

Otimização: Introdução para AMPL

Gurvan Huiban
ghuiban@cin.ufpe.br

09 de abril de 2014

Porque Linguagens de modelagem?

Motivação

- Tamanho de problemas de otimização
- Uso de bibliotecas de resolução (Cplex, Gurobi, SCIP, ...)

⇒ Intermediário entre o modelo e o *solver*

Tarefas

- Modelagem
- Resolução
 - Algoritmos
 - Interações com *solver*
 - Análise de resultados

AMPL

A Mathematical Programming Language

- Linguagem de modelagem
- Linguagem de programação
- Ferramenta comercial

Outras ferramentas

- GMPL, ZIMPL, Concert technology...

Neste curso

- GMPL (Subconjunto do AMPL)
- Modelagem

Problema de otimização

Elementos

- Variáveis
- Objetivo
- Restrições

Refinaria: Modelagem

```
var xs >=0 <=9;
var xv >=0 <=6;
minimize custo:
    20*xs + 15*xv;
subject to gas:
    0.3*xs + 0.4*xv >= 2;
subject to que:
    0.4*xs + 0.2*xv >= 1.5;
subject to ole:
    0.2*xs + 0.3*xv >= 0.5;
end;
```

Refinaria: Resolução

Resolução usando a ferramenta GLPK

```
glpsol -m refinaria.mod -o refinaria.sol
```

Refinaria: Arquivo LP

Arquivo que tem o modelo *in extenso*

```
\* Problem: refinaria *\
Minimize
  custo: + 20 xs + 15 xv
Subject To
  gas: + 0.3 xs + 0.4 xv >= 2
  que: + 0.4 xs + 0.2 xv >= 1.5
  ole: + 0.2 xs + 0.3 xv >= 0.5
Bounds
  0 <= xs <= 9
  0 <= xv <= 6
End
```

⇒ Ganho?

Refinaria: Abstração

Dados

- n : tipos de produtos a serem produzidos
- m : tipos de petróleos
- d_n : demanda do produto n
- $p_{m,n}$: proporção do produto n gerado com petróleo m
- d_m : disponibilidade do petróleo m
- c_m : custo do petróleo m

Modelo abstrato

- Identifique os conjuntos, parâmetros, variáveis
- Escreva o modelo de forma abstrata

GMPL: Tutorial (1)

Sintaxe

- **Conjunto:**

```
set <nome>;
```

- **Parâmetros:**

```
param <nome>{<Conjunto,Conjunto,...>;
```

- **Variáveis:**

```
var <nome>{Conjunto} restrições;
```

- **Objetivo:**

```
minimize/maximize <nome>:  
expressão;
```

- **Restrições:**

```
subject to <nome>{<índice> in <Conjunto>}:  
expressão;
```

GMPL: Tutorial (2)

Sintaxe

- Acesso a um elemento de índice i :

`<nome> [i]`

- Expressões: Símbolos matemáticos

- Soma:

`sum{<índice> in <Conjunto>} elementos[índice];`

GMPL: Tutorial (3)

Sintaxe para dados

- Conjuntos:

```
set <nome> := <valores>;
```

- Parâmetros:

- Indexado por um conjunto

```
param <nome> :=  
<elemento do conjunto> valor  
...  
<elemento do conjunto> valor;
```

- Indexado com dois conjuntos

```
param <nome> : <elementos do conjunto 2>  
:=  
<elemento do conjunto1> valores  
...  
<elemento do conjunto1> valores;
```