Tutorial AMPL

Daniela Cristina Lubke
danielalubke@cos.ufrj.br

Universidade Federal do Rio de Janeiro, COPPE, PESC

12 de Maio de 2016

A Mathematical Programming Language - AMPL

- É uma linguagem de modelagem algébrica para descrever e resolver problemas de otimização;
- O AMPL foi desenvolvido por Robert Fourer , David Gay, e Brian Kernighan;
- Exemplos de solvers suportados pelo AMPL: Couenne, CPLEX, FortMP, Gurobi, Xpress, MINOS, IPOPT, MOSEK, SNOPT, KNITRO e LGO.

COPPE UFRJ

Problema da ração

"Uma empresa de comida canina produz dois tipos de rações: Tobi e Rex. Para a manufatura das rações são utilizados cereais e carne. Sabe-se que:

- a ração Tobi utiliza 5 kg de cereais e 1 kg de carne, e a ração Rex utiliza 4 kg de carne e 2 kg de cereais;
- o pacote de ração Tobi custa \$ 20 e o pacote de ração Rex custa \$ 30;
- o kg de carne custa \$ 4 e o kg de cereais custa \$ 1;
- estão disponíveis por mês 10 000 kg de carne e 30 000 kg de cereais.

Deseja-se saber qual a quantidade de cada ração deve ser produzida de modo a maximizar o lucro."

Formulação do Problema da ração

$$
\begin{array}{lll}
\text{maximizar} & 11x_1 + 12x_2 & \\
\text{sujeito a:} & 5x_1 + 2x_2 \leq 30000 & \text{(restrição de cereais)} \\
 & 1x_1 + 4x_2 \leq 10000 & \text{(restrição de carne)} \\
 & x_1 \geq 0, \quad x_2 \geq 0 &
\end{array}
$$

Código Problema da ração - racao.mod

```
# declaração das variáveis
var x1 >= 0;
var x2 >= 0;

# Função objetivo
maximize FO: 11*x1 + 12*x2;

# Restrições
subject to restricaocarne: 1*x1 + 4*x2 <= 10000;
subject to restricaocereais: 5*x1 + 2*x2 <= 30000;
```

COPPE UFRJ

Executando o modelo racao.mod

```
ampl: model racao.mod;
ampl:
```

Executando o modelo racao.mod

```
ampl: model racao.mod;
ampl: option solver cplex;
ampl:
```

Executando o modelo racao.mod

```
ampl: model racao.mod;
ampl: option solver cplex;
ampl: solve;
CPLEX 12.6.3.0: optimal solution; objective 74444.44444
0 dual simplex iterations (0 in phase I)
ampl:
```

Executando o modelo racao.mod

```
ampl: model racao.mod;
ampl: option solver cplex;
ampl: solve;
CPLEX 12.6.3.0: optimal solution; objective 74444.44444
0 dual simplex iterations (0 in phase I)
ampl: display FO,x1,x2;
FO = 74444.4
x1 = 5555.56
x2 = 1111.11
ampl:
```

Executando o modelo racao.mod

```
ampl: model racao.mod;
ampl: option solver cplex;
ampl: solve;
CPLEX 12.6.3.0: optimal solution; objective 74444.44444
0 dual simplex iterations (0 in phase I)
ampl: display FO,x1,x2;
FO = 74444.4
x1 = 5555.56
x2 = 1111.11
ampl: quit;
```

COPPE UFRJ

Problema da Marcenaria

"Uma marcenaria deseja estabelecer uma programação diária de produção. Atualmente, a oficina faz apenas dois produtos: mesa e armário, ambos de um só modelo. Para efeito de simplificação, vamos considerar que a marcenaria tem limitações em somente dois recursos: madeira e mão-de-obra, cujas disponibilidades diárias são mostradas na tabela a seguir.

Recurso	Disponibilidade
Madeira	12 m^2
Mão-de-obra	8 H.h

Problema da Marcenaria

O processo de produção é tal que, para fazer uma mesa a fábrica gasta 2 m^2 de madeira e 2 H.h de mão-de-obra. Para fazer um armário, a fábrica gasta 3 m^2 de madeira e 1 H.h de mão de obra. Além disso, o fabricante sabe que cada mesa dá uma margem de contribuição para o lucro de \$ 4 e cada armário de \$ 1. O problema é encontrar o programa de produção que maximiza a margem de contribuição total para o lucro."

Formulação do Problema da Marcenaria

$$
\begin{array}{lll}
\text{maximizar} & 4x_1 + x_2 & \\
\text{sujeito a:} & 2x_1 + 3x_2 \leq 12 & \text{(restrição de madeira)} \\
 & 2x_1 + x_2 \leq 8 & \text{(restrição de mão-de-obra)} \\
 & x_1 \geq 0, \quad x_2 \geq 0 &
\end{array}
$$

COPPE UFRJ

Código Problema da Marcenaria - marcenaria.mod

```
# declaração das variáveis
var x1 >= 0;
var x2 >= 0;

# Função Objetivo
maximize FO: 4*x1 + 1*x2;

# Restrições
subject to restricaomadeira: 2*x1 + 3*x2 <= 12;
subject to restricaomaodeobra: 2*x1 + 1*x2 <= 8;
```

COPPE UFRJ

ampl:

```
ampl: model marcenaria.mod
ampl:
```

```
ampl: model marcenaria.mod
ampl: option solver cplex;
ampl:
```

```
ampl: model marcenaria.mod
ampl: option solver cplex;
ampl: solve;
CPLEX 12.6.3.0: optimal solution; objective 16
2 dual simplex iterations (1 in phase I)
ampl:
```

```
ampl: model marcenaria.mod
ampl: option solver cplex;
ampl: solve;
CPLEX 12.6.3.0: optimal solution; objective 16
2 dual simplex iterations (1 in phase I)
ampl: display FO, x1,x2;
FO = 16
x1 = 4
x2 = 0
```

Sufixo das restrições

- .lb limite inferior;
- .body valor da restrição;
- .ub limite superior;
- .slack folga.

Sufixo das restrições

$$2x_1 + x_2 \leq 8 \quad \text{(restrição de mão-de-obra)}$$

- $x_1 = 4$
- $x_2 = 0$

Sufixo das restrições

$$2x_1 + x_2 \leq 8 \quad \text{(restrição de mão-de-obra)}$$

- $x_1 = 4$
- $x_2 = 0$

```
ampl: display restricaomaodeobra.ub, restricaomaodeobra.lb,
restricaomaodeobra.slack, restricaomaodeobra.body;
```

Sufixo das restrições

$$2x_1 + x_2 \leq 8 \quad \text{(restrição de mão-de-obra)}$$

- $x_1 = 4$
- $x_2 = 0$

ampl: display restricaomaodeobra.ub, restricaomaodeobra.lb, restricaomaodeobra.slack, restricaomaodeobra.body;
restricaomaodeobra.ub = 8
restricaomaodeobra.lb = -Infinity
restricaomaodeobra.slack = 0
restricaomaodeobra.body = 8

Sufixo das restrições

$$2x_1 + 3x_2 \leq 12 \qquad \text{(restrição de madeira)}$$

- $x_1 = 4$
- $x_2 = 0$

COPPE UFRJ

Sufixo das restrições

$$2x_1 + 3x_2 \leq 12 \qquad \text{(restrição de madeira)}$$

- $x_1 = 4$
- $x_2 = 0$

```
ampl: display restricaomadeira.ub, restricaomadeira.lb,
restricaomadeira.slack, restricaomadeira.body;
```

Sufixo das restrições

$$2x_1 + 3x_2 \leq 12 \qquad \text{(restrição de madeira)}$$

- $x_1 = 4$
- $x_2 = 0$

ampl: display restricaomadeira.ub, restricaomadeira.lb,
restricaomadeira.slack, restricaomadeira.body;
restricaomadeira.ub = 12
restricaomadeira.lb = -Infinity
restricaomadeira.slack = 4
restricaomadeira.body = 8

Problema do Transporte

Uma empresa tem 3 fábricas que produzem um determinado produto. A capacidade de produção mensal das 3 fábricas é de 6, 1 e 10 unidades respectivamente. A empresa tem 4 armazéns de vendas que vendem mensalmente 7, 5, 3 e 2 unidades do produto respectivamente. O custo de transportar uma unidade de cada fábrica para cada armazém está dado na tabela abaixo:

	Fábrica 1	Fábrica 2	Fábrica 3
Armazém 1	\$2	\$1	\$5
Armazém 2	\$3	\$0	\$8
Armazém 3	\$11	\$6	\$15
Armazém 4	\$7	\$1	\$9

COPPE UFRJ

Problema do Transporte

O objetivo da Empresa é atender as necessidades dos armazéns com a produção das fábricas de forma a minimizar o custo total.

Formulação do Problema

Sejam:

- c_{ij} - Custo de distribuição entre fábrica i e armazém j;
- x_{ij} - Total a ser distribuído da fábrica i até o armazém j;
- F_i - Total produzido pela fábrica i;
- D_j - Demanda total do armazém j.

COPPE UFRJ

Formulação do Problema

$$\text{minimizar} \quad \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}$$

Formulação do Problema

$$\text{minimizar} \quad \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}$$

$$\text{sujeito a:} \quad \sum_{j=1}^{n} x_{ij} = F_i$$

Formulação do Problema

$$\text{minimizar} \quad \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}$$

$$\text{sujeito a:} \quad \sum_{j=1}^{n} x_{ij} = F_i$$

(Soma do que vai para cada armazém
não pode ultrapassar a produção)

COPPE UFRJ

Formulação do Problema

minimizar $\sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}$

sujeito a: $\sum_{j=1}^{n} x_{ij} = F_i$ (Soma do que vai para cada armazém não pode ultrapassar a produção)

$\sum_{i=1}^{m} x_{ij} = D_j$

Formulação do Problema

$$\text{minimizar} \quad \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}$$

$$\text{sujeito a:} \quad \sum_{j=1}^{n} x_{ij} = F_i$$

(Soma do que vai para cada armazém

não pode ultrapassar a produção)

$$\sum_{i=1}^{m} x_{ij} = D_j$$

(quantidade enviada da fábrica para o armazém

são iguais as demandas do armazém)

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

COPPE UFRJ

Problema do Transporte

Formulação do problema:

minimizar

$$
\begin{aligned}
& 2\ \text{FAB1:ARM1} + 1\ \text{FAB2:ARM1} + 5\ \text{FAB3:ARM1} + \\
& 3\text{FAB1:ARM2} + 0\text{FAB2:ARM2} + 8\text{FAB3:ARM2} + \\
& 11\text{FAB1:ARM3} + 6\text{FAB2:ARM3} + 15\text{FAB3:ARM3} + \\
& 7\text{FAB1:ARM4} + 1\text{FAB2:ARM4} + 9\text{FAB3:ARM4}
\end{aligned}
$$

sujeito a:

$$
\begin{aligned}
& \text{FAB1:ARM1} + \text{FAB1:ARM2} + \text{FAB1:ARM3} + \text{FAB1:ARM4} = 6 \\
& \text{FAB2:ARM1} + \text{FAB2:ARM2} + \text{FAB2:ARM3} + \text{FAB2:ARM4} = 1 \\
& \text{FAB3:ARM1} + \text{FAB3:ARM2} + \text{FAB3:ARM3} + \text{FAB3:ARM4} = 10 \\
\\
& \text{FAB1:ARM1} + \text{FAB2:ARM1} + \text{FAB3:ARM1} = 7 \\
& \text{FAB1:ARM2} + \text{FAB2:ARM2} + \text{FAB3:ARM2} = 5 \\
& \text{FAB1:ARM3} + \text{FAB2:ARM3} + \text{FAB3:ARM3} = 3 \\
& \text{FAB1:ARM4} + \text{FAB2:ARM4} + \text{FAB3:ARM4} = 2
\end{aligned}
$$

Código: transporte.mod

```
# Conjunto de Objetos
set ORIG; # ORIG - Fábricas
set DEST; # DEST - Armazém

param oferta {ORIG} >= 0; # quantidade disponível na origem
param demanda {DEST} >= 0; # quantidade requerida no
destino

# Verifica se o problema está balanceado
check: sum {i in ORIG} oferta[i] = sum {j in DEST} demanda[j];

param custo {ORIG,DEST} >= 0; # custo de transporte por
unidade
var Trans {ORIG,DEST} >= 0; # unidades que serão enviadas
da fábrica para o armazém
```

Código: transporte.mod - continuação...

```
# Função objetivo
minimize Custo_Total: sum {i in ORIG, j in DEST} custo[i,j] *
Trans[i,j];

# Restrições
subject to Oferta {i in ORIG}:
sum {j in DEST} Trans[i,j] = oferta[i];

subject to Demanda {j in DEST}:
sum {i in ORIG} Trans[i,j] = demanda[j];
```

Dados de entrada: transporte.dat

```
param: ORIG: oferta := # define o conjunto ORIG e o param oferta
FAB1    6
FAB2    1
FAB3    10;
param: DEST: demanda := # define o conjunto DEST e o param demanda
ARM1    7
ARM2    5
ARM3    3
ARM4    2;
param: custo:
        ARM1    ARM2    ARM3    ARM4 :=
FAB1    2       3       11      7
FAB2    1       0       6       1
FAB3    5       8       15      9;
```

Problema do Transporte

ampl:

Problema do Transporte

```
ampl: model transporte.mod;
ampl:
```

Problema do Transporte

```
ampl: model transporte.mod;
ampl: data transporte.dat;
ampl:
```

Problema do Transporte

```
ampl: model transporte.mod;
ampl: data transporte.dat;
ampl: option solver cplex;
ampl:
```

Problema do Transporte

```
ampl: model transporte.mod;
ampl: data transporte.dat;
ampl: option solver cplex;
ampl: solve;
CPLEX 12.6.3.0: optimal solution; objective 100
9 dual simplex iterations (0 in phase I)
```

COPPE UFRJ

Problema do Transporte

ampl:

Problema do Transporte

```
ampl: display Trans;
Trans :=
FAB1 ARM1 0
FAB1 ARM2 5
FAB1 ARM3 1
FAB1 ARM4 0
FAB2 ARM1 0
FAB2 ARM2 0
FAB2 ARM3 1
FAB2 ARM4 0
FAB3 ARM1 7
FAB3 ARM2 0
FAB3 ARM3 1
FAB3 ARM4 2;
ampl:
```

Problema do Transporte

```
ampl: display Trans;
Trans :=
FAB1 ARM1 0
FAB1 ARM2 5
FAB1 ARM3 1
FAB1 ARM4 0
FAB2 ARM1 0
FAB2 ARM2 0
FAB2 ARM3 1
FAB2 ARM4 0
FAB3 ARM1 7
FAB3 ARM2 0
FAB3 ARM3 1
FAB3 ARM4 2;
ampl: quit;
```

Fim

Obrigada!

Slide e exemplos disponíveis em:

www.cos.ufrj.br/∼danielalubke/

Tutorial AMPL

Daniela Cristina Lubke
danielalubke@cos.ufrj.br

Universidade Federal do Rio de Janeiro, COPPE, PESC

12 de Maio de 2016