MODELAGEM DE PROBLEMAS DE OTIMIZAÇÃO COM PYTHON E PYOMO

Prof. Esp. Antonio C. da Silva Júnior

UFPR
UNIVERSIDADE FEDERAL DO PARANÁ

Dezembro/2021

Conteúdo

- O que é o Pyomo?
- Componentes necessários
- Requisitos para modelagem no Pyomo
- Elementos do Pyomo
- Modelo abstrato x Modelo concreto
- Exemplos práticos

UFPR
UNIVERSIDADE FEDERAL DO PARANÁ

O que é o Pyomo?

- Biblioteca para modelagem de problemas de otimização em Python
- Sintaxe natural na representação de modelos matemáticos
- Facilidade para representar grandes modelos
- Isolação do código do modelo com relação aos dados de entrada
- Necessita de um solver

Componentes necessários

Componentes necessários

- Editor de código
- Python
- Pyomo
- Solver

Componentes necessários

- Editor de código
- Python
- Pyomo
- Solver

ANACONDA®

- Criação do ambiente virtual
- Instalação do Jupyter Notebook

Componentes necessários

- Editor de código
- Python
- Pyomo
- Solver

Componentes necessários

- Editor de código
- Python
- Pyomo
- Solver

GLPK

(GNU Linear Programming Kit)

GUROBI
OPTIMIZATION

Componentes necessários

- Editor de código
- Python
- Pyomo
- Solver

GLPK

(GNU Linear Programming Kit)

GUROBI
OPTIMIZATION

- [Gurobi Optimizer (sistema)]
- [Gurobi (Anaconda)]

Componentes necessários

- Login > Downloads & Licenses > Academic License
- Aceitar os termos de uso
- Instalar a licença via prompt ou terminal

Requisitos para modelagem no Pyomo

Lidando com vetores e matrizes

$$x = 1$$

Lidando com vetores e matrizes

$$x = 1$$

```
In [1]: x = 1
```

Lidando com vetores e matrizes

$$x = 1$$

```
In [1]: x = 1

In [2]: x
Out[2]: 1
```

Lidando com vetores e matrizes

$$\mathbf{x} = \begin{bmatrix} 10 \\ 20 \\ 30 \\ 40 \end{bmatrix}$$

Lidando com vetores e matrizes

$$\mathbf{x} = \begin{bmatrix} 10 \\ 20 \\ 30 \\ 40 \end{bmatrix}$$

```
In [3]: x = [10, 20, 30, 40]

In [4]: x

Out[4]: [10, 20, 30, 40]
```

Lidando com vetores e matrizes

x_1

```
In [5]: x[0]
Out[5]: 10
```

$$\mathbf{x} = \begin{bmatrix} 10 \\ 20 \\ 30 \\ 40 \end{bmatrix}$$

Lidando com vetores e matrizes

x_3

```
In [6]: x[2]
Out[6]: 30
```

$$\mathbf{x} = \begin{bmatrix} 10 \\ 20 \\ 30 \\ 40 \end{bmatrix}$$

Lidando com vetores e matrizes

$I \colon \{1, 2, \dots, m\}$

Lidando com vetores e matrizes

$I: \{1, 2, \dots, m\}$

```
In [7]: m = len(x)

In [8]: m

Out[8]: 4
```

Lidando com vetores e matrizes

$$I: \{1, 2, \dots, m\}$$

```
In [7]: m = len(x)
```

```
In [8]: m
```

```
Out[8]: 4
```

```
In [9]: I = range(m)
```

Lidando com vetores e matrizes

$$I\colon \{1, 2, \ldots, m\}$$

```
In [7]: m = len(x)

In [8]: m
Out[8]: 4

In [9]: I = range(m)

In [10]: [i for i in I]
Out[10]: [0, 1, 2, 3]
```

Lidando com vetores e matrizes

$$x_i \quad \forall i \in I$$

Lidando com vetores e matrizes

$$x_i \quad \forall i \in I$$

```
In [11]: [x[i] for i in I]
Out[11]: [10, 20, 30, 40]
```

$$\mathbf{x} = \begin{bmatrix} 10 \\ 20 \\ 30 \\ 40 \end{bmatrix}$$

Lidando com vetores e matrizes

```
In [12]: i = 0
```

```
In [13]: x[i]
Out[13]: 10
```

$$\mathbf{x} = \begin{bmatrix} 10 \\ 20 \\ 30 \\ 40 \end{bmatrix}$$

Lidando com vetores e matrizes

$$\sum_{i \in I} x_i$$

Lidando com vetores e matrizes

$$\sum_{i \in I} x_i$$

```
In [14]: sum(x)
Out[14]: 100
```

Lidando com vetores e matrizes

$$\sum_{i \in I} x_i$$

```
In [14]: sum(x)
Out[14]: 100
```

```
In [15]: sum(x[i] for i in I)
Out[15]: 100
```

$$\mathbf{x} = \begin{bmatrix} 10 \\ 20 \\ 30 \\ 40 \end{bmatrix}$$

Lidando com vetores e matrizes

$$\sum_{i \in I} x_i, \quad \forall i > 2$$

Lidando com vetores e matrizes

$$\sum_{i \in I} x_i, \quad \forall i > 2$$

In [16]:
```
sum(x[i] for i in I if i > 1)
```
Out[16]: 70

$$\mathbf{x} = \begin{bmatrix} 10 \\ 20 \\ 30 \\ 40 \end{bmatrix}$$

Lidando com vetores e matrizes

$$\mathbf{x} = \begin{bmatrix} 10 & 20 & 30 \\ 40 & 50 & 60 \end{bmatrix}$$

Lidando com vetores e matrizes

$$\mathbf{x} = \begin{bmatrix} 10 & 20 & 30 \\ 40 & 50 & 60 \end{bmatrix}$$

In [17]:
```
x = [[10, 20, 30],
     [40, 50, 60]]
```

Lidando com vetores e matrizes

$$\mathbf{x} = \begin{bmatrix} 10 & 20 & 30 \\ 40 & 50 & 60 \end{bmatrix}$$

```
In [17]: x = [[10, 20, 30],
              [40, 50, 60]]

In [18]: x

Out[18]: [[10, 20, 30], [40, 50, 60]]
```

Lidando com vetores e matrizes

x_{11}

In [19]:
```
x[0][0]
```
Out[19]: 10

$$\mathbf{x} = \begin{bmatrix} 10 & 20 & 30 \\ 40 & 50 & 60 \end{bmatrix}$$

Lidando com vetores e matrizes

x_{13}

In [20]:
```
x[0][2]
```
Out[20]:
```
30
```

$$\mathbf{x} = \begin{bmatrix} 10 & 20 & 30 \\ 40 & 50 & 60 \end{bmatrix}$$

Lidando com vetores e matrizes

$I: \{1, \dots, m\}$

$J: \{1, \dots, n\}$

Lidando com vetores e matrizes

$I: \{1, \ldots, m\}$

$J: \{1, \ldots, n\}$

In [21]:
```
m = len(x)
```

In [22]:
```
m
```

Out[22]:
```
2
```

Lidando com vetores e matrizes

$I: \{1, \dots, m\}$

$J: \{1, \dots, n\}$

```
In [21]: m = len(x)

In [22]: m
Out[22]: 2

In [23]: n = len(x[0])

In [24]: n
Out[24]: 3
```

Lidando com vetores e matrizes

```
In [25]: I = range(m)

In [26]: [i for i in I]
Out[26]: [0, 1]
```

Lidando com vetores e matrizes

```
In [25]: I = range(m)

In [26]: [i for i in I]
Out[26]: [0, 1]

In [27]: J = range(n)

In [28]: [j for j in J]
Out[28]: [0, 1, 2]
```

Lidando com vetores e matrizes

$$x_{ij} \quad \forall i \in I, \forall j \in J$$

Lidando com vetores e matrizes

$$x_{ij} \quad \forall i \in I, \forall j \in J$$

In [29]:
```
[x[i][j] for i in I for j in J]
```

Out[29]:
```
[10, 20, 30, 40, 50, 60]
```

$$\mathbf{x} = \begin{bmatrix} 10 & 20 & 30 \\ 40 & 50 & 60 \end{bmatrix}$$

Lidando com vetores e matrizes

$$x_{ij} \quad \forall j \in J, \forall i \in I$$

Lidando com vetores e matrizes

$$x_{ij} \quad \forall j \in J, \forall i \in I$$

```
In [30]: [x[i][j] for j in J for i in I]
Out[30]: [10, 40, 20, 50, 30, 60]
```

$$\mathbf{x} = \begin{bmatrix} 10 & 20 & 30 \\ 40 & 50 & 60 \end{bmatrix}$$

Lidando com vetores e matrizes

$$\sum_{i \in I} x_{ij}, \quad \forall j \in J$$

Lidando com vetores e matrizes

$$\sum_{i \in I} x_{ij}, \quad \forall j \in J$$

In [31]:
```
for j in J:
    print(sum(x[i][j] for i in I))
```

```
50
70
90
```

$$\mathbf{x} = \begin{bmatrix} 10 & 20 & 30 \\ 40 & 50 & 60 \end{bmatrix}$$

Lidando com vetores e matrizes

$$\sum_{j \in J} x_{ij}, \quad \forall i \in I$$

Lidando com vetores e matrizes

$$\sum_{j \in J} x_{ij}, \quad \forall i \in I$$

```
In [32]: for i in I:
             print(sum(x[i][j] for j in J))
```

```
60
150
```

$$\mathbf{x} = \begin{bmatrix} 10 & 20 & 30 \\ 40 & 50 & 60 \end{bmatrix}$$

Lidando com funções

```
In [33]: def calcular_dobro(x):
             return x*2
```

Lidando com funções

```
In [33]: def calcular_dobro(x):
             return x*2

In [34]: calcular_dobro(10)

Out[34]: 20
```

Lidando com funções

```
In [33]: def calcular_dobro(x):
             return x*2

In [34]: calcular_dobro(10)

Out[34]: 20

In [35]: calcular_dobro2 = lambda x: x*2
```

Lidando com funções

```
In [33]: def calcular_dobro(x):
             return x*2

In [34]: calcular_dobro(10)

Out[34]: 20

In [35]: calcular_dobro2 = lambda x: x*2

In [36]: calcular_dobro2(10)

Out[36]: 20
```

Lidando com funções

```
In [37]: def retornar_item(v, i):
             return v[i-1]
```

Lidando com funções

```
In [37]: def retornar_item(v, i):
             return v[i-1]

In [38]: x = [10,20,30,40]
```

Lidando com funções

```
In [37]: def retornar_item(v, i):
             return v[i-1]

In [38]: x = [10,20,30,40]

In [39]: retornar_item(x, 1)

Out[39]: 10
```

Lidando com funções

```
In [37]: def retornar_item(v, i):
             return v[i-1]

In [38]: x = [10,20,30,40]

In [39]: retornar_item(x, 1)

Out[39]: 10

In [40]: retornar_item(x, 2)

Out[40]: 20
```

Lidando com funções

```
In [41]: retornar_item2 = lambda v,i: v[i-1]
```

Lidando com funções

```
In [41]: retornar_item2 = lambda v,i: v[i-1]

In [42]: retornar_item2(x, 1)

Out[42]: 10
```

Lidando com funções

```
In [41]: retornar_item2 = lambda v,i: v[i-1]

In [42]: retornar_item2(x, 1)

Out[42]: 10

In [43]: retornar_item2(x, 2)

Out[43]: 20
```

Elementos do Pyomo

UFPR
UNIVERSIDADE FEDERAL DO PARANÁ

Elementos do Pyomo

Modelo

- ConcreteModel()
- AbstractModel()

Elementos do Pyomo

Modelo

- ConcreteModel()
- AbstractModel()

Conjuntos

- RangeSet()

Elementos do Pyomo

Modelo

- ConcreteModel()
- AbstractModel()

Conjuntos

- RangeSet()

Parâmetros

- Param()

Elementos do Pyomo

Modelo

- ConcreteModel()
- AbstractModel()

Conjuntos

- RangeSet()

Parâmetros

- Param()

Variáveis

- Var()

Elementos do Pyomo

Modelo

- ConcreteModel()
- AbstractModel()

Conjuntos

- RangeSet()

Parâmetros

- Param()

Variáveis

- Var()

Função objetivo

- Objective()

Elementos do Pyomo

Modelo

- ConcreteModel()
- AbstractModel()

Conjuntos

- RangeSet()

Parâmetros

- Param()

Variáveis

- Var()

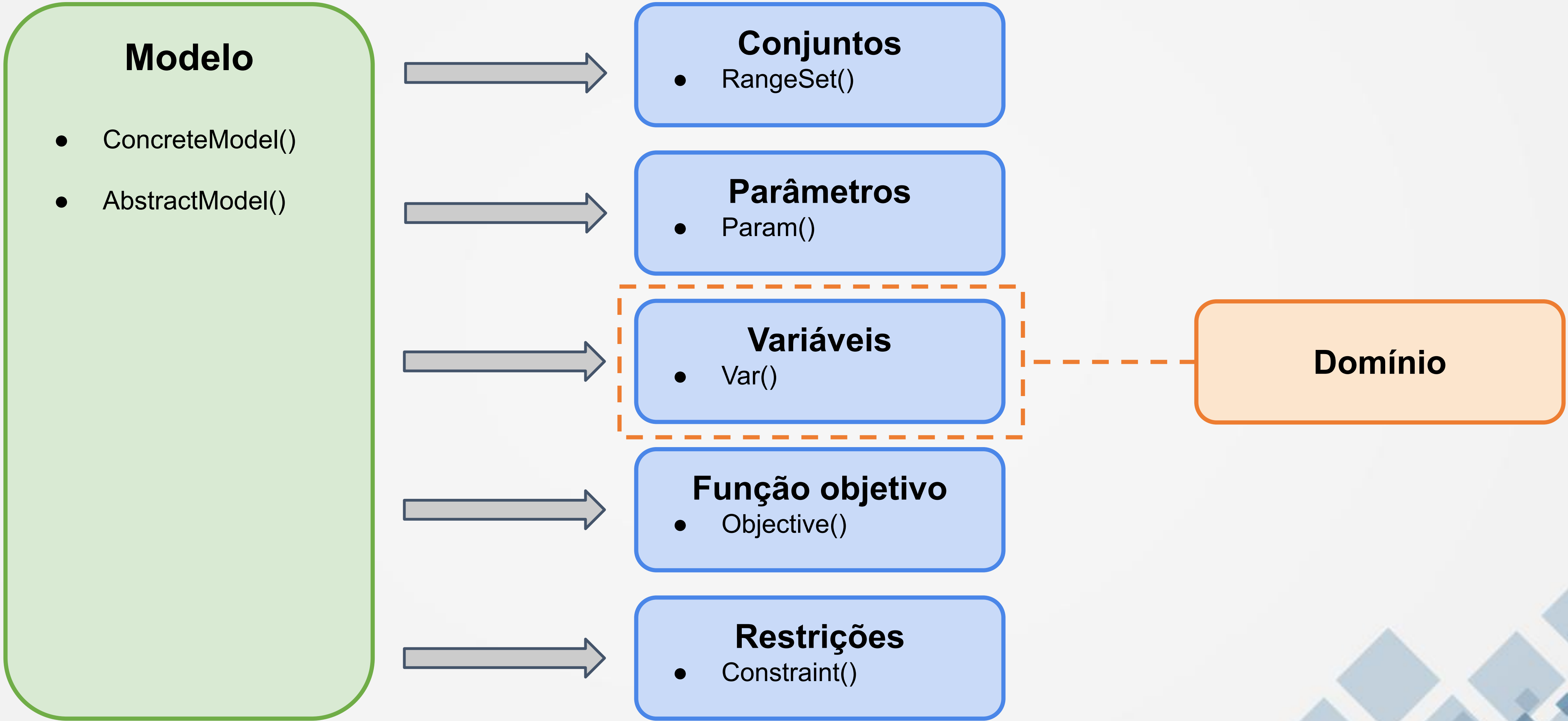
Função objetivo

- Objective()

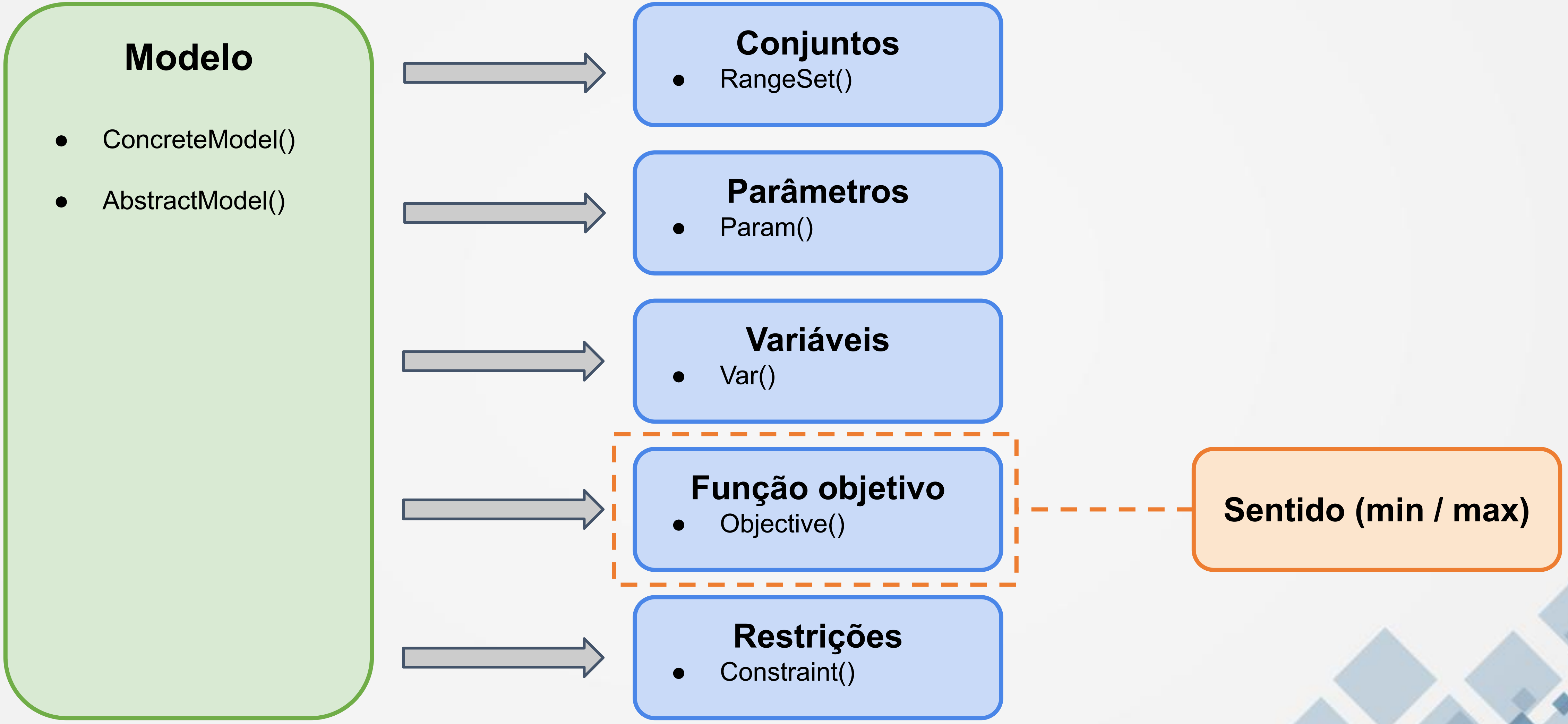
Restrições

- Constraint()

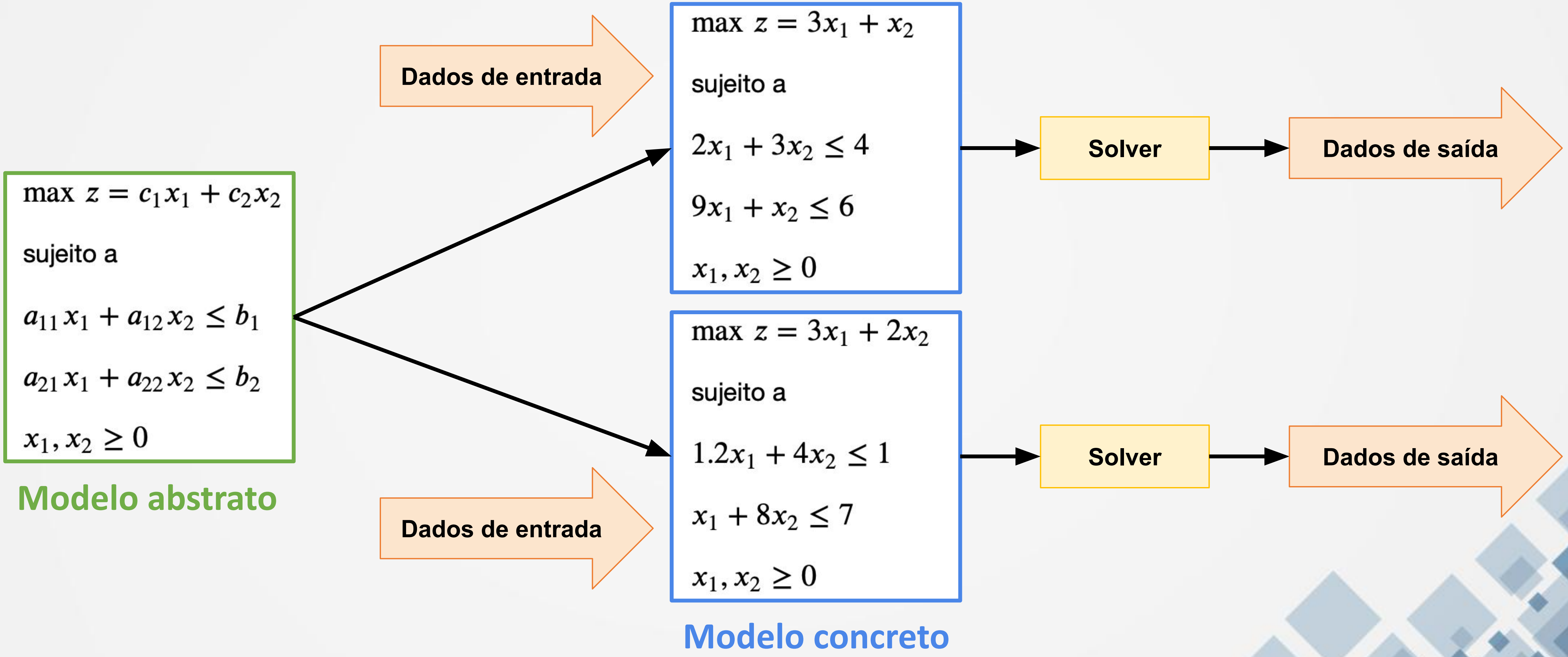
Elementos do Pyomo



Elementos do Pyomo



Modelo abstrato x Modelo concreto



Exemplos práticos

Problema do mix de produção

(Belfiore e Fávero) A empresa Venix de brinquedos está revendo seu planejamento de produção de carrinhos e triciclos. O lucro líquido por unidade de carrinho e triciclo produzido é de **R$12** e **R$60**, respectivamente. As matérias primas e os insumos necessários para a fabricação de cada um dos produtos são terceirizados, cabendo à empresa os processos de usinagem, pintura e montagem. O processo de **usinagem** requer **15 minutos** de mão de obra especializada por unidade de carrinho e **30 minutos** por unidade de triciclo produzida. O processo de **pintura** requer **6 minutos** de mão de obra especializada por unidade de carrinho e **45 minutos** por unidade de triciclo produzida. Já o processo de **montagem** necessita de **6 minutos** e **24 minutos** para uma unidade de carrinho e de triciclo produzida, respectivamente. O tempo disponível por semana é de **36**, **22** e **15** horas para os processos de usinagem, pintura e montagem, respectivamente.

A empresa quer determinar quanto produzir de cada produto por semana, respeitando as limitações de recursos, de forma a maximizar o lucro líquido semanal. Formular o problema de programação linear que maximiza o lucro líquido da empresa Venix.

R$	Carrinho	Triciclo
Lucro	12	60

Horas	Carrinho	Triciclo	Disponib.
Usinagem	0,25	0,5	36
Pintura	0,1	0,75	22
Montagem	0,1	0,4	15

Problema do mix de produção

Variáveis de decisão:

x_1 : quantidade de carrinhos a ser fabricada por semana

x_2 : quantidade de triciclos a ser fabricada por semana

Função objetivo:

$$\max z = 12x_1 + 60x_2$$

Restrições:

$$0,25x_1 + 0,5x_2 \leq 36$$

$$0,1x_1 + 0,75x_2 \leq 22$$

$$0,1x_1 + 0,4x_2 \leq 15$$

$$x_1, x_2 \geq 0$$

R$	Carrinho	Triciclo
Lucro	12	60

Horas	Carrinho	Triciclo	Disponib.
Usinagem	0,25	0,5	36
Pintura	0,1	0,75	22
Montagem	0,1	0,4	15

Problema do mix de produção

Índices / conjuntos:

I: Conjunto de processos, $\{1, \ldots, m\}$

J: Conjunto de produtos, $\{1, \ldots n\}$

Parâmetros:

c_j: lucro líquido por unidade do produto $j \in J$ produzido

a_{ij}: quantidade de horas necessárias para a execução do processo $i \in I$ na fabricação do produto $j \in J$

b_i: quantidade de horas disponíveis para execução do processo $i \in I$

Variáveis de decisão:

x_j: quantidade do produto $j \in J$ a ser fabricada por semana

Problema do mix de produção

Modelo matemático:

$$\max z = \sum_{j \in J} c_j x_j$$

sujeito a:

$$\sum_{j \in J} a_{ij} x_j \leq b_i, \quad \forall\, i \in I$$

$$x_j \geq 0 \quad \forall j \in J$$

Problema do transporte

(Belfiore e Fávero) A Karpet Ltda é uma empresa fabricante de autopeças, cujas sedes estão localizadas em **Osasco, Sorocaba e São Sebastião**. Seus clientes encontram-se em **São Paulo, Rio de Janeiro e Curitiba**. Os custos unitários de transporte de cada origem para cada destino, assim como a capacidade de cada fornecedor e a demanda de cada cliente, encontram-se na tabela abaixo.

O objetivo é atender a demanda de cada consumidor final, respeitando as capacidades de fornecimento, de forma a minimizar o custo total de transporte. Modelar o problema de transporte.

	São Paulo	**Rio de Janeiro**	**Curitiba**	**Capacidade**
Osasco	R$ 12	R$ 22	R$ 30	**100**
Sorocaba	R$ 18	R$ 24	R$ 32	**140**
São Sebastião	R$ 22	R$ 15	R$ 34	**160**
Demanda	**120**	**130**	**150**	

Problema do transporte

Índices / Conjuntos

I: Conjunto de fornecedores, $\{1, 2, \dots, m\}$

J: Conjunto de consumidores, $\{1, 2, \dots, n\}$

Parâmetros

c_{ij}: Custo unitário de transporte do fornecedor $i \in I$ para o consumidor $j \in J$

a_i: Capacidade de abastecimento do fornecedor $i \in I$

b_j: Demanda do consumidor $j \in J$

Variáveis de decisão

x_{ij}: Quantidades transportadas do fornecedor $i \in I$ para o consumidor $j \in J$

Problema do transporte

Formulação matemática

$$\min z = \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij}$$

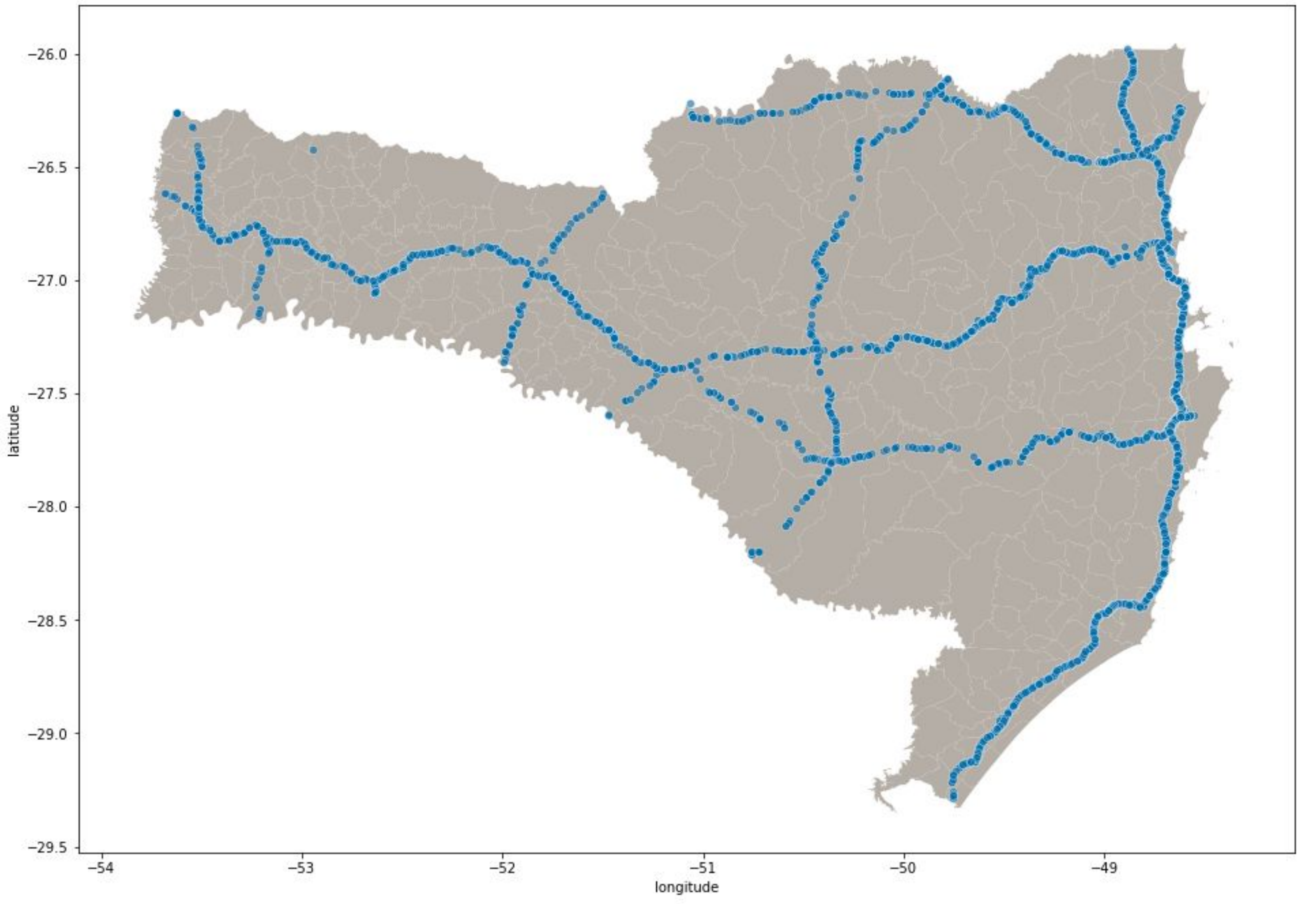
sujeito a

$$\sum_{j \in J} x_{ij} \leq a_i, \forall\, i \in I$$

$$\sum_{i \in I} x_{ij} \geq b_j, \forall\, j \in J$$

$$x_{ij} \geq 0 \quad \forall i \in I, j \in J$$

Problema das P-medianas



Problema das P-medianas

Índices / Conjuntos

I: Conjunto pontos de demanda, $\{1, 2, \ldots, m\}$

J: Conjunto pontos candidatos à instalação de UOP, $\{1, 2, \ldots, n\}$

Parâmetros

d_{ij}: Distância entre o ponto de demanda $i \in I$ e o ponto candidato $j \in J$

h_i: Número de acidentes no ponto de demanda $i \in I$

p: Número de UOPs a serem instaladas

Problema das P-medianas

Variáveis de decisão

$$y_j: \begin{cases} 1, & \text{se o ponto candidato } j \in J \text{ é escolhido como mediana} \\ 0, & \text{caso contrário} \end{cases}$$

$$x_{ij}: \begin{cases} 1, & \text{se o ponto de demanda } i \in I \text{ é alocado à mediana } j \in J \\ 0, & \text{caso contrário} \end{cases}$$

Problema das P-medianas

Formulação matemática

$$\min z = \sum_{i \in I} \sum_{j \in J} h_i d_{ij} x_{ij}$$

sujeito a

$$\sum_{j \in J} y_j = p$$

$$\sum_{j \in J} x_{ij} = 1, \forall\, i \in I$$

$$x_{ij} \leq y_j, \forall i \in I, j \in J$$

$$y_j \in \{0, 1\}, \forall\, j \in J$$

$$x_{ij} \in \{0, 1\}, \forall i \in I, j \in J$$

Referências

BELFIORE. P.; FÁVERO, L. P. Pesquisa Operacional para Cursos de Engenharia. Elsevier Brasil, 2013.

CARVALHO, C. W. V.; PITOMBEIRA NETO, A. R. Manual de uso da biblioteca Pyomo para Programação Matemática. 2020. Disponível em <http://www.opl.ufc.br/files/Manual_Pyomo_2020.pdf>. Acesso em: 12 de dez. 2021.

Optimization Team. Python Optimization from Beginner to Advance. 2021. Disponível em <https://www.udemy.com/course/optimization-in-python/>. Acesso em jul. 2021.

Pyomo. Pyomo Documentation 6.2. 2017. Disponível em <https://pyomo.readthedocs.io/en/stable/>. Acesso em: 10 de dez. 2021.

Sandia National Laboratories. Pyomo. 2011. Disponível em <https://www.osti.gov/biblio/1376827>. Acesso em: 10 de dez. 2021.

SILVA JÚNIOR, A. C. Python e Pyomo na modelagem de problemas de otimização. Disponível em <https://www.youtube.com/playlist?list=PLB0Bkje224EHp-IPd9i_FADBXXQcft5L6>. Acesso em: 12 de dez. 2021.

UFPR
UNIVERSIDADE FEDERAL DO PARANÁ

Download do material

https://github.com/juniorssz/ppgold-pyomo

UFPR
UNIVERSIDADE FEDERAL DO PARANÁ

Antonio C. da Silva Júnior
acsjunior.com

UNIVERSIDADE FEDERAL DO PARANÁ