



Construção de Modelos de Otimização e Ferramentas Computacionais

Socorro Rangel

DMAp

Departamento de Matemática Aplicada

e-mail: socorro@ibilce.unesp.br

<http://www.ibilce.unesp.br/#!/departamentos/matematica-aplicada/docentes>

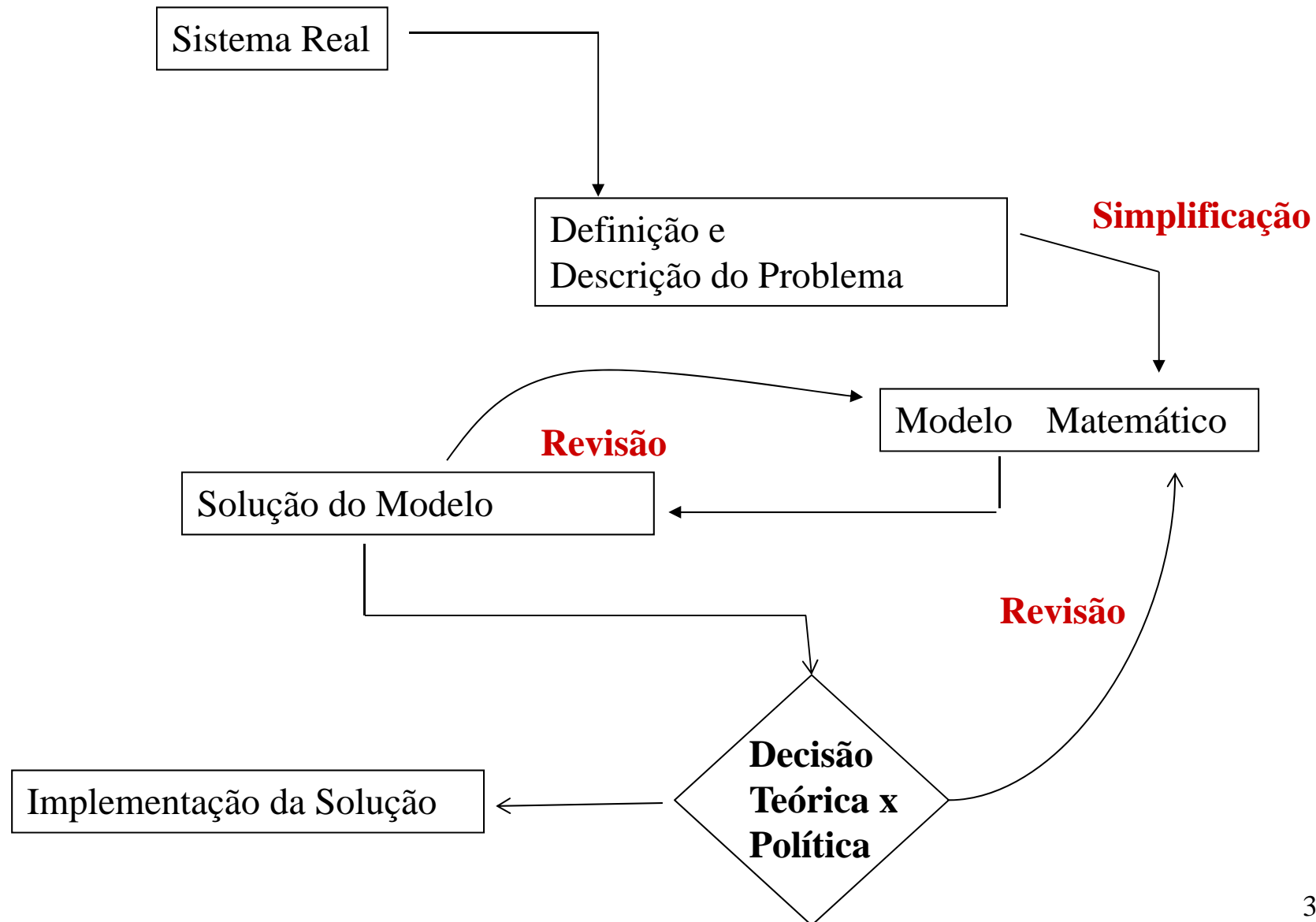
Motivação

Kantarovich (1939) (em Dantzig, 1963, pg 22)

"Existem duas maneiras de aumentar a eficiência de uma loja, empresa, ou indústria:

- 1. Uma delas requer a melhoria tecnológica, isto é, atualização dos equipamentos, mudança no processo tecnológico, descoberta de novos e melhores tipos de matéria prima.*
- 2. A outra maneira, até hoje muito menos utilizada, envolve melhorias na organização do planejamento e da produção. Isto é, melhorias no processo de distribuição do trabalho entre as máquinas da empresa, distribuição de matéria prima, combustível, entre outros fatores."*

Processo de Construção de um Modelo Matemático



Construção de um modelo

- Descreva com a maior riqueza de detalhes o problema a ser tratado
- Identifique a classe do modelo matemático mais apropriado
- Defina as variáveis , funções. Se necessário, simplifique o problema.

Elementos de um modelo de otimização

DECISÕES

Identificar as possíveis soluções

(Definir Variáveis de Decisão)

OBJETIVOS

Definir critérios de avaliação capazes de indicar que uma decisão é preferível a outras

(Definir Função Objetivo)

RESTRIÇÕES

Identificar quais as restrições que limitam as decisões a serem tomadas

(Definir Conjunto de Equações ou Inequações)

Forma Geral de um Modelo de Otimização

min (ou max) (função objetivo)

sujeito a

**(restrições principais -
equações ou inequações)**

(tipo das variáveis de decisão)

Classes de Modelos de Otimização

- Não linear
- Linear Contínuo
- Linear Inteiro
- Misto

Modelo de Otimização Linear Contínuo

$$\min(\text{ou max}) z = c_1x_1 + c_2x_2 + \dots + c_nx_n$$

sujeito a

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \sim b_1$$

$$a_{22}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \sim b_2$$

...

$$a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \sim b_m$$

$$x_1, x_2, \dots, x_n \geq 0$$

onde \sim pode ser \geq, \leq , ou $=$

Forma Padrão:

$$\min \quad z = c^t x$$

s.a

$$Ax = b$$

$$x \in \mathcal{R}_+^n$$

Exemplo - O Problema da Dieta

Problema: Paula deseja saber quanto gastar para fazer uma dieta alimentar que forneça diariamente toda a energia, proteína e cálcio que ela necessita. Seu médico recomendou que ela se alimente de forma a obter diariamente no mínimo 2000 kcal de energia, 65g de proteína e 800 mg de cálcio. O Valor nutritivo e o preço (por porção) de cada alimento que ela esta considerando comprar é dado na Tabela 1.

Tabela 1 – Valor nutritivo e custo dos alimentos

alimento	tamanho da porção	energia (kcal)	Proteína (g)	cálcio (mg)	preço p/ porção (centavos)
arroz	100g	170	3	12	14
ovos	2un	160	13	54	13
leite	237ml	160	8	285	9
feijão	260g	337	22	86	19

Construindo um modelo para o Problema da Dieta

Neste problema temos:

elementos conhecidos: valor nutritivo dos alimentos, custo dos alimentos

elementos desconhecidos: quanto consumir de cada alimento

objetivo a ser alcançado: obter uma dieta de baixo custo

restrições: a dieta deve fornecer uma quantidade mínima de nutrientes.

Construindo um modelo para o Problema da Dieta

Índices

A dieta deve ser feita a partir de 4 itens:

arroz, ovos, leite, feijão.

Faça $j = 1, 2, 3, 4$ representar respectivamente cada um dos itens

VARIÁVEIS DE DECISÃO

Defina então:

x_j = número de porções adquirida do alimento j
para ser usada na dieta

Tipo das variáveis

$$x_j \geq 0, x_j \in \mathbb{R} \quad \text{Divisibilidade}$$

Construindo um modelo para o Problema da Dieta

Objetivo

Obter a dieta de menor custo possível.

Proporcionalidade:

1 porção de arroz \Rightarrow 14 centavos,

2 porções de arroz \Rightarrow 28 centavos,

x_1 porções de arroz \Rightarrow $14 * x_1$ centavos.

gasto associado a compra de ovos: $13 * x_2$

Aditividade

gasto total com arroz e ovos é dado pôr:

$$14 x_1 + 13 x_2$$

Construindo um modelo para o Problema da Dieta

Custo total da dieta é então:

$$z = 14x_1 + 13x_2 + 9x_3 + 19x_4$$

Custo do:

arroz

ovos

leite

feijão

Objetivo

Obter a dieta de menor custo possível.

$$\min z = 14x_1 + 13x_2 + 9x_3 + 19x_4$$

Construindo um modelo para o Problema da Dieta

Restrições

Obter quantidade mínima de nutrientes:

energia:

1 porção de arroz \Rightarrow 170 kcal, x1 porções de arroz \Rightarrow 170 x1

1 porção de ovos \Rightarrow 160 kcal, x2 porções de ovos \Rightarrow 160 x2

1 porção de leite \Rightarrow 160 kcal, x3 porções de leite \Rightarrow 160 x3

1 porção de feijão \Rightarrow 337 kcal, x4 porções de feijão \Rightarrow 337 x4

quantidade total de energia \geq quantidade mínima necessária

Proporcionalidade e aditividade

Temos então:

$$170 x_1 + 160 x_2 + 160 x_3 + 337 x_4 \geq 2000$$

Modelo de Otimização Linear Contínuo Para o Problema da Dieta

$$\min z = 14x_1 + 13x_2 + 9x_3 + 19x_4 \quad (\text{função-objetivo})$$

sujeito a: (restrições)

$$170x_1 + 160x_2 + 160x_3 + 337x_4 \geq 2000 \quad (\text{energia})$$

$$3x_1 + 13x_2 + 8x_3 + 22x_4 \geq 65 \quad (\text{proteína})$$

$$12x_1 + 54x_2 + 285x_3 + 86x_4 \geq 800 \quad (\text{cálcio})$$

$$x_j \geq 0, \quad j = 1, 2, 3, 4 \quad (\text{tipo das variáveis})$$

Solução Para o Problema da Dieta

Função Objetivo: 112.500000000

VARIÁVEL	VALOR
X1	0.00 (arroz)
X2	0.00 (ovos)
X3	12.50 (leite)
X4	0.00 (feijão)

Isto é consumir $12.5 * 237\text{ml} = 2,9625 \text{ l}$ de leite
e gastar com a dieta 112,5 u.m.

Esta solução é aceitável?

Novo Modelo Para o Problema da Dieta

Se limitarmos a quantidade de leite na dieta:

No **máximo 2 porções**

$$\min z = 14x_1 + 13x_2 + 9x_3 + 19x_4$$

sujeito a:

$$170x_1 + 160x_2 + 160x_3 + 337x_4 \geq 2000$$

$$3x_1 + 13x_2 + 8x_3 + 22x_4 \geq 65$$

$$12x_1 + 54x_2 + 285x_3 + 86x_4 \geq 800$$

$$x_j \geq 0, j = 1, 2, 3, 4 \quad x_3 \leq 2$$

Nova Solução Para o Problema da Dieta

Função Objetivo: 112,72

VARIÁVEL	VALOR
----------	-------

X1	0,00 (arroz)
----	--------------

X2	0.00 (ovos)
----	-------------

X3	2.00 (leite)
----	--------------

X4	2.03 (feijão)
----	---------------

Isto é consumir:

$2 * 237\text{ml} = 474 \text{ ml de leite}$

$4,99 * 260\text{g} = 1297,4 \text{ g de feijão}$

e gastar com a dieta **112,72 u.m.**



Ferramentas Computacionais

Sistemas de Resolução

- Comerciais
 - CPLEX, XPRESS-MP,
 - Problema de otimização: contínua, inteira, quadrática
 - Arquivos no formato:MPS, próprio (algébrico)
 - Possuem linguagem de modelagem
 - GUROBI
- Não-Comerciais
 - CLP (COIN-OR Linear Program Solver)
 - LPSOLVE

Sistemas Algébricos de Modelagem: Objetivos

- Interface com sistemas de resolução
- Separar o Modelo dos dados
- Facilitar a construção de um modelo
- Documentar
- Facilitar a Manutenção do modelo

Sistemas Algébricos de Modelagem: Estrutura Geral

- **Conjuntos e índices**
 - locais: {Rio, SP, Goiânia}, códigos: {A11, B45}, mês: {jan, fev, ...}
- **dados, parâmetros, tabelas**
 - Separa o modelo de um exemplar do mesmo
 - fornecidos em arquivos de dados; retirados de planilhas de cálculo ou banco de dados
- **variáveis de decisão**
 - agrupar por tipos, definir para subconjuntos de índices
- **função objetivo**
 - linear ou não linear
- **Restrições**
 - agrupar por tipos e expandir, definir para subconjuntos de índices

Linguagens de Modelagem

MPL

- Modelagem:
 - otimização contínua, inteira, não linear
- Formato de arquivos (MPS, CPLEX,...)
- Conexão com EXCEL, Banco de dados
- Gráfico da Estrutura da matriz de restrições
- Conexão com sistemas de resolução (CPLEX, FORTMP,...)

XPRESS-MOSEL

- Linguagem Procedural
- Integração com Linguagens de Programação (C, Java, Visual Basic)

AMPL

- Linguagem Procedural
- Modelagem
 - otimização contínua, inteira, quadrática
- Interface gráfica com poucos recursos
- Permite a criação de subrotinas

Linguagens de Modelagem:

Principais Comandos

MPL

TITLE

INDEX

DATA

VARIABLES

MODEL

MIN (ou MAX)

SUBJECT TO

END

XPRESS-MOSEL

MODEL nome do model

Instruções para compilação

Definição de parâmetros

Definição do modelo

Definição de algoritmos

END-MODEL

AMPL

SET

define um índice;

PARAM

define uma estrutura (vetor ou matriz) que irá armazenar os elementos conhecidos do exemplar, fornecidos no arquivo nomemodelo.dat;

VAR

define variáveis de decisão;

MINIMIZE (ou MAXIMIZA)

define a função-objetivo e o critério de otimização

SUBJECT TO

define um conjunto de restrições

MPL: O Problema da Dieta

Índices

{ Dieta.mpl }

{ Problema da Dieta (Chvátal, 1986)

Determinar uma combinação de alimentos que forneça uma quantidade mínima de nutrientes }

TITLE

Dieta

INDEX

! Considerar na dieta os alimentos abaixo, Tamanho da porção: 100g, 2un, 237ml, 260g respectivamente;

alimento = (arroz, ovos, leite, feijao);

! Nutrientes necessários na dieta

nutriente = (energia, proteina, calcio);

MPL: O Problema da Dieta

Dados

DATA

! Custo de cada alimento a ser considerado na dieta, por porção

`preco[alimento] = (14 13 9 19);`

! Quantidade mínima total de cada nutriente na dieta;

`nivel[nutriente] = (2000, 65 , 800);`

! Quantidade de nutrientes presente em cada tipo de alimento considerado na dieta, por porção;

`quant[alimento,nutriente] = (`

170	3	12
160	13	54
160	8	285
337	22	86)

`;`

MPL: O Problema da Dieta

Modelo

VARIABLES

comprar[alimento];

MODEL

MIN Custo_total = **SUM**(alimento: preco*comprar) ;

SUBJECT TO

N_[nutriente] : **SUM**(alimento: quant*comprar) > nivel[nutriente];

END

MPL: O Problema da Dieta

Modelo gerado no Formato LP

```
\ dieta.lp
\ Generated with the MPL Modeling System
\ Constraints:      3      Variables:      4      Nonzeros:      12
\ Density:         100 %
MINIMIZE
  Custo_to: 14 cp_arr + 13 cp_ov + 9 cp_lei + 19 cp_fj
SUBJECT TO
  N_ene: 170 cp_arr + 160 cp_ov + 160 cp_lei + 337 cp_fj >= 2000
  N_prot: 3 cp_arr + 13 cp_ov + 8 cp_lei + 22 cp_fj >= 65
  N_Cal: 12 cp_arr + 54 cp_ov + 285 cp_lei + 86 cp_fj >= 800
END
```

Os sistemas de
resolução supõem
que $x \geq 0$

MPL: O Problema da Dieta

Manutenção do Modelo

Mudança no problema e nos dados: O que muda no modelo?

- Se os preços dos alimentos mudarem?
 - Mudar a seção **DATA**
 - ! Custo de cada alimento a ser considerado na dieta
 - `preco[alimento] = (11 10 5 15);`
- Se quisermos restringir a quantidade de cada tipo de alimento usada na dieta?
 - Mudar seção **DATA** e o conjunto de restrições
- Se quisermos incluir mais alimentos na dieta?
 - Mudar as seções **INDEX** e **DATA**

MPL: O Problema da Dieta

Restrição quanto à quantidade de Alimentos

DATA

!limite máximo de leite na dieta;

lim =2 ;

BOUNDS

{ limite máximo de leite na dieta,

neste caso apenas o consumo de leite esta limitado}

comprar[alimento=leite]<=lim;

MPL: O Problema da Dieta

Novo Modelo

```
\ dieta.lp
\ Generated with the MPL Modeling System
\ Constraints:      3      Variables:      4      Nonzeros:      12
\ Density:         100 %
MINIMIZE
  Custo_to: 14 cp_arr + 13 cp_ov + 9 cp_lei + 19 cp_fj
SUBJECT TO
  N_ene: 170 cp_arr + 160 cp_ov + 160 cp_lei + 337 cp_fj >= 2000
  N_prot: 3 cp_arr + 13 cp_ov + 8 cp_lei + 22 cp_fj >= 65
  N_Cal: 12 cp_arr + 54 cp_ov + 285 cp_lei + 86 cp_fj >= 800
BOUNDS
  cp_lei <= 2
END
```

Manutenção do Modelo:

Inclusão de Alimentos

INDEX

! Considerar na dieta os alimentos abaixo, Tamanho da porção: 100g, 2un, 237ml, 260g, **50g**, respectivamente;

alimento = (arroz, ovos, leite, feijao, **verdura**);

DATA

! custo de cada alimento a ser considerado na dieta, por porção;

preco[alimento] = (14 13 9 19 **2**);

! Quantidade de nutrientes presente em cada tipo de alimento considerado na dieta, por porção;

quant[alimento,nutriente] = (170 3 12
 160 13 54
 160 8 285
 337 22 86
 21 **1.8** **78**);

Edit Search Project Run View Graph Options Window Help



About MPL for Windows



MPL Modeling System
Release 4.2g

Copyright © 1988-2004 Maximal Software, Inc.
All rights reserved.

Max problem size: 300 constraints.
Expiration date: January 31, 2006

Student Version
Single User License
Serial No.: 5230171039

OK

Model Definitions

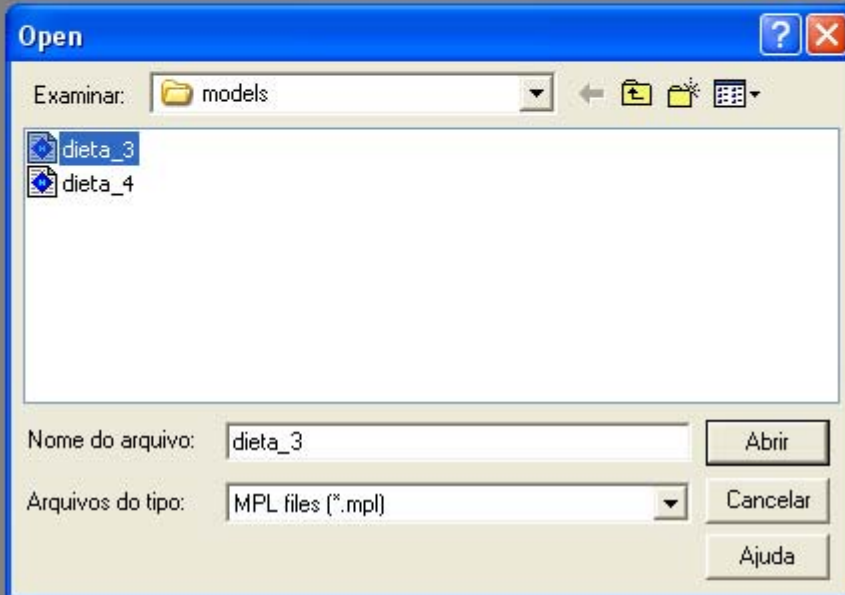
Model not parsed

View

Goto

Messages

Model file: dieta_1.mpl



MPL for Windows 4.2 - [C:\models\dieta_3.mpl]

File Edit Search Project Run View Graph Options Window Help

{ Dieta.mpl }
{ Problema da Dieta (Adaptado de Chuváta, 1986)
Determinar uma combinação de alimentos que forneça uma quantidade mínima de nutrientes}
TITLE
Dieta

INDEX
alimento = (arroz, ovos, leite, feijao, verdura) -> (arr, ov, lei, fj, vd)
nutriente = (energia, proteina, Caloria) -> (ene, prot, Cal)

DATA
! Custo de cada alimento a ser considerado na dieta
preco[alimento] = (14 13 9 19 2);

! Quantidade mínima total de cada nutriente na dieta;
nivel[nutriente] = (2000 65 800);

! Quantidade de nutrientes presente em cada tipo de alimento considerado na dieta;
quant[alimento, nutriente] = (
170 3 12
160 13 54
160 8 285
337 22 86
50 18 78);

! limite máximo leite na dieta;
lim[alimento] = (3,1,2,3,2) ;

VARIABLES
comprar[alimento] -> cp_

MODEL
MIN Custo_total = SUM(alimento: preco*comprar) ;

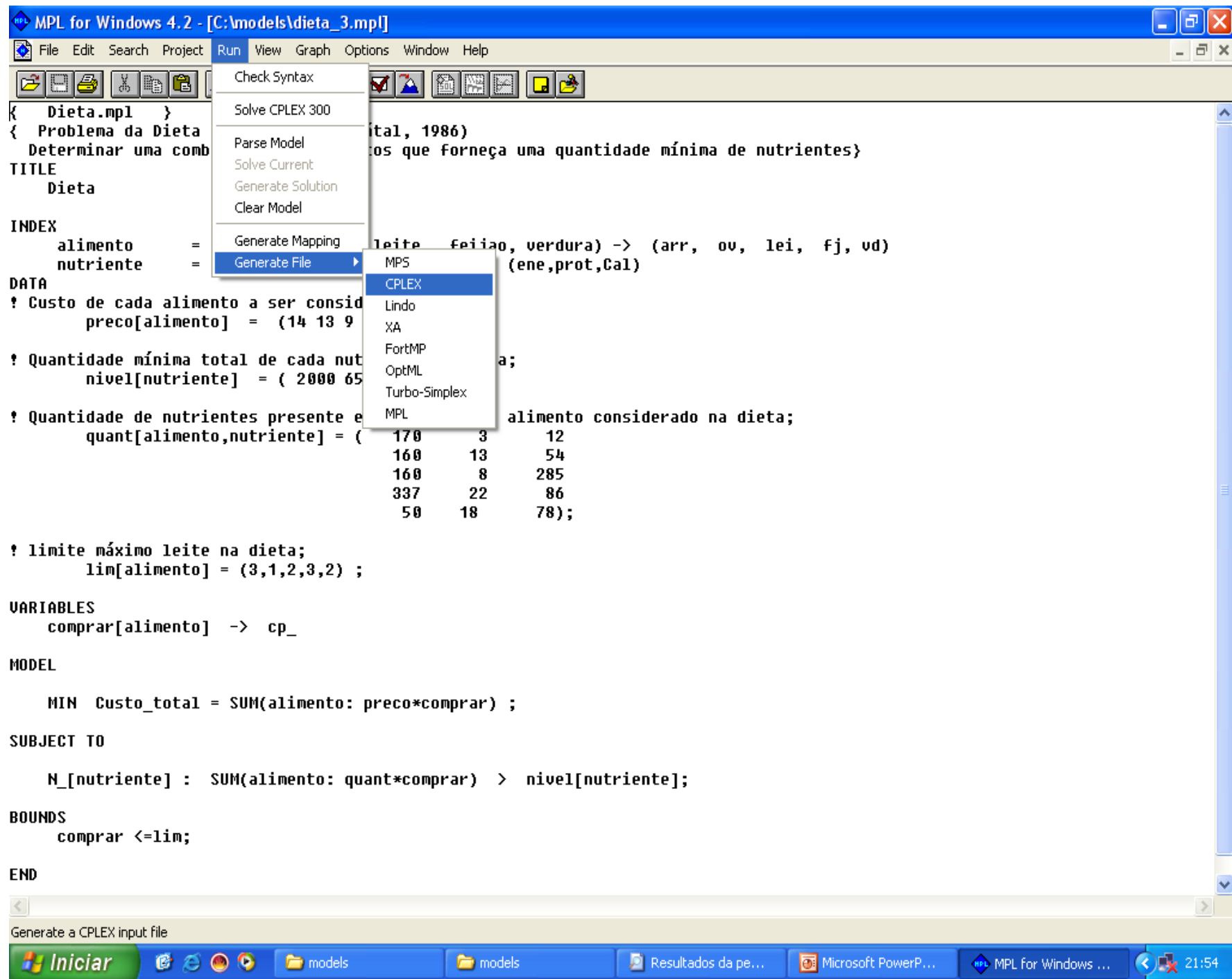
SUBJECT TO
N_[nutriente] : SUM(alimento: quant*comprar) > nivel[nutriente];

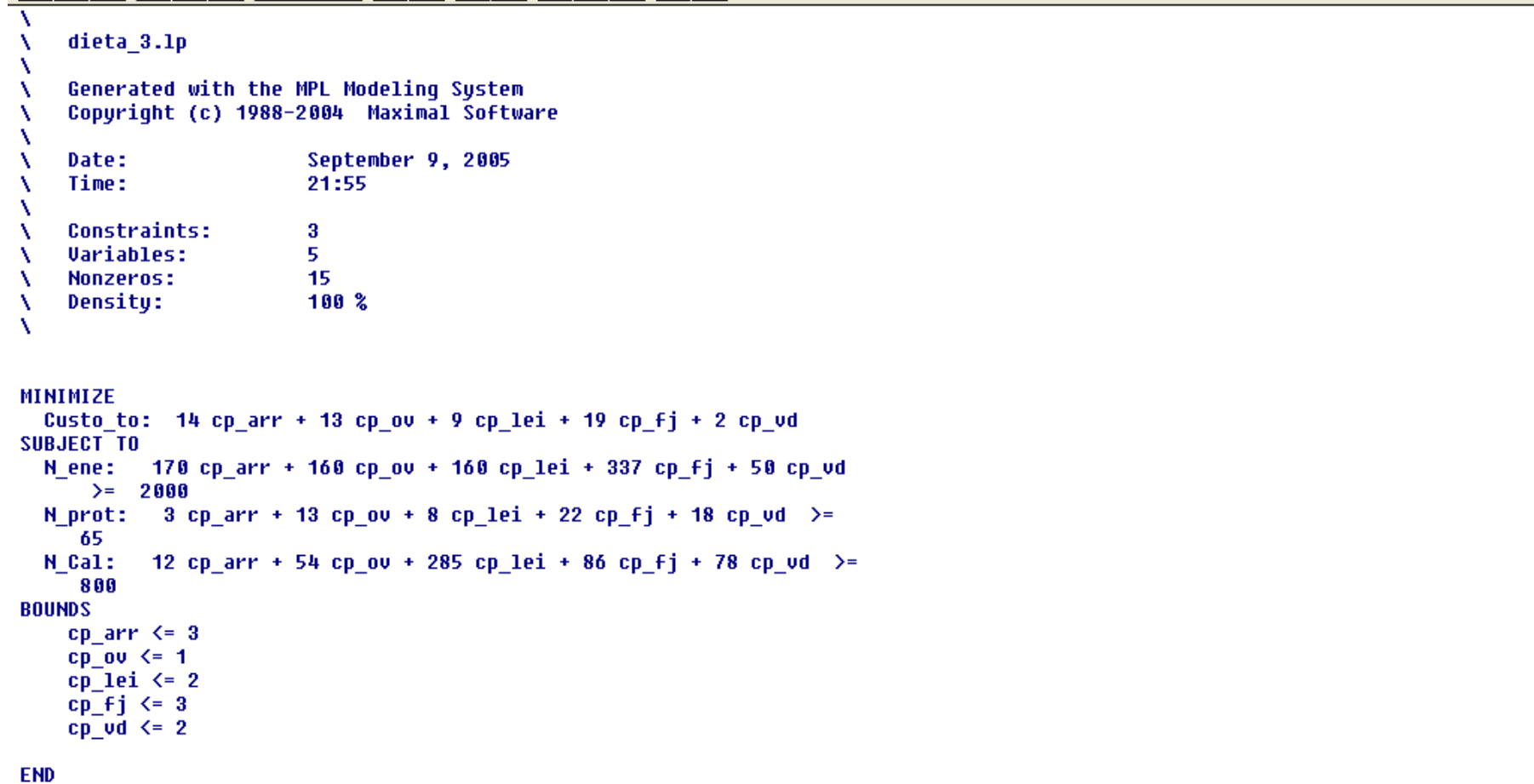
BOUNDS
comprar <= lim;

END

Main model file: dieta_3.mpl 1:1 Solved

Windows taskbar: Iniciar, models, Resultados da pe..., Microsoft PowerP..., MPL for Windows ... 21:52





Parsed

MPL for Windows 4.2 - [C:\models\dieta_3.mpl]

File Edit Search Project Run View Graph Options Window Help

Check Syntax
Solve CPLEX 300
Parse Model
Solve Current
Generate Solution
Clear Model
Generate Mapping
Generate File

```

{ Dieta.mpl }
{ Problema da Dieta
  Determinar uma combinação ótima de alimentos que forneça uma quantidade mínima de nutrientes}
TITLE
  Dieta

INDEX
  alimento      =
  nutriente     =

DATA
  ! Custo de cada alimento a ser considerado na dieta
  preco[alimento] = (14 13 9 19 2);

  ! Quantidade mínima total de cada nutriente na dieta;
  nivel[nutriente] = ( 2000 65 800 );

  ! Quantidade de nutrientes presente em cada tipo de alimento considerado na dieta;
  quant[alimento,nutriente] = (
    170    3    12
    160    13   54
    160    8   285
    337    22   86
    50     18   78);

  ! limite máximo leite na dieta;
  lim[alimento] = (3,1,2,3,2) ;

VARIABLES
  comprar[alimento] -> cp_

MODEL

  MIN  Custo_total = SUM(alimento: preco*comprar) ;

SUBJECT TO

  N_[nutriente] : SUM(alimento: quant*comprar) > nivel[nutriente];

BOUNDS
  comprar <= lim;

END

```

Solve the model using the default solver

Windows taskbar: Iniciar, models, Resultados da pe..., Microsoft PowerP..., MPL for Windows ..., 21:59

MPL for Windows 4.2 - [View File: dieta_3.sol]

File Edit Search Project Run View Graph Options Window Help

SOLUTION RESULT

Optimal solution found

MIN Custo_to = 125.6824

DECISION VARIABLES

VARIABLE comprar[alimento] :

alimento	Activity	Reduced Cost
arroz	2.4059	0.0000
ovos	1.0000	-0.1765
leite	2.0000	-4.1765
feijao	3.0000	-8.7529
verdura	2.0000	-2.1176

CONSTRAINTS

CONSTRAINT N[nutriente] :

nutriente	Slack	Shadow Price
energia	0.0000	0.0024
proteina	-73.2176	0.0000
Caloria	-266.8706	0.0000

FND

Main model file: dieta_3.mpl

Solved

22:02

XPRESS-MOSEL: Problema da Dieta

! Arquivo: dieta.mos

MODEL "dieta1"

USES "mmxprs" !use o sistema mmxprs para resolver o exemplar

DECLARATIONS

!definição dos **índices**

alimento = {"arroz", "ovos", "leite", "feijao"};

nutriente = {"energia", "proteina", "Caloria"};

!definição das estruturas para receber **dados**

! Custo de cada alimento a ser considerado na dieta

preco: **array**(alimento) **of real**;

! Quantidade mínima total de cada nutriente na dieta;

nivel: **array** (nutriente) **of real**;

! Nutrientes presente em cada tipo de alimento considerado na dieta;

quant: **array** (alimento,nutriente) **of real**;

! definição das **variáveis** de decisão- Número de porções de cada alimento que ira compor a dieta

comprar: **array**(alimento) **of mpvar**;

END-DECLARATIONS

XPRESS-MOSEL: Problema da Dieta

! Arquivo: dieta.mos (continuação)

! dados do exemplar

! Custo de cada alimento a ser considerado na dieta

preco := [14, 13, 9, 19];

! Quantidade mínima total de cada nutriente na dieta;

nivel := [2000, 65, 800];

! Quantidade de nutrientes presente em cada tipo de alimento considerado na dieta;

quant := [170, 3, 12,
 160, 13, 54,
 160, 8, 285,
 337, 22, 86];

XPRESS-MOSEL: Problema da Dieta

! Arquivo: dieta.mos (continuação)

!definição da função objetivo

Custo_total := **SUM**(j in alimento) preco(j)*comprar(j);

!definição das restrições

FORALL (i in nutriente)

SUM(j in alimento) quant(j,i)*comprar(j) >= nivel(i);

! Define Critério de otimização e Resolve o exemplar

minimize(Custo_total)

! Relatório da solução

writeln("Custo_total: ", getobjval)

writeln("Numero de porções a ser incluída na dieta: ")

forall(j in alimento) **writeln**(j, ": ", getsol(comprar(j)))

END-MODEL

AMPL: O Problema da Dieta

Arquivo: dieta.mod

Definição dos índices

set alimento;

set nutriente;

Estruturas para receber dados do exemplar

param preco {alimento};

param nivel {nutriente};

param quant {nutriente, alimento};

#variável de decisão:quanto comprar de cada alimento

var comprar {j in alimento} >=0;

#defini função-objetivo e critério de otimização

minimize custo_total: sum {j in alimento} preco[j] *comprar[j];

#Níveis mínimos de nutrientes devem ser satisfeitos

subject to N_ {i in nutriente}:

sum {j in alimento} quant[i,j] * comprar[j] >= nivel[i];

#Arquivo dieta.dat

set alimento := arroz ovos leite
feijao;

set nutriente := energia proteina
caloria;

param preco :=

arroz 14

ovos 13

leite 9

feijao 19;

param nivel :=

energia 2000

proteina 65

caloria 800;

param quant:

	arroz	ovos	leite	feijao	:=
energia	170	80	130	100	
proteina	3	6	6.1	6	
caloria	12	25	232	28;	

Endereços na WWW

- **Comerciais (versão de estudante ou Licença Acadêmica gratuita)**

MPL: <http://www.maximal-usa.com/>

XPRESS: <http://www.dashoptimization.com/>

AMPL: <http://www.ampl.com/>

GUROBI: <http://www.gurobi.com/products/gurobi-optimizer/try-for-yourself>

- **Não Comerciais**

CLP (COIN-OR Linear Program Solver)

<http://www.coin-or.org/Clp/>

LPSOLVE - <http://lpsolve.sourceforge.net/5.5/>

ZIMPL - <http://www.zib.de/koch/zimpl/>

Para Saber Mais

1. Rangel, S. *Introdução à construção de modelos de otimização linear e inteira*. 2. ed. São Carlos-SP: Sociedade Brasileira de Matemática Aplicada e Computacional-SBMAC, 2012. v. único. 82 p. (disponível em http://www.sbmac.org.br/arquivos/notas/livro_18.pdf)
2. Williams, H.P., *Model Building in Mathematical Programming*, Ed. John Wiley & Sons, 1990.
3. Wolsey, L., *Integer Programming*, Ed. John Wiley & Sons, 1998.
4. Arenales, M., Armentano, V., Morabito, R. E Yanasse, H.- *Pesquisa Operacional*, Elsevier, 2007.