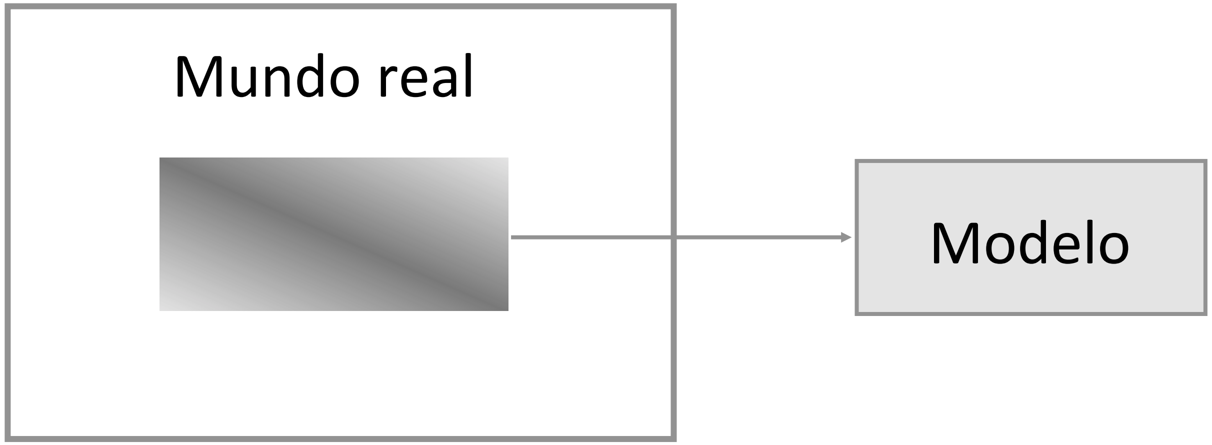
Planejamento e Produção
Modelos e Soluções

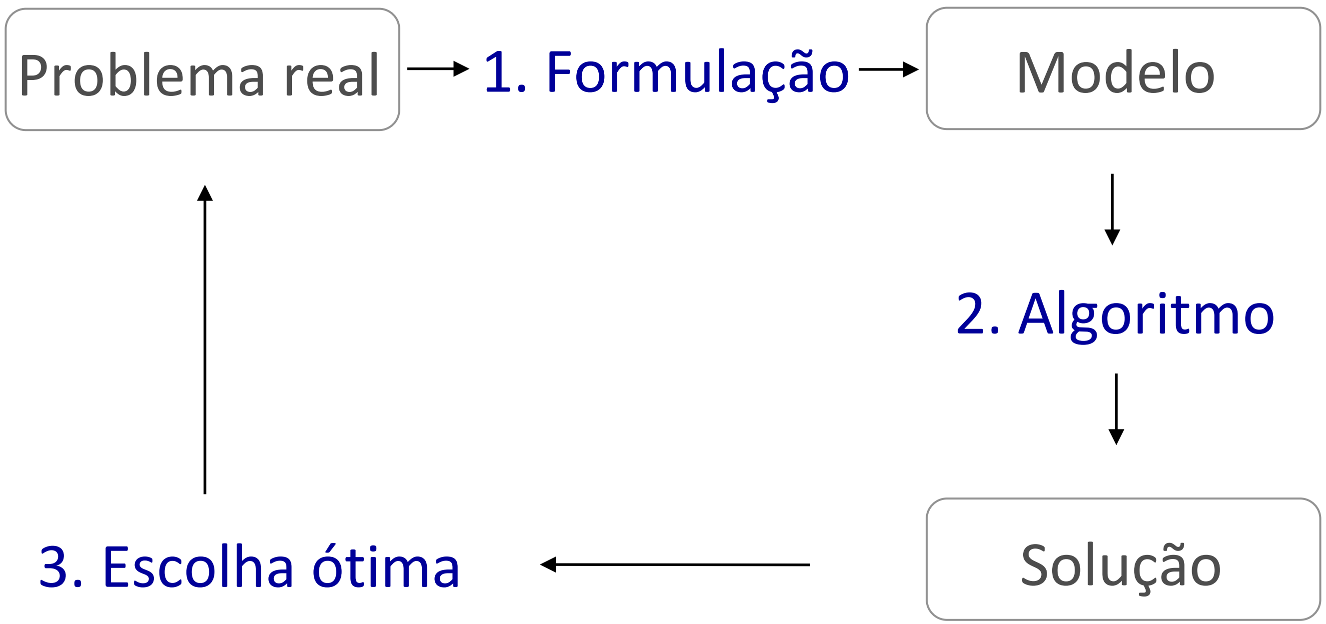
Prof. Fernando Gomide
Unicamp - FEEC - DCA

Criando modelos



Nível de abstração em modelagem

Modelagem e solução problemas de otimização



Exemplo: aquário de anfíbios

1. Formulação

Quantidade de alimento consumida por cada espécie

Alimentos	Sapo	Salamandra	Cobra cega	Disponibilidade
Minhocas	2	1	1	1500
Grilos	1	3	2	3000
Mosquitos	1	2	3	5000

Recursos: alimentos
Decisão: quantos anfíbios de cada espécie?

Minhocas:	1500
Grilos:	3000
Mosquitos:	5000

Espécies interagem compartilhando recursos comuns

Restrição: máximo de 1000 de cada espécie

Quantos anfíbios de cada espécie?

Ideia: maior número possível de anfíbios

Modelo

1. Variáveis de decisão (questão a resolver)

x_i: quantidade do i-ésimo anfíbio, $i = 1,2,3$

2. Função objetivo (o que se pretende)

Maximizar número total dos diferentes anfíbios

$f(\mathrm{x}) = x_1 + x_2 + x_3 \qquad \mathrm{x} = (x_1, x_2, x_3)$

3. Restrições principais e limites das variáveis (limitações recursos)

$$2x_1 + x_2 + x_3 \leq 1500$$

$$x_1 + 3x_2 + 2x_3 \leq 3000$$

$$x_1 + 2x_2 + 3x_3 \leq 5000$$

$$0 \leq x_1,\ x_2,\ x_3 \leq 1000$$

Modelo em uma linguagem de modelagem: OR-Tools

```
from ortools.linear_solver import pywraplp

def exemploAnfibios():
    """Exemplo amfíbios EA 044 FEEC Unicamp"””
#   Dados
    # anfíbio, minhocas, grilos, mosquitos
    A = [['sapo', 2, 1, 1],
         ['salamandra', 1, 3, 2],
         ['cobra', 1, 2, 3]]

    # disponbilidade
    b = [['minhocas', 1500],
         ['grilos', 3000],
         ['mosquitos', 5000]]

    # coeficientes
    c = [['sapo', 1],
         ['salamandra', 1],
         ['cobra', 1]]

    # Instância do Glop Solver para Anfíbios

    solver = pywraplp.Solver('Exemplo Anfíbios',

pywraplp.Solver.GLOP_LINEAR_PROGRAMMING)
```

```
# Cria variáveis de decisão e define seus limites, inferior e superior

    x = [solver.NumVar(0.0, 1000, A[i][0]) for i in range(0,len(A))]

    # x[0] número de sapos, x[1] número de salamandras, x[3] número de cobras cegas

    # Cria restrições principais

    constraints = [0]*len(b)

    for i in range(0, len(b)):
        constraints[i] = solver.Constraint(-solver.infinity(), b[i][1])
        for j in range(0, len(A)):
            constraints[i].SetCoefficient(x[j], A[i][j+1])

    # Cria função objetivo

    objective = solver.Objective()

    for i in range(0, len(A)):
        objective.SetCoefficient(x[i], c[i][1])

    # Define se é min ou max

    objective.SetMaximization()
```

```
 # Resolve modelo

    status = solver.Solve()

    if status == solver.OPTIMAL:

        print('Valor função objetivo', objective.Value())
        print('Solução ótima')
        for i in range(0,len(A)):
            print(x[i], '= {:.2f}'.format(x[i].solution_value()))
    else:
        print('Solução ótima não encontrada ou modelo ilimitado')

exemploAnfibios()
```

2. Algoritmo

Simplex

3. Escolha ótima

```
Valor função objetivo 1400.0
Solução ótima
sapo = 100.00
salamandra = 300.00
cobra = 1000.00
```

https://developers.google.com/optimization

Exemplo: planejamento mensal produção

1. Formulação

Produto	Máquina 1	Máquina 2	Moes	Mone	Preço
1	11	4	8	7	300
2	7	6	5	8	260
3	6	5	5	7	220
4	5	4	6	4	180

Recursos: máquinas, mão obra (horas)
Moes: mão de obra especializada
Mone: mão de obra não especializada
Preço: preço venda unitário ($)

Recursos

Máquina 1:	700 h/mês
Máquina 2:	500 h/mês
Moes:	600 h/mês a \$8/h
Mone:	650 h/mês a \$6/h

Quanto produzir e quanta mão de obra contratar no mês?

Ideia: maximizar o lucro

Recursos

Máquina 1:	700 h/mês
Máquina 2:	500 h/mês
Moes:	600 h/mês a $8/h
Mone:	650 h/mês a $6/h

Quanto produzir e quanta mão de obra contratar no mês?

Ideia: maximizar lucro

Modelo

1. Variáveis de decisão (o que decidir)

x_i: quantidade do *i*-ésimo produto, $i = 1,2,3,4$

y_s: número horas mo especializada

y_u: número de horas mo não especializada

2. Função objetivo (o que se pretende)

Lucro = receita – custos

$f(\mathrm{x}) = 300x_1 + 260x_2 + 220x_3 + 180\mathrm{x}_4 - (8y_s + 6y_u)$

$\mathrm{x} = (x_1, x_2, x_3, \mathrm{x}_4, y_s, y_u)$

3. Restrições (limitações recursos)

$$11x_1 + 7x_2 + 6x_3 + 5x_4 \leq 700$$

$$4x_1 + 6x_2 + 5x_3 + 4x_4 \leq 500$$

$$8x_1 + 5x_2 + 5x_3 + 6x_4 \leq y_s$$

$$7x_1 + 8x_2 + 7x_3 + 4x_4 \leq y_u$$

$$y_s \leq 600$$

$$y_u \leq 650$$

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

$$\max\ f(x) = 300x_1 + 260x_2 + 220x_3 + 180x_4 - (8y_s + 6y_u)$$

$$\begin{aligned}
\text{sa} \qquad & 11x_1 + 7x_2 + 6x_3 + 5x_4 \le 700 \\
& 4x_1 + 6x_2 + 5x_3 + 4x_4 \le 500 \\
& 8x_1 + 5x_2 + 5x_3 + 6x_4 \le y_s \\
& 7x_1 + 8x_2 + 7x_3 + 4x_4 \le y_u \\
& y_s \le 600 \\
& y_u \le 650 \\
& x_1,\ x_2,\ x_3,\ x_4, y_s, y_u \ge 0
\end{aligned}$$

2. Algoritmo

Simplex

3. Escolha ótima

$x_1 = 16 + 2/3$

$x_2 = 50$

$x_3 = 0$

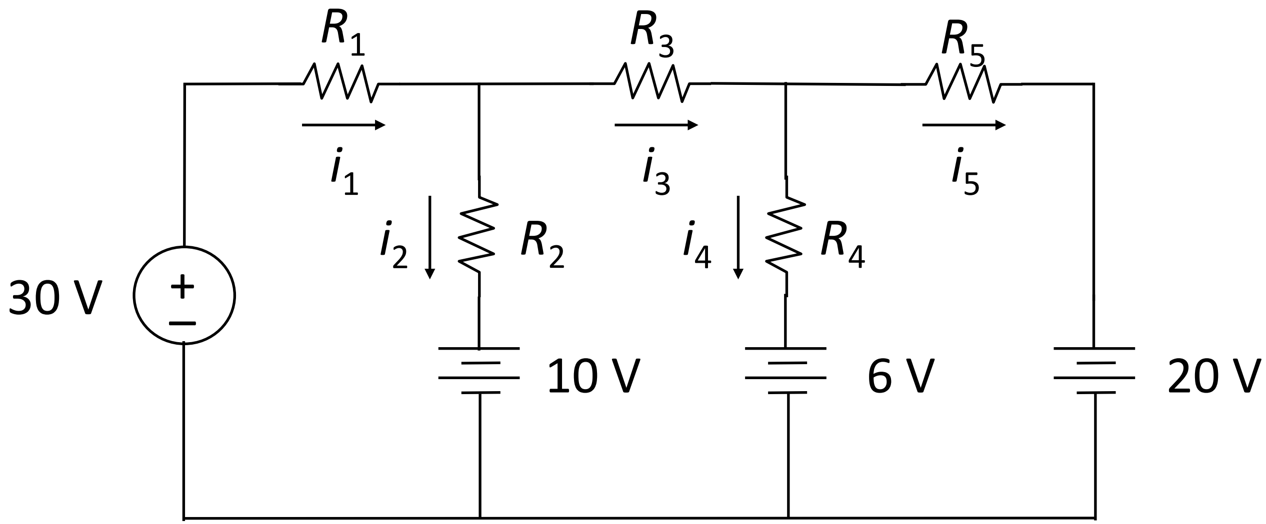
$x_4 = 33 + 1/3$

$y_s = 583$

$y_u = 650$

f(x) = \$ 15.433 + 1/3

Exemplo: carregador de baterias



$$\max \ f(x) = 10\, i_2 + 6\, i_4 + 20\, i_5$$

sa

$$i_1 = i_2 + i_3$$

$$i_3 = i_4 + i_5$$

$$i_1 \leq 4$$

$$i_2 \leq 3$$

$$i_3 \leq 3$$

$$i_4 \leq 2$$

$$i_5 \leq 2$$

$$i_1,\ i_2,\ i_3,\ \mathrm{i}_4,\ i_5 \geq 0$$

Recurso: fonte de tensão 30 V

Correntes máximas

$i_1 = 4$
$i_2 = 3$
$i_3 = 3$
$i_4 = 2$
$i_5 = 2$

Baterias não podem descarregar

Ideia: maximizar transferência de energia

Exemplo: P.Drosa

1. Formulação

$$\min \; 3.50 \, [\, (r - 55) + q/2 \,] + 2000 \, / \, (q/55)$$

$$\text{sa} \quad q \geq 100$$

$$r \geq 55$$

Recurso: diamante
Decisão: quando e quanto repor?

Modelo em uma linguagem de modelagem: Scipy

```
import numpy as np
import scipy.optimize as op

def custo(x):
    return 3.5*((x[0] - 55) + x[1]/2) + 2000*55/x[1]

limites = ((55,np.inf),(100,np.inf))

x0 = np.array([50,0])

solucao = op.minimize(custo, x0, bounds = limites)

print(solucao.x)
```

https://www.scipy.org

Observação

Este material refere-se às notas de aula do curso EA 044 Planejamento e Análise de Sistemas de Produção da Faculdade de Engenharia Elétrica e de Computação da Unicamp. Não substitui o livro texto, as referências recomendadas e nem as aulas expositivas. Este material não pode ser reproduzido sem autorização prévia dos autores. Quando autorizado, seu uso é exclusivo para atividades de ensino e pesquisa em instituições sem fins lucrativos.