

## Laboratório 1

### Preparação do trabalho

- Windows: Baixar e instalar o GNU Linear Programming Kit (GLPK) em <http://gnuwin32.sourceforge.net/packages/glpk.htm>.
- Eventualmente modificar o ambiente (`set PATH=%PATH%;<dir>/bin`).
- Linux: `sudo apt-get install glpk`
- Teste: Executar `glpsol --help`.
- Documentação MathProg: <http://www.inf.ufrgs.br/~mrpritt/oc/lang.pdf>

### Exemplo

O exemplo do importador de Whisky vista em aula mostra a especificação de um problema nos formatos CPLEX lp e GNU mathprog. O problema foi

Um importador de Whisky tem as seguintes restrições de importação

- no máximo 2000 garrafas de *Johnny Ballantine* por 70 R\$ cada uma,
- no máximo 2500 garrafas de *Old Gargantua* por 50 R\$ cada uma,
- no máximo 1200 garrafas de *Misty Deluxe* por 40 R\$ cada uma.

Dos Whiskies importados ele produz três misturas *A*, *B*, *C*, que ele vende por 68 R\$, 57 R\$ e 45 R\$, respectivamente. As misturas são

- A: no mínimo 60% Johnny Ballantine, no máximo 20% Misty Deluxe,
- B: no mínimo 15% Johnny Ballantine, no máximo 60% Misty Deluxe,
- C: no máximo 50% Misty Deluxe.

Quais seriam as misturas ótimas, e quantas garrafas de cada mistura devem ser produzidas para maximizar o lucro?

A formulação do problema é

$$\begin{aligned} \max \quad & 68x_A + 57x_B + 45x_C - (70x_J + 50x_O + 40x_M) \\ \text{s.a} \quad & x_J \leq 2000 \\ & x_O \leq 2500 \\ & x_M \leq 1200 \\ & x_{J,A} \geq 0.6x_A \\ & x_{M,A} \leq 0.2x_A \\ & x_{J,B} \geq 0.15x_B \\ & x_{M,B} \leq 0.6x_B \\ & x_{M,C} \leq 0.5x_C \\ & x_m = x_{m,A} + x_{m,B} + x_{m,C} & m \in \{J, O, M\} \\ & x_m = x_{J,m} + x_{O,m} + x_{M,m} & m \in \{A, B, C\} \\ & x_{m,n} \geq 0 & m \in \{J, O, M\}, n \in \{A, B, C\} \end{aligned}$$

### Especificação com CPLEX lp

<http://www.inf.ufrgs.br/~mrpritt/oc/whisky.lp>

Maximize  
lucro : 68xa+57xb+45xc-70xj-50xo-40xm  
Subject to  
limiteJohnny : xj<=2000  
limiteOldGarg : xo<=2500  
limiteMisty : xm<=1200

```
percJohnnyA:      xja - 0.6xa >= 0
percMistzA:       xma - 0.2xa <= 0
percJohnnyB:      xjb - 0.15xb >= 0
percMistyB:       xmb - 0.6xb <= 0
percMistyC:       xmc - 0.5xc <= 0

totalJohnny:      xj - xja - xjb - xjc = 0
totalOldGarg:     xo - xoa - xob - xoc = 0
totalMisty:       xm - xma - xmb - xmc = 0
totalA:           xa - xja - xoa - xma = 0
totalB:           xb - xjb - xob - xmb = 0
totalC:           xc - xjc - xoc - xmc = 0
```

Bounds

```
0 <= xja
0 <= xjb
0 <= xjc
0 <= xma
0 <= xmb
0 <= xmc
0 <= xoa
0 <= xob
0 <= xoc
```

End

## Especificação com GNU mathprog

<http://www.inf.ufrgs.br/~mrpritt/oc/whisky.mod>

```
# conjuntos basicos de indices
set misturas;
set marcas ;

# custo de cada garrafa por marca
param custoMarca { marcas } >= 0;
# preco de cada mistura
param precoMistura { misturas } >= 0;
# limite de importacao de cada marca
param limiteMarca { marcas } >= 0;
# percentagens minimas das marcas nas misturas
param percMin { marcas, misturas } >= 0, <= 1, default 0;
# percentages maximas das marcas nas misturas
param percMax { marcas, misturas } >= 0, <= 1, default 1;

# numero de garrafas
var numMistura { misturas } >= 0;
var numMarca { marcas } >= 0;
var numGarrafas { marcas, misturas } >= 0;

# definicao do modelo

maximize lucro:
    sum { j in misturas } precoMistura[j]*numMistura[j] -
    sum { j in marcas } custoMarca[j]*numMarca[j];

subject to limitesImportacao { m in marcas }:
    numMarca[m] <= limiteMarca[m];
subject to maxPercentagens { m in marcas, i in misturas }:
    numGarrafas[m,i] <= percMax[m,i]*numMistura[i];
```

```
subject to minPercentagens { m in marcas, i in misturas }:  
    numGarrafas[m,i] >= percMin[m,i]*numMistura[i];  
subject to totalMarca { m in marcas }:  
    numMarca[m] = sum { j in misturas } numGarrafas[m,j];  
subject to totalMistura { m in misturas }:  
    numMistura[m] = sum { j in marcas } numGarrafas[j,m];  
  
# definicao dos dados  
data;  
  
# formato de tabela: a 1a coluna define o conjunto marcas  
# e os seguintes colunas os valores de parametros que  
# dependem de marcas  
param : marcas : custoMarca limiteMarca :=  
    'J'      70      2000  
    'O'      50      2500  
    'M'      40      1200    ;  
  
# tabela para misturas  
param : misturas : precoMistura :=  
    'A'      68  
    'B'      57  
    'C'      45    ;  
  
# definicao de percMin em formato de matriz:  
# a 1a linha contem os 1os indices da matriz  
# a 1a coluna contem os 2os indices da matriz  
# o valor "." significa usar o default  
param percMin : 'A' 'B' 'C' :=  
    'J'      0.6  0.15  .  
    'O'      .   .     .  
    'M'      .   .     .    ;  
  
# matrix das percentagens maximos  
param percMax : 'A' 'B' 'C' :=  
    'J'      .   .     .  
    'O'      .   .     .  
    'M'      0.20 0.60 0.50    ;  
  
end;
```

### Questão 0.1 (Empresa de aço)

Formule e resolva o problema da empresa de aço.

- Coloque o problema em formato CPLEX lp (p.ex. arquivo `aco.lp`)
- Resolva o problema com

```
glpsol --cpxlp aco.lp -o aco.sol
```

### Questão 0.2 (Produção de TVs)

Formule e resolva o problema da produção de TVs.

- Coloque o problema em formato CPLEX lp (p.ex. arquivo `tv.lp`)
- Resolva o problema com

```
glpsol --cpxlp tv.lp -o tv.sol
```

### Questão 0.3 (Empresa aérea)

Resolva o exercício (3.4) da empresa aérea:

Uma pequena empresa aérea oferece um voo de Pelotas, com escala em Porto Alegre para Torres. Logo os clientes podem voar em três trechos diferentes Pelotas–Porto Alegre, Pelotas–Torres e Porto Alegre–Torres. A linha também oferece três tipos de bilhetes:

- Tipo A: bilhete regular.
- Tipo B: sem cancelamento.
- Tipo C: sem cancelamento, pagamento deve ser efetuado três semanas antes de viajar.

Os preços (em R\$) dos bilhetes são os seguintes

|   | Pelotas–Porto Alegre | Porto Alegre–Torres | Pelotas–Torres |
|---|----------------------|---------------------|----------------|
| A | 600                  | 320                 | 720            |
| B | 440                  | 260                 | 560            |
| C | 200                  | 160                 | 280            |

Baseado em experiência com esses voos, o marketing tem a seguinte predição de passageiros:

|   | Pelotas–Porto Alegre | Porto Alegre–Torres | Pelotas–Torres |
|---|----------------------|---------------------|----------------|
| A | 4                    | 8                   | 3              |
| B | 8                    | 13                  | 10             |
| C | 22                   | 20                  | 18             |

O objetivo da empresa é maximizar o lucro determinando o número ótimo de bilhetes de cada tipo para vender, respeitando um limite de 30 passageiros em cada voo, e o limite dos passageiros previstos em cada categoria. Observe que no primeiro trecho, por exemplo, os passageiros viajando de Pelotas para Porto Alegre e os passageiros viajando de Pelotas para Torres compartilham a avião.

- Formula o programa linear correspondente e resolve com GLPK.
- Qual o número ótimo de bilhetes para todos tipos de voos?

#### Questão 0.4 (McDonalds)

Resolve o problema da dieta:

Suponha que temos uma tabela de nutrientes de diferentes tipos de alimento. Sabendo o valor diário de referência (VDR) de cada nutriente e o preço de cada unidade de alimento, qual a dieta ótima, i.e. que contém ao menos o valor diário de referência, mas custo de menos?

Com  $m$  nutrientes e  $n$  alimentos, seja  $a_{ij}$  a quantidade de nutriente  $i$  na alimento  $j$ ,  $r_i$  valor diário de referência do nutriente  $i$  e  $c_j$  o preço de alimento  $j$ . Queremos saber as quantidades  $x_j$  de cada alimento que

$$\text{minimiza} \quad c_1x_1 + \cdots + c_nx_n$$

sujeito a

$$a_{11}x_1 + \cdots + a_{1n}x_n \geq r_1$$

...

$$a_{m1}x_1 + \cdots + a_{mn}x_n \geq r_m$$

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

Os dados do McDonalds são disponíveis em

<http://www.inf.ufrgs.br/~mrpritt/oc/McDonalds.dat>

no formato AMPL. Os dados usam os nomes `cost[i]` para  $c_i$ , `amt[i,j]` para  $a_{ij}$  e `n_min[i]` para  $r_i$ . Os valores dos nutrientes são em % do VDR, tal que as restrições são simplesmente  $r_i = 100\%$ . Além disso, o arquivo contém os dados `f_min[i]`, `f_max[i]` e `n_max[i]` para o valor mínimo e máximo de alimento da cada tipo, e a percentagem máxima de cada nutriente. Essas colunas contém um . que significa, que o valor “default” é usado. Esse valor tem que ser especificado no modelo.

- Implemente o problema da dieta usando AMPL (arquivo `McDonalds.mod`).
- Resolve o problema com `glpsol -m McDonalds.mod -d McDonalds.dat -o McDonalds.sol` usando GLPK.
- Qual a dieta ótima com as restrições acima? Qual o custo dela?

Para resolver os seguintes itens, modifique o sistema adequadamente e resolve de novo.

- O que custa uma dieta ótima entre 2000 kcal (o VDR) e 2100 kcal?
- Qual a dieta ótima vegetariana?
- Qual a dieta ótima em que cada alimento é consumido uma única vez?