de índices, usando o comando `set`:  
`set Ligas;`
- ▶ Definição de parâmetros, usando o comando `param`:  
`param composicao{MPrimas, Ligas};`
- ▶ Definição de variáveis, usando o comando `var`:  
`var x{Ligas} >= 0;`  
`var y{Produtos} >= 0, integer;`  
`var w{Veiculos} binary;`
- ▶ Definição da função objetivo, usando o comando `maximize` ou `minimize`:  
`maximize ReceitaTotal: sum{i in Ligas} receita[i] * x[i];`

# GNU Mathematical Programming Language (GMPL)

## ▷ Seções de um arquivo GMPL

- ▶ Definição de restrições, com o domínio e nome (opcional):

```
estoques{j in MPrimas}: sum{i in Ligas} composicao[j,i] *  
x[i] <= estoque[j];
```



# GNU Mathematical Programming Language (GMPL)

## ▷ Seções de um arquivo GMPL

- ▶ Definição de restrições, com o domínio e nome (opcional):

```
estoques{j in MPrimas}: sum{i in Ligas} composicao[j,i] *  
x[i] <= estoque[j];
```

- ▶ Resolução do problema, por meio do comando solve;

# GNU Mathematical Programming Language (GMPL)

## ▷ Seções de um arquivo GMPL

- ▶ Definição de restrições, com o domínio e nome (opcional):

```
estoques{j in MPrimas}: sum{i in Ligas} composicao[j,i] *  
x[i] <= estoque[j];
```

- ▶ Resolução do problema, por meio do comando solve;

- ▶ Impressão de dados e resultados, por meio do comando display;

```
display ReceitaTotal, x;  
display {i in Tarefas, k in Estacoes: x[i,k] > 0} x[i,k];
```

# GNU Mathematical Programming Language (GMPL)

## ▷ Seções de um arquivo GMPL

- ▶ Definição de restrições, com o domínio e nome (opcional):  
`estoques{j in MPrimas}: sum{i in Ligas} composicao[j,i] *  
x[i] <= estoque[j];`
- ▶ Resolução do problema, por meio do comando `solve`;
- ▶ Impressão de dados e resultados, por meio do comando `display`;  
`display ReceitaTotal, x;  
display {i in Tarefas, k in Estacoes: x[i,k] > 0} x[i,k];`
- ▶ Entrada de dados da instância, por meio do comando `data`;

# GNU Mathematical Programming Language (GMPL)

## ▷ Exemplo: modelo algébrico

```
# Dados de uma instância do problema
data;

set Ligas := 1 2;
set MPrimas := Cobre Zinco Chumbo;

param: receita :=
1 3
2 2;

param: estoque :=
Cobre 3
Zinco 1
Chumbo 3;

param composicao : 1 2 :=
Cobre 0.5 0.3
Zinco 0.1 0.2
Chumbo 0.4 0.5;
end;
```

# GNU Mathematical Programming Language (GMPL)

## ▷ Arquivos `.mod` e `.dat`

- ▶ O modelo em GMPL deve ser salvo no formato `.mod`;
- ▶ Podemos fazer arquivos separados para o modelo algébrico (`.mod`) e para os dados de um exemplar (instância) específico (`.dat`);
- ▶ Essa separação é importante, principalmente em projetos grandes, e permite resolver diferentes instâncias sem alterar o modelo.

# GNU Mathematical Programming Language (GMPL)

## ▷ SolverStudio

- ▶ O modelo em GMPL pode ser resolvido pelo Gurobi ou usando qualquer outro software de otimização capaz de ler o formato GMPL ou AMPL;
- ▶ Uma alternativa interessante e que permite a integração com planilhas é o SolverStudio, um suplemente para o Excel disponível em:  
[www.solverstudio.org](http://www.solverstudio.org);
- ▶ É uma ferramenta livre, gratuita e muito útil para o desenvolvimento de aplicações de PO integradas ao Excel;
- ▶ Permite integração com VBA para automatizar a resolução.

# GNU Mathematical Programming Language (GMPL)

## ▷ SolverStudio

The screenshot displays the SolverStudio interface, which is overlaid on an Excel spreadsheet. The Excel spreadsheet contains data for a linear programming problem, including ligament types (Liga1, Liga2), their composition (Cobre, Zinco, Chumbo), and stock levels (Estoque). The SolverStudio window shows the GMPL model for this problem, including the objective function and constraints.

**Excel Spreadsheet Data:**

	A	B	C	D	E	F	G
1							
2			Liga1	Liga2			
3							
4			Solução				
5							
6			Receita	3	2		
7							
8				Composição	Estoque		
9			Cobre	0,5	0,3	3	
10			Zinco	0,1	0,2	1	
11			Chumbo	0,4	0,5	3	
12							
13			Valor Otimo				
14							
15							
16							
17							
18							
19							
20							
21							
22							
23							

**SolverStudio GMPL Model:**

```

# Modelo para o problema de ligas metalicas (v2.0)

# Conjuntos de indices
set Ligas;
set MPrimas;

# Parametros (dados de entrada)
param receita(Ligas);
param estoque(MPrimas);
param composicao(MPrimas, Ligas);

# Variaveis de decisao
var x(Ligas) >= 0;

# Funcao objetivo
maximize ReceitaTotal: sum(i in Ligas) receita[i] * x[i];
  
```

# GNU Mathematical Programming Language (GMPL)

## ▷ SolverStudio

- ▶ Para ser usado no SolverStudio, o modelo em GMPL não deve conter a seção 'data';
- ▶ Todos os dados devem ser inseridos na planilha e associados ao modelo por meio do botão 'Edit data' da barra de ferramentas do SolverStudio;
- ▶ No formulário que se abre após clicar neste botão, devem ser definidos todos os conjuntos (*sets*) e parâmetros, bem como as variáveis e função objetivo (com o mesmo nome em que são definidos no arquivo GMPL);



# GNU Mathematical Programming Language (GMPL)

## ▷ SolverStudio

The screenshot shows the SolverStudio interface overlaid on an Excel spreadsheet. The spreadsheet contains a linear programming model with the following data:

	A	B	C	D
1				
2			Liga1	Liga2
3				
4		Solução		
5				
6		Receita	3	2
7				
8			Composição	
9		Cobre	0,5	0,3
10		Zinco	0,1	0,2
11		Chumbo	0,4	0,5
12				
13		Valor Otimo		
14				
15				
16				
17				
18				
19				
20				
21				
22				
23				

The SolverStudio - Data Items Editor dialog is open, showing the following table of data items:

Name:	Cell Range:	Index Range(s):	Value if Missing:
<Add New Data Item>			
composicao	C9:D11	MPrimas, Ligas	<Report an error>
estoque	E9:E11	MPrimas	<Report an error>
Ligas	C2:D2		
MPrimas	B9:B11		
receita	C6:D6	Ligas	<Report an error>
ReceitaTotal	C13		
x	C4:D4	Ligas	<Report an error>

At the bottom of the dialog, there are input fields for Name, Cell Range (set to \$C\$13), Index Range(s), and Unknown Indices return (set to an error). Buttons for Delete Data Item, Add Data Item, and Cancel are also present.

This dialog allows you to define cell ranges (data items) on your spreadsheet that you want to use in your optimisation model. You can define sets of items, indexed lists (a cell range containing one entry for each value in an associated index range) and indexed tables (which need 2 index ranges).

# GNU Mathematical Programming Language (GMPL)

## ▷ SolverStudio

- ▶ Ao final do arquivo GMPL, devem ser inseridos comandos 'printf', para que a solução obtida seja escrita na planilha;
- ▶ Para o exemplo das ligas, temos:

```
# Escrever valor otimo NA PLANILHA
printf "ReceitaTotal = " >> "Sheet";
printf ReceitaTotal >> "Sheet";
printf "\n" >> "Sheet";

# Escrever solucao otima NA PLANILHA
printf "x :=\n" >> "Sheet";
printf {i in Ligas}: "'s' '%s'\n", i, x[i] >> "Sheet";
printf ";\n" >> "Sheet";
```

# GNU Mathematical Programming Language (GMPL)

## ▷ SolverStudio

The screenshot displays the SolverStudio interface overlaid on an Excel spreadsheet. The spreadsheet contains data for a linear programming problem, including objective function coefficients, constraint coefficients, and resource limits.

ReceitaTo...	A	B	C	D	E	F	G
	1						
	2		Liga1	Liga2			
	3						
	4	Solução					
	5						
	6	Receita	3	2			
	7						
	8		Composição		Estoque		
	9	Cobre	0,5	0,3	3		
	10	Zinco	0,1	0,2	1		
	11	Chumbo	0,4	0,5	3		
	12						
	13	Valor Otimo					
	14						
	15						
	16						
	17						
	18						
	19						
	20						
	21						
	22						
	23						

The SolverStudio window shows the following GMPL code:

```

solve:
# Escrever valor otimo e solucao otima
display ReceitaTotal, x;

# Escrever valor otimo NA PLANILHA
printf "ReceitaTotal = " >> "Sheet";
printf ReceitaTotal >> "Sheet";
printf "\n" >> "Sheet";

# Escrever solucao otima NA PLANILHA
printf "x :=\n" >> "Sheet";
printf (i in Ligas): "%s' '%s'\n", i, x[i] >> "Sheet";
printf "\n" >> "Sheet";
  
```

The SolverStudio window also displays the "Model Output" section, which is currently empty. The status bar at the bottom right indicates "Row 12, Column 22".

# GNU Mathematical Programming Language (GMPL)

## ▷ SolverStudio

The screenshot displays the SolverStudio interface integrated with Microsoft Excel. The Excel spreadsheet, titled 'ligas\_v2 - Excel', contains the following data:

	A	B	C	D	E	F	G
1							
2			Liga1	Liga2			
3							
4		Solução	4,615385	2,307692			
5							
6		Receita	3	2			
7							
8			Composição	Estoque			
9		Cobre	0,5	0,3	3		
10		Zinco	0,1	0,2	1		
11		Chumbo	0,4	0,5	3		
12							
13		Valor Otimo	18,46154				
14							
15							
16							
17							
18							
19							
20							
21							
22							
23							

The SolverStudio window, titled 'SolverStudio ©Andrew Mason', shows the following GMPL code:

```

solve:
# Escrever valor otimo e solucao otima
display ReceitaTotal, x;

# Escrever valor otimo NA PLANILHA
printf "ReceitaTotal = " >> "Sheet";
printf ReceitaTotal >> "Sheet";
printf "\n" >> "Sheet";

# Escrever solucao otima NA PLANILHA
printf "x :=\n" >> "Sheet";
printf (i in Ligas): "%s" "%s"\n", i, x[i] >> "Sheet";
printf "\n" >> "Sheet";
  
```

The Model Output window shows the following text:

```

Model Output
Model has been successfully processed

### GMPL run completed.
### Results loaded for data items: ReceitaTotal x*
### (*=data item values changed on sheet)
### Done
  
```

The SolverStudio window also displays the SolverStudio logo and the name 'Andrew Mason'.

- ▶ Obrigado pela atenção!
- ▶ Dúvidas?