Softwares de Otimização

Problemas de otimização

Componentes

- Função objetivo
- Restrições
- Informações de variáveis
  - limites
  - variáveis inteiras genéricas
  - variáveis binárias

Problema exemplo

```
Minimize
        obj: 3 x1 + 2.5 x2

subject to
        row1: 6 x1 + 4 x2 >= 32
        row2: 5 x1 + 6 x2 >= 3

Bounds
        0 <= x1 <= 40
        2 <= x2 <= 3
End
```

Formulações

Programação Linear

$$(LP)\ \max\{cx : A\,x \leq b,\ x \geq 0\},$$

onde $A \in R^{m \times n}$, $c \in R^{n}$, $b \in R^{n}$.

- Se algumas (não todas) variáveis de decisão devem assumir valores inteiros, temos o problema de **Programação Linear Inteira Mista**

- Se todas as variáveis de decisão devem assumir valores inteiros, temos o problema de **Programação Linear Inteira**

$$(IP)\ \max\{cx : Ax \leq b,\ x \geq 0,\ x \text{ inteiro}\}$$

- Se todas as varáveis de decisão devem assumir valores 0 ou 1, temos o problema de **Programação Inteira Binário**

$$(BIP)\ \max\{cx : Ax \leq b,\ x \in \{0,1\}^{n}\}.$$

O que é um LP Solver ?

- *Solver* é um termo genérico para definir uma parte de um software matemático (programa de computador ou biblioteca) que ‘resolve’ um problema matemático.

- Baseado na descrição do problema ele apresenta de alguma forma uma solução.

- A ênfase está em definir um programa “caixa-preta” que poderá ser utilizado para facilmente resolver uma gama de problemas similares.

- Um *LP solver* é, dessa forma, um pacote de software capaz de resolver problemas de Programação Linear.

Pacotes de otimização

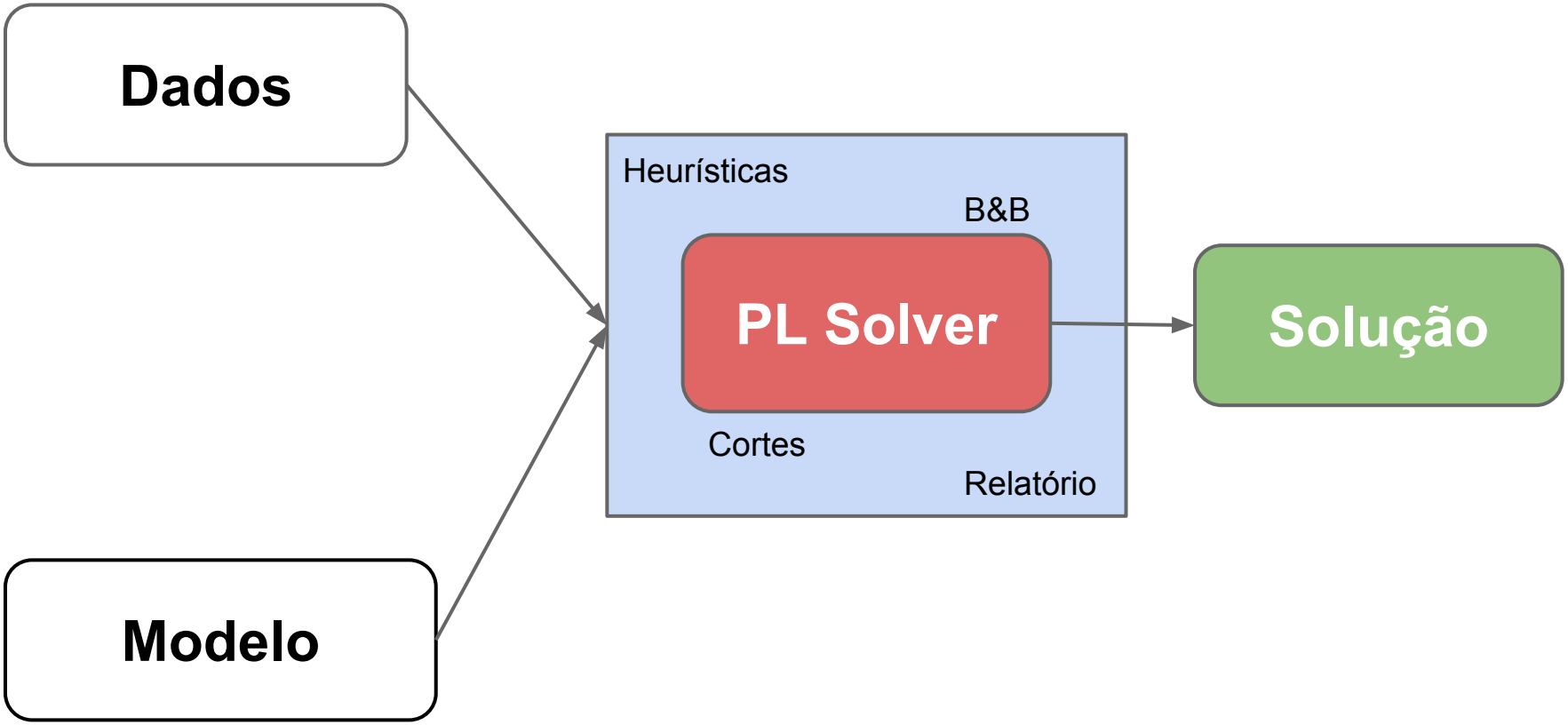
- Pacotes de otimização são ferramentas para modelagem e resolução de problema de Programação Linear, Programação Linear Inteira, etc.

- Estes pacotes são constituídos de: linguagens de modelagem, PL solver, heurísticas é rotinas para diversos métodos baseados em programação matemática.

- Possibilitam a geração de relatórios para avaliação dos resultados.

- Fornecem a programadores interfaces para diversas linguagens de programação (C, C++, Java, Python, etc.).

Pacotes de otimização



Exemplos de Pacotes:

Existem diversos pacotes comerciais e gratuitos disponíveis. Em geral, eles diferem nos métodos implementados e tipos de problemas que são capazes de resolver.

- **ILOG CPLEX (IBM):** Resolve problemas de programação inteira, programação linear de larga escala, programação quadrática, problemas com restrições quadráticas convexas.
  - Suporte às linguagens: C, C++, C#, Java e Python;
  - Sistemas de modelagem: AIMMS, AMPL AIMMS, AMPL, GAMS, MPL, OpenOpt, OptimJ e TOMLAB.
  - Licença: Software proprietário;
  - **Métodos:** Simplex, Pontos Interiores, Barrier, Branch &Bound, etc.

Exemplos de Pacotes:

- **Gorubi**: Resolve problemas de programação inteira, programação linear de larga escala, programação inteira mista, programação quadrática, problemas com restrições quadráticas convexas.
  - Suporte às linguagens: C, C++, Java, .NET e Python;
  - Sistemas de modelagem: AIMMS, AMPL, GAMS, e MPL.
  - Licença: Software proprietário;

- **GNU Linear Programming Kit (GLPK):** Resolve problemas de programação linear e programação inteira mista.
  - Suporte às linguagens: C, C++, Java
  - Linguagem de modelagem: GAMS e GMPL
  - Licença: Software Livre.

- Outros: **XPRESS, Mosek, CERES**, etc.

Solvers: benchmark

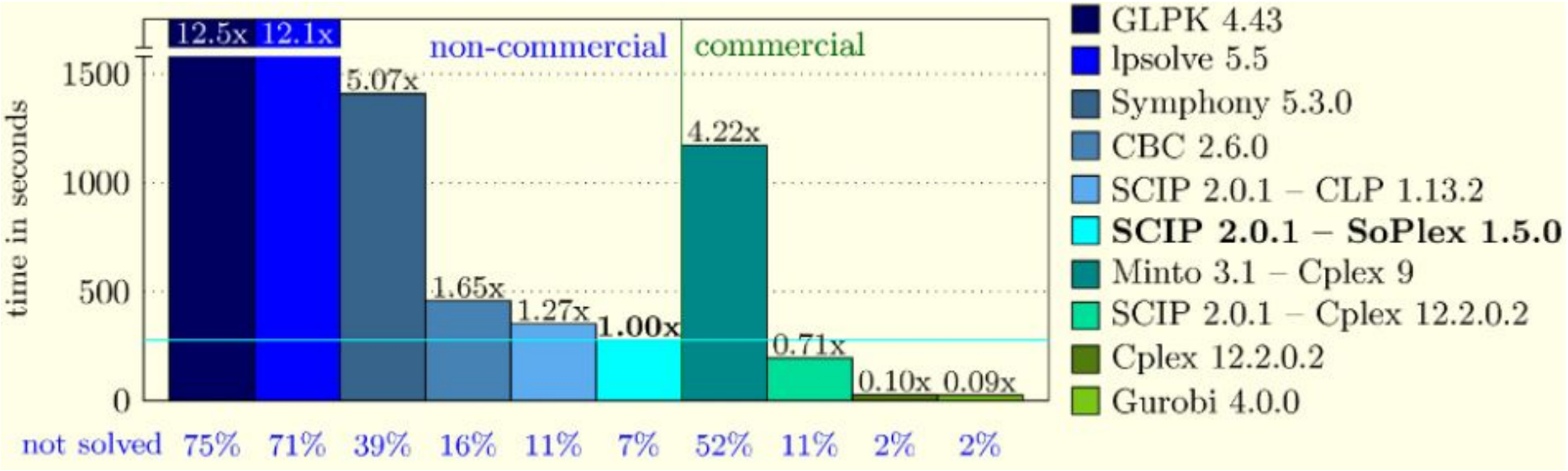


Figura. 1. Resultados extraídos de: http://plato.asu.edu/ftp/milpf.html

Sumário

1. Mathematical Programming Language (AMPL/GMPL);
2. Exemplo: Problema da dieta;
3. Analisar de relatórios no GLPK.
4. Utilizar uma linguagens de programação (C++) integrada a um Solver (ILOG Cplex Concert Technology).
    a. Exemplo: Problema de atribuição.

AMPL(GMPL)

Modelos AMPL envolvem: **variáveis, restrições, objetivos, parâmetros e conjuntos**.

Case sensitive: faz distinção entre maiúscula e minuscula.

Expressões terminam com ponto e vírgula.

Ex.: minimize obj: sum{j in Foods} cost[j]*quantity[j];

Strings são delimitadas por aspas simples ou duplas.

Ex.: ‘This is a string’, "This is another string"

Comentários são definidos por: # ou /* e */

Ex.: # isso é um comentário

/* isso é outro comentário. */

AMPL(GMPL)

Conjuntos:

Ex: set V = {1, 2, 3, 4, 5};

set A = {‘v1’, ‘v2’, ‘v3’, ‘v4’, ‘v5’};

set Foods;

Parâmetros: qualquer valor para uma instância de determinado problema.

Ex.: param c {i in V, j in V: i!= j } >= 0;

param cost{Foods} >= 0;

Declaração de variáveis:

Ex.: var x >= 0 <= 20;

var x {i in V, j in V: i != j} binary;

var quantity{Foods} >= 0;

AMPL(GMPL)

Declaração das restrições

Ex.: subject to {

<desigualdade 1>;

<desigualdade 2>;

}

ou

suject to cardinality: sum {e in E} x[e] = n-1;

subject to blue_limit: 0 <= PaintB <= 1000;

ou

s.t. minnutr{i in Nutrients}:

sum{j in Foods} amount[i,j]*quantity[j] >= minimum[i];

Exemplo: Problema da Dieta

Objetivo: Comprar alimentos de forma a determinar uma dieta que cumpra com as necessidades nutricionais mínimas.

Existem 3 opções de alimentos: **maça, pão e doce**

-	Nutrientes		
Alimento	**A**	**B**	**C**
Maça	5	5	10
Pão	2	10	1
Doce	3	0	0

A quantidade mínima necessária cada nutriente é:	Custo por unidade
30 unidades de nutriente A 50 unidades de nutriente B 30 unidades de nutriente C;	Maça: R$ 2,00 Pão: R$ 1,00 Doce: R$ 0,20

Formato LP

Problema dieta - GLPK/GMPL (Math-Prog)

Minimize

obj: 2 x1 + 1 x2 + 0.2 x3

subject to

nutrienteA: 5 x1 + 2 x2 + 3x3 >= 30

nutrienteB: 5 x1 + 10 x2 + 0x3 >= 50

nutrienteC: 10 x1 + 1 x2 + 0x3 >= 30

Bounds

0 <= x1

0 <= x2

0 <= x3

End

Comando: `glpsol --cpxlp dieta.lp -o dieta.txt`

Exemplo: Problema da Dieta (modelo)

```
# Conjuntos
        set Alimento;
        set Nutrientes;

# Variáveis
        var x{Alimento} >= 0;

# Parâmetros
        param custo{Alimento} >= 0;
        param quantidade{Nutrientes, Alimento} >= 0;
        param qtd_minima{Nutrientes} >= 0;


# Função Ojetivos
        minimize obj: sum{j in Alimento} custo[j]*x[j];

# Restrições
s.t. nutr_{i in Nutrientes}:
            sum{j in Alimento} quantidade[i,j]*x[j] >= qtd_minima[i];


end;
```

Exemplo: Problema da Dieta (dados)

```
set Alimento   := maca, pao, doce;

set Nutrientes := vitamina_a, vitamina_b, vitamina_c;

param custo :=
                    maca  2
                    pao  1
                    doce 0.2;

param quantidade:    maca   pao doce :=
          vitamina_a        5      2    3
          vitamina_b        5     10    0
          vitamina_c       10     1     0;

param qtd_minima   :=
          vitamina_a 30
          vitamina_b 50
          vitamina_c 30;

end;
```

Executar: glpsol --model dieta.mod --data dieta.data --output dieta.sol

Relatório (Problema da Dieta)

```
Problem:    dieta
Rows:       4
Columns:    3
Non-zeros:  10
Status:     OPTIMAL
Objective:  obj = 9.578947368 (MINimum)

   No.   Row name   St   Activity     Lower bound   Upper bound    Marginal
------ ------------ -- ------------ ------------- ------------- -------------
     1 obj          B        9.57895
     2 nutr_[nutriente_a]
                    NL            30            30             0.0666667
     3 nutr_[nutriente_b]
                    NL            50            50             0.0736842
     4 nutr_[nutriente_c]
                    NL            30            30              0.129825

   No. Column name  St   Activity     Lower bound   Upper bound    Marginal
------ ------------ -- ------------ ------------- ------------- -------------
     1 x[maca]      B        2.63158             0
     2 x[pao]       B        3.68421             0
     3 x[doce]      B        3.15789             0

End of output
```

example_cplex_2: - Problema de atribuição

Minimizar $C = \sum_{i,j=1}^{n} c_{ij} x_{ij}$

sujeito a:

$$\sum_{j=1}^{n} x_{ij} = 1 \quad , i = 1,2,\ldots,n$$

cada trabalhador é designado a uma só tarefa

$$\sum_{i=1}^{n} x_{ij} = 1 \quad , j = 1,2,\ldots,n$$

cada tarefa é executada apenas por um trabalhador

$$x_{ij} = 0,1 \ , i = 1,2,\ldots,n \ , j = 1,2,\ldots,n$$

Referências

Baixar GLPK: http://winglpk.sourceforge.net/.

Pequeno Tutorial: The GNU Linear Programming Kit, Part 1: Introduction to linear optimization.

GUSEK (Windows) = http://gusek.sourceforge.net/gusek.html

CPLEX: http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/

AMPL: http://ampl.com/