



De 28/10 a 01/11

UENP - CAMPUS LUIZ MENEGHEL
CENTRO DE CIÊNCIAS TECNOLÓGICAS

VII SEMINÁRIO DE INFORMÁTICA
E TECNOLOGIA

DESIGNAÇÃO DE CORREDORES EM PERCURSOS DE UMA CORRIDA DE REVEZAMENTO

OMERO FRANCISCO BERTOL

UTFPR - Universidade Tecnológica Federal do Paraná, Sede Curitiba.
CPGEI - Pós-Graduação em Engenharia Elétrica e Informática Industrial.
Av. Sete de Setembro 3165, 80230-901, Curitiba/PR, Brasil.

SUMÁRIO

INTRODUÇÃO

PROBLEMA

OBJETIVO GERAL

METODOLOGIA

RESULTADOS

DISCUSSÃO E CONCLUSÕES

REFERÊNCIAS

Os modelos de Programação Matemática podem ser entendidos como um conjunto de equações, inequações e dependências lógicas que correspondem a relacionamentos apresentados por estruturas reais.

Em um modelo matemático, são incluídos três conjuntos principais de elementos:

- 1) Variáveis de decisão e parâmetros:** as variáveis de decisão são as incógnitas a serem determinadas pela solução do modelo e os parâmetros são valores fixos do problema (dados de entrada);
- 2) Restrições:** limitam os valores possíveis (ou viáveis) das variáveis;
- 3) Função objetivo:** é uma função matemática que define a qualidade da solução em função das variáveis de decisão empregadas. Engloba considerações de "maximização" de lucros ou "minimização" de custos, por exemplo.

INTRODUÇÃO

(2/2)

O GUSEK é uma interface de desenvolvimento para modelos de Programação Linear e Programação Linear Inteira Mista. O pacote consiste em uma versão customizada do editor de textos SciTE (*SCIntilla Text Editor*) integrada a uma versão pré-compilada do solver GLPK (*GNU Linear Programming Kit*) para a plataforma Win32.

```
1 # Trabalho Final - Disciplin: Otimização de Sistemas (Prof. Leandro Magatão)
2 # CPGEI - UTFPR Sede Curitiba
3 # Aluno: 40916- Omero Francisco Bertol
4 # Data: Agosto/2013 (Fase II)
5
6 # Modelos que Utilizam Grandezas Discretas
7 # Determinação de Tarefas (Corredor versus Percurso - Corrida de Revezamento)
8
9 param nc 'número de corredores', integer;
10 param np 'número de percursos', integer;
11
12 set I 'índice corredores' := 1..nc;
13 set J 'índice percursos' := 1..np;
14
15 # Indica se o corredor vai realizar o percurso ou não
16 # cp[i,j] = 1: o i-ésimo corredor 'sim' vai realizar o j-ésimo percurso
17 # cp[i,j] = 0: o i-ésimo corredor 'não' vai realizar o j-ésimo percurso
18 var cp 'corredor versus percurso' (i in I, j in J) binary;
19
20 # Tempo dos Corredores versus Percursos:
21 # tp[i,j] - tempo do i-ésimo corredor no j-ésimo percurso
22 param tp 'tempo corredor versus percurso' (i in I, j in J);
23
24 # Função Objetivo: minimizando a soma dos tempos (corredores versus percursos)
25 minimize z: sum (i in I, j in J) ( tp[i,j] * cp[i,j] );
26
27 # Cada Corredor realiza (número de percursos / número de corredores) Percursos
28 subject to CorredorXPercurso (i in I): sum (j in J) cp[i,j] = (np div nc);
29
30 # Cada Percurso é realizado por um e somente um Corredor
31 subject to PercursoXCorsdor (j in J): sum (i in I) cp[i,j] = 1;
32
33 solve;
```

```
14 rows, 144 columns, 288 non-zeros
144 integer variables, all of which are binary
Scaling...
A: min|a[ij]| = 1.000e+000 max|a[ij]| = 1.000e+000 ratio = 1.000e+000
Problem data seem to be well scaled
Constructing initial basis...
Size of triangular part = 23
Solving LP relaxation...
GLPK Simplex Optimizer, v4.47
14 rows, 144 columns, 288 non-zeros
* 0: obj = 3.257500000e+004 infeas = 1.000e+001 (1)
* 25: obj = 3.405300000e+004 infeas = 0.000e+000 (1)
* 76: obj = 3.103900000e+004 infeas = 0.000e+000 (1)
OPTIMAL SOLUTION FOUND
Integer optimization begins...
Gomory's cuts enabled
MIR cuts enabled
Cover cuts enabled
Clique cuts enabled
Creating the conflict graph...
The conflict graph has 2144 vertices and 1728 edges
* 76: mip = not found yet >>
* 76: >>>> 3.103900000e+004 >= 3.103900000e+004 0.0% (1: 0)
* 76: mip = 3.103900000e+004 >> tree is empty 0.0% (0: 1)
INTEGER OPTIMAL SOLUTION FOUND
Time used: 0.0 secs
Memory used: 0.3 Mb (294772 bytes)

..... P E R C U R S O S
.....
Corredor 1: 2435 3375 2700 4200 2700 3000 3640 3310 4971 3000 1900 2340
Corredor 2: 2580 3720 3300 4620 3180 3900 3360 3900 6300 3900 2400 3000
Corredor 3: 2400 3600 3000 4200 3000 3120 3000 3600 5400 3600 2100 2100
Corredor 4: 2400 3000 3000 3600 2700 2700 3520 2880 4440 2880 1800 2100
Corredor 5: 2400 3600 4200 3900 2700 3000 3000 3120 5280 3600 2340 2520
Corredor 6: 1920 2700 2400 4140 3000 2565 2469 2580 4453 2603 1623 1865
Corredor 7: 2280 2460 2400 2840 2940 2340 2840 2840 3960 2640 1540 1800
Corredor 8: 2160 3120 2640 3900 2700 3000 2580 3000 4212 3000 1320 2100
Corredor 9: 2280 3360 2700 4500 2700 2760 2640 3360 5400 3420 2100 2220
Corredor 10: 1944 2880 2065 3660 2115 2670 2055 2450 4440 3540 1700 3940
Corredor 11: 1502 2196 1792 2714 1740 1924 1740 2100 3384 2100 1320 1440
Corredor 12: 1700 2400 2000 3100 2150 2150 2050 2350 3700 2100 1550 1650

..... P E R C U R S O S
.....
Corredor 1: ..... 3000 .....
Corredor 2: 2580 ..... 2100 .....
Corredor 3: ..... 1800 .....
Corredor 4: ..... 2700 .....
Corredor 5: ..... 2880 .....
Corredor 6: ..... 2580 .....
Corredor 7: ..... 1540 .....
Corredor 8: ..... 1320 .....
Corredor 9: ..... 2065 .....
Corredor 10: ..... 3940 .....
Corredor 11: ..... 1440 .....
Corredor 12: ..... 1650 .....

Tempo de conclusão da Prova:
18 horas
17 minutos
19 segundos
Model has been successfully processed
>EXIT code: 0 Time: 0.356
```



**Prefeitura
de Curitiba**

Secretaria do Esporte, Lazer e Juventude



CURITIBA



IX CORRIDA DE REVEZAMENTO DAS
NASCENTES DO IGUAÇU

A IX Corrida de Revezamento das Nascentes do Iguaçu tem por objetivo além de incentivar a prática esportiva e a integração metropolitana, divulgar as peculiaridades turísticas, naturais e ambientais dos municípios da Região Metropolitana de Curitiba. A Corrida se desenvolve ao longo de nove Municípios: **Piraquara, Quatro Barras, Pinhais, Campina Grande do Sul, Bocaiúva do Sul, Colombo, Almirante Tamandaré, Campo Magro e Curitiba**, cortados pelos afluentes do Rio Iguaçu tais como: Piraquara, Irai, Palmital, Atuba, Barigui e Passaúna, percorrendo estradas vicinais com paisagens bucólicas e natureza exuberante.

Distância: **108 km**

Data: **25 de agosto 2013 (domingo)**

Largada: **Piraquara às 6h15min**

Chegada: **Curitiba, Bosque São Cristóvão em Santa Felicidade**

Fonte: <http://www.curitiba.pr.gov.br/multimidia/00133623.pdf>

PROBLEMA

(3/3)

IX Corrida de Revezamento das Nascentes do Iguau:

- Modalidade de corrida de revezamento a pé;
- Equipes com no máximo 12 corredores;
- 12 percursos: Piraquara – Curitiba (Santa Felicidade).

	LARGADA	Piraquara em frente a Prefeitura	0 km
Percurso 1	Posto 1	Quatro Barras - Praça Borda do Camp	7,2 km
Percurso 2	Posto 2	Pinhais - Parque Castelo Branco	10,1 km
Percurso 3	Posto 3	Campina Grande do Sul - Centro Esportivo Ricieri Bernardi	7 km
Percurso 4	Posto 4	Colombo – Igreja da Roseira	12,2 km
Percurso 5	Posto 5	Bocaiúva do Sul - Esc. Antonio Ceccon	8,2 km
Percurso 6	Posto 6	Colombo - Escola Comunidade São João	8,9 km
Percurso 7	Posto 7	Colombo - Escola Mun. Bortolo Cavassin	7,1 km
Percurso 8	Posto 8	Almirante Tamandaré - Escola Mun. Inácio Lipinski	10 km
Percurso 9	Posto 9	Almirante Tamandaré - Escola Ver. Osvaldo Avelino Trevisan	14,8 km
Percurso 10	Posto 10	Campo Magro - Lagoa	9,4 km
Percurso 11	Posto 11	Campo Magro - Escola Mun. João Menegusso Filho	6,3 km
Percurso 12	CHEGADA	Curitiba - Parque São Cristóvão (Santa Felicidade)	6,8 km
TOTAL:			108 km

OBJETIVO GERAL

Apresentar a implementação da solução de um problema para Designação de Corredores nos Percursos da IX Corrida de Revezamento das Nascentes do Iguaçu usando o ambiente de modelagem GUSEK.

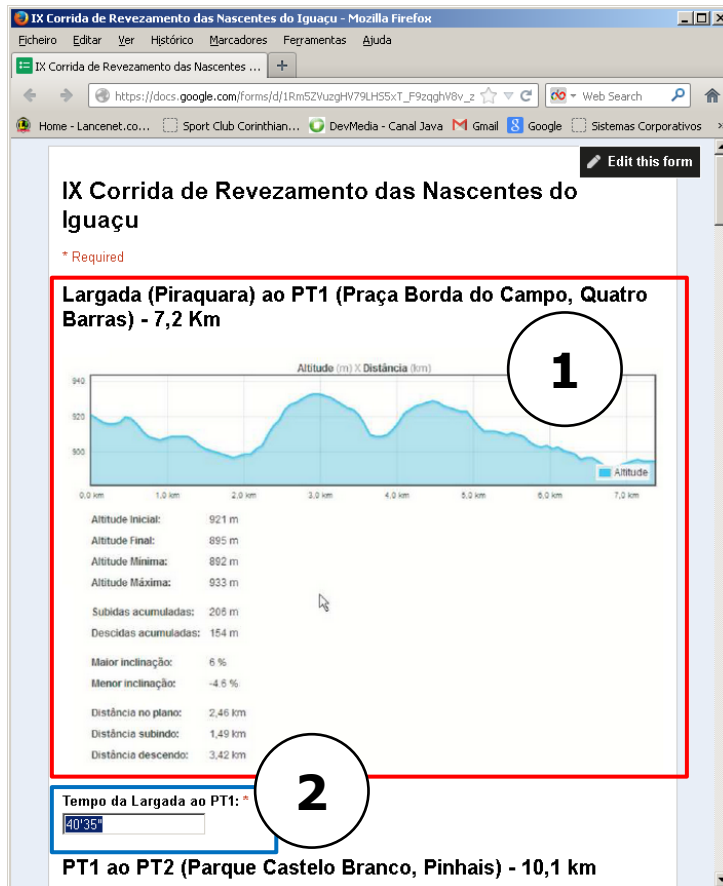
Problema de designação de corredores em percursos de uma corrida de revezamento com o objetivo de minimizar o tempo de conclusão da prova.

Na elaboração do modelo para abordar este cenário foi necessário levantar o desempenho dos componentes da equipe do revezamento.

Qual os tempos médios de cada corredor na conclusão de cada um dos 12 percursos ?

- 12 corredores foram pré-selecionados
- cada corredor respondeu um formulário web:
https://docs.google.com/forms/d/1Rm5ZVuzgHV79LHS5xT_F9zqghV8v_zD4JzqRlpha9fM/viewform (acesso em 23/ago./2013)

Levantamento dos tempos médios:



1. distância e altimetria
2. tempo médio percurso
3. respostas enviadas

3

	Tempo da Largada ao PT1:	Tempo do PT2 ao PT3:	Tempo do PT3 ao PT4:	Tempo do PT4 ao PT5:	Tempo do PT5 ao PT6:	Tempo do PT6 ao PT7:	Tempo do PT7 ao PT8:	Tempo do PT8 ao PT9:	Tempo do PT9 ao PT10:	Tempo do PT10 ao PT11:	Tempo do PT11 a Chegada:	Nome:	
1	40'35"	56'15"	45'	70'	45'	50'	44'	52'	82'51"	50'	33'	39'	Omero Francisco Bertol
2	43	62	55	77	63	65	56	65	105	65	65	40	50 Edén Ricardo Dosciatti
4	40	60	50	70	50	52	50	60	90	60	35	35	35 Otén Natalino Dutra
5	40	60	50	65min	45	45	45	48	74	48	30	35	35 Marcio
6	40	60	70 min	65min	45	50	52	52	88	55	39	42	42 Juca
7	31.68	44.94	41	69	50	42.45	41.09	43	74.13	44.23	27.3	31.5	Richard Panga Bertol
8	38	41	40	1h04	34	39	34	39	1h06	44	26	30	rafael
9	38	52	44	65	45	50	44	43	50	72	50	32	35 douglas burgardt
10	38 min	58 min	45 min	1h 15 min	45 min	46 min	44 min	58 min	1h 30 min	57 min	35 min	37 min	Leandro Freitas Tonial
11	4.30	4.55	4.65	5.00	4.30	5.00	4.50	5.00	4.30	4.30	4.45	4.45	Mauricio Bonamigo
12	25,2	36,36	29,62	45,14	28,7	32,04	26,76	4,25	35,56,24	36,40	21,74	23,66	Rudinei Presotto Nunes
13													

Aplicativo para pré-processamento dos tempos médios:

Tempo Estimado: Hora: 0 Minutos: 39 Segundos: 0 [Adicionar]

Dados de Entrada: Tempos do Corredor x Percursos

2435	3375	2700	4200	2700	3000	2640	3120	4971	3000	1980	2340
------	------	------	------	------	------	------	------	------	------	------	------

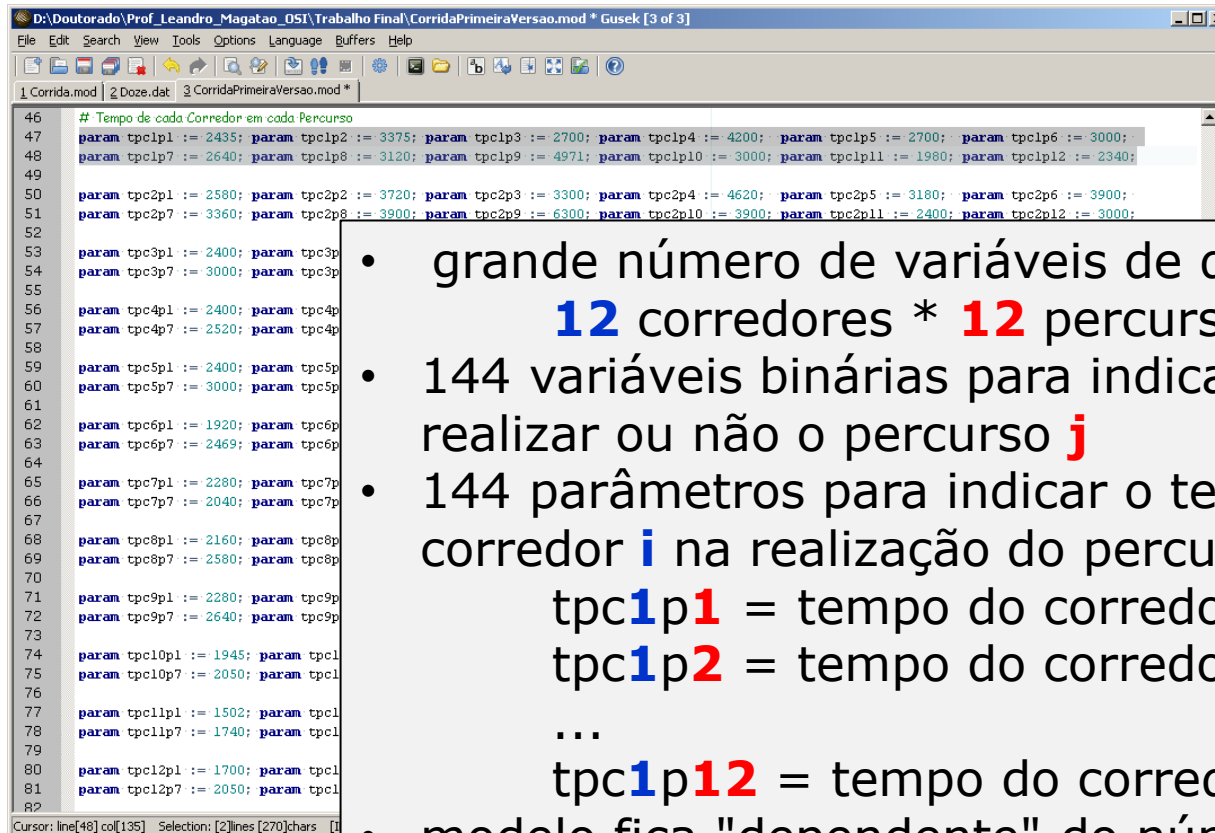
```
1 data:
2
3 param nc := 12;
4 param np := 12;
5
6 param tp :
7   →| 1 2 3 4 5 6 7 8 9 10 11 12 :=
8 →1 → 2435 3375 2700 4200 2700 3000 2640 3120 4971 3000 1980 2340
9 →2 → 2580 3720 3300 4620 3180 3900 3360 3900 6300 3900 2400 3000
10 →3 → 2400 3600 3000 4200 3000 3120 3000 3600 5400 3600 2100 2100
11 →4 → 2400 3000 3000 3660 2700 2700 2520 2880 4440 2880 1800 2100
12 →5 → 2400 3600 4200 3900 2700 3000 3000 3120 5280 3300 2340 2520
13 →6 → 1920 2700 2460 4140 3000 2565 2469 2580 4453 2663 1623 1865
14 →7 → 2280 2460 2400 3840 2040 2340 2040 2340 3960 2640 1560 1800
15 →8 → 2160 3120 2640 3900 2700 3000 2580 3000 4320 3000 1920 2100
16 →9 → 2280 3360 2700 4500 2700 2760 2640 3360 5400 3420 2100 2220
17 →10 → 1945 2980 2065 3660 2215 2670 2050 2650 4440 2540 1700 1940
18 →11 → 1502 2196 1792 2714 1740 1924 1740 2100 3384 2200 1320 1440
19 →12 → 1700 2400 2000 3100 2150 2150 2050 2350 3700 2500 1550 1650
20 ;
21
22 end;
```

tempos em segundos

RESULTADOS

(1/6)

Solução inicial: utilizando variáveis de decisão, parâmetros de entrada e restrições do tipo simples.



```
46 # Tempo de cada Corredor em cada Percurso
47 param tpc1p1 := 2435; param tpc1p2 := 3375; param tpc1p3 := 2700; param tpc1p4 := 4200; param tpc1p5 := 2700; param tpc1p6 := 3000;
48 param tpc1p7 := 2640; param tpc1p8 := 3120; param tpc1p9 := 4971; param tpc1p10 := 3000; param tpc1p11 := 1980; param tpc1p12 := 2340;
49
50 param tpc2p1 := 2580; param tpc2p2 := 3720; param tpc2p3 := 3300; param tpc2p4 := 4620; param tpc2p5 := 3180; param tpc2p6 := 3900;
51 param tpc2p7 := 3360; param tpc2p8 := 3900; param tpc2p9 := 6300; param tpc2p10 := 3900; param tpc2p11 := 2400; param tpc2p12 := 3000;
52
53 param tpc3p1 := 2400; param tpc3p2 := 2400; param tpc3p3 := 2400; param tpc3p4 := 2400; param tpc3p5 := 2400; param tpc3p6 := 2400;
54 param tpc3p7 := 3000; param tpc3p8 := 3000; param tpc3p9 := 3000; param tpc3p10 := 3000; param tpc3p11 := 3000; param tpc3p12 := 3000;
55
56 param tpc4p1 := 2400; param tpc4p2 := 2400; param tpc4p3 := 2400; param tpc4p4 := 2400; param tpc4p5 := 2400; param tpc4p6 := 2400;
57 param tpc4p7 := 2520; param tpc4p8 := 2520; param tpc4p9 := 2520; param tpc4p10 := 2520; param tpc4p11 := 2520; param tpc4p12 := 2520;
58
59 param tpc5p1 := 2400; param tpc5p2 := 2400; param tpc5p3 := 2400; param tpc5p4 := 2400; param tpc5p5 := 2400; param tpc5p6 := 2400;
60 param tpc5p7 := 3000; param tpc5p8 := 3000; param tpc5p9 := 3000; param tpc5p10 := 3000; param tpc5p11 := 3000; param tpc5p12 := 3000;
61
62 param tpc6p1 := 1920; param tpc6p2 := 1920; param tpc6p3 := 1920; param tpc6p4 := 1920; param tpc6p5 := 1920; param tpc6p6 := 1920;
63 param tpc6p7 := 2469; param tpc6p8 := 2469; param tpc6p9 := 2469; param tpc6p10 := 2469; param tpc6p11 := 2469; param tpc6p12 := 2469;
64
65 param tpc7p1 := 2280; param tpc7p2 := 2280; param tpc7p3 := 2280; param tpc7p4 := 2280; param tpc7p5 := 2280; param tpc7p6 := 2280;
66 param tpc7p7 := 2040; param tpc7p8 := 2040; param tpc7p9 := 2040; param tpc7p10 := 2040; param tpc7p11 := 2040; param tpc7p12 := 2040;
67
68 param tpc8p1 := 2160; param tpc8p2 := 2160; param tpc8p3 := 2160; param tpc8p4 := 2160; param tpc8p5 := 2160; param tpc8p6 := 2160;
69 param tpc8p7 := 2580; param tpc8p8 := 2580; param tpc8p9 := 2580; param tpc8p10 := 2580; param tpc8p11 := 2580; param tpc8p12 := 2580;
70
71 param tpc9p1 := 2280; param tpc9p2 := 2280; param tpc9p3 := 2280; param tpc9p4 := 2280; param tpc9p5 := 2280; param tpc9p6 := 2280;
72 param tpc9p7 := 2640; param tpc9p8 := 2640; param tpc9p9 := 2640; param tpc9p10 := 2640; param tpc9p11 := 2640; param tpc9p12 := 2640;
73
74 param tpc10p1 := 1945; param tpc10p2 := 1945; param tpc10p3 := 1945; param tpc10p4 := 1945; param tpc10p5 := 1945; param tpc10p6 := 1945;
75 param tpc10p7 := 2050; param tpc10p8 := 2050; param tpc10p9 := 2050; param tpc10p10 := 2050; param tpc10p11 := 2050; param tpc10p12 := 2050;
76
77 param tpc11p1 := 1502; param tpc11p2 := 1502; param tpc11p3 := 1502; param tpc11p4 := 1502; param tpc11p5 := 1502; param tpc11p6 := 1502;
78 param tpc11p7 := 1740; param tpc11p8 := 1740; param tpc11p9 := 1740; param tpc11p10 := 1740; param tpc11p11 := 1740; param tpc11p12 := 1740;
79
80 param tpc12p1 := 1700; param tpc12p2 := 1700; param tpc12p3 := 1700; param tpc12p4 := 1700; param tpc12p5 := 1700; param tpc12p6 := 1700;
81 param tpc12p7 := 2050; param tpc12p8 := 2050; param tpc12p9 := 2050; param tpc12p10 := 2050; param tpc12p11 := 2050; param tpc12p12 := 2050;
82
```

- grande número de variáveis de decisão e parâmetros:
12 corredores * 12 percursos
- 144 variáveis binárias para indicar se o corredor i vai realizar ou não o percurso j
- 144 parâmetros para indicar o tempo previsto pelo corredor i na realização do percurso j :
tpc1p1 = tempo do corredor 1 no percurso 1
tpc1p2 = tempo do corredor 1 no percurso 2
...
tpc1p12 = tempo do corredor 1 no percurso 12
- modelo fica "dependente" do número de corredores

RESULTADOS

(2/6)

Solução final adotada: utilizando a notação de conjuntos na declaração das variáveis de decisão, parâmetros de entrada e restrições.

Matriz 12 x 12 as linhas indicam os corredores e as colunas os percursos
 $tp[i,j]$ tempo do i -ésimo corredor no j -ésimo percurso

```
D:\Doutorado\Prof_Leandro_Magatao_OSI\Trabalho Final\Doze.dat - Gusek
File Edit Search View Tools Options Language Buffers Help
1 Doze.dat
1 data:
2
3 param nc := 12;
4 param np := 12;
5
6 param tp :
7
8   → 1 → 2 → 3 → 4 → 5 → 6 → 7 → 8 → 9 → 10 → 11 → 12 :=
9   → 1 → 2435 3375 2700 4200 2700 3000 2640 3120 4971 3000 1980 2340
10  → 2 → 2580 3720 3300 4620 3180 3900 3360 3900 6300 3900 2400 3000
11  → 3 → 2400 3600 3000 4200 3000 3120 3000 3600 5400 3600 2100 2100
12  → 4 → 2400 3000 3000 3660 2700 2700 2520 2880 4440 2880 1800 2100
13  → 5 → 2400 3600 4200 3900 2700 3000 3000 3120 5280 3300 2340 2520
14  → 6 → 1920 2700 2460 4140 3000 2565 2469 2580 4453 2663 1623 1865
15  → 7 → 2280 2460 2400 3840 2040 2340 2040 2340 3960 2640 1560 1800
16  → 8 → 2160 3120 2640 3900 2700 3000 2580 3000 4320 3000 1920 2100
17  → 9 → 2280 3360 2700 4500 2700 2760 2640 3360 5400 3420 2100 2220
18  → 10 → 1945 2980 2065 3660 2215 2670 2050 2650 4440 2540 1700 1940
19  → 11 → 1502 2196 1792 2714 1740 1924 1740 2100 3384 2200 1320 1440
20  → 12 → 1700 2400 2000 3100 2150 2150 2050 2350 3700 2500 1550 1650
21 ;
22 end;
```

onde:

nc – número de corredores
np – número de percursos

Arquivos de dados:

doze.dat nc := 12;
seis.dat nc := 6;
quatro.dat nc := 4;
tres.dat nc := 3;
dois.dat nc := 2;

Percursos por corredor:
(np **div** nc)

```
D:\Doutorado\Prof_Leandro_Magatao_OSI\Trabalho Final\Corrida.mod * Gusek [2 of 2]
File Edit Search View Tools Options Language Buffers Help
1 Doze.dat 2 Corrida.mod *
9 param nc 'número de corredores', integer;
10 param np 'número de percursos', integer;
11
12 set I 'índice corredores' := 1..nc;
13 set J 'índice percursos' := 1..np;
14
15 # Indica se o corredor vai realizar o percurso ou não
16 # cp[i,j] = 0; i-ésimo corredor "não" vai realizar o j-ésimo percurso
17 # cp[i,j] = 1; i-ésimo corredor "sim", vai realizar o j-ésimo percurso
18 var cp 'corredor versus percurso' {i in I, j in J} binary;
19
20 # Tempo dos Corredores versus Percursos:
21 # tp[i,j] - tempo do i-ésimo corredor no j-ésimo percurso
22 param tp 'tempo corredor versus percurso' {i in I, j in J};

Cursor: line[25] col[1] Selection: [0]lines [0]chars [INS] [CR+LF] File: Corrida.mod, 70 lines
```

- Elementos do modelo:
- 1) Variáveis de decisão e parâmetros
 - 2) Função objetivo
 - 3) Restrições

Elementos do modelo:

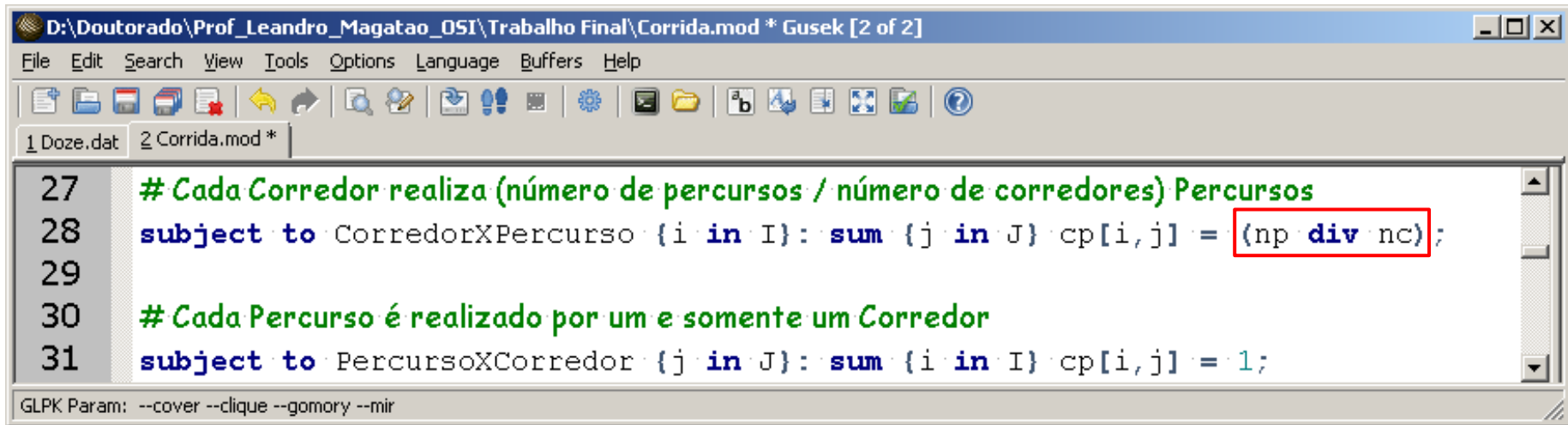
- 1) Variáveis de decisão e parâmetros
- 2) Função objetivo**
- 3) Restrições

```
D:\Doutorado\Prof_Leandro_Magatao_OSI\Trabalho Final\Corrida.mod * Gusek [2 of 2]
File Edit Search View Tools Options Language Buffers Help
1 Doze.dat 2 Corrida.mod *
24 # Função Objetivo: minimizando a soma dos tempos (corredores versus percursos)
25 minimize z: sum {i in I, j in J} ( tp[i,j] * cp[i,j] );
Cursor: line[28] col[1] Selection: [0]lines [0]chars [INS] [CR+LF] File: Corrida.mod, 77 lines
```

se $cp[i,j] = 0$ **então** tempo $tp[i,j]$ não será considerado na soma
se $cp[i,j] = 1$ **então** tempo $tp[i,j]$ será considerado na soma

Elementos do modelo:

- 1) Variáveis de decisão e parâmetros
- 2) Função objetivo
- 3) Restrições**



```
D:\Doutorado\Prof_Leandro_Magatao_OSI\Trabalho Final\Corrida.mod * Gusek [2 of 2]
File Edit Search View Tools Options Language Buffers Help
1 Doze.dat 2 Corrida.mod *
27 # Cada Corredor realiza (número de percursos / número de corredores) Percursos
28 subject to CorredorXPercurso {i in I}: sum {j in J} cp[i,j] = {np div nc};
29
30 # Cada Percurso é realizado por um e somente um Corredor
31 subject to PercursoXCorredor {j in J}: sum {i in I} cp[i,j] = 1;
GLPK Param: --cover --clique --gomory --mir
```

onde:

I – índice corredores
J – índice percursos

np = número de percursos
nc = número de corredores

RESULTADOS

(6/6)

```
D:\Doutorado\Prof_Leandro_Magatao_DSI\Trabalho Final\Corrida.mod - Gusek [2 of 2]
File Edit Search View Tools Options Language Buffers Help
1 Doze.dat 2 Corrida.mod
33 solve;
34
35 printf "\n";
36 printf "----- PERCURSOS ----- \n";
37 printf "1..2..3..4..5..6..7..8..9..10..11..12.\n";
38 printf "----- \n";
39 for {i in I} {
40   printf "Corredor %2d:", i;
41   for {j in J} {
42     printf "%4d", tp[i,j];
43   }
44   printf "\n";
45 }
46 printf "----- \n\n";
47
48 printf "----- PERCURSOS ----- \n";
49 printf "1..2..3..4..5..6..7..8..9..10..11..12.\n";
50 printf "----- \n";
51 for {i in I} {
52   printf "Corredor %2d:", i;
53   for {j in J} {
54     printf "%4s", if (cp[i,j] = 1) then tp[i,j] else "....";
55   }
56   printf "\n";
57 }
58 printf "----- \n";
59
60 printf "\nTempo de Conclusão da Prova:\n";
61 # converte o tempo total (z) em horas (h), minutos (m) e segundos (s)
62 printf "%2.0f horas\n", (z div 3600);
63 printf "%2.0f minutos\n", (z mod 3600) div 60;
64 printf "%2.0f segundos\n", (z mod 3600) mod 60;
65
```

```
24 rows, 144 columns, 288 non-zeros
144 integer variables, all of which are binary
Scaling...
A: min|a[ij]| = 1.000e+000 max|a[ij]| = 1.000e+000 ratio = 1.000e+000
Problem data seem to be well scaled
Constructing initial basis...
Size of triangular part = 23
Solving LP relaxation...
GLPK Simplex Optimizer, v4.47
24 rows, 144 columns, 288 non-zeros
0: obj = 3.257500000e+004 infeas = 1.000e+001 (1)
* 28: obj = 3.405000000e+004 infeas = 0.000e+000 (1)
* 76: obj = 3.103900000e+004 infeas = 0.000e+000 (1)
OPTIMAL SOLUTION FOUND
Integer optimization begins...
Gomory's cuts enabled
MIR cuts enabled
Cover cuts enabled
Clique cuts enabled
Creating the conflict graph...
The conflict graph has 2*144 vertices and 1728 edges
+ 76: mip = not found yet >= -inf (1; 0)
+ 76: mip = 3.103900000e+004 >= 3.103900000e+004 0.0% (1; 0)
+ 76: mip = 3.103900000e+004 >= tree is empty 0.0% (0; 1)
INTEGER OPTIMAL SOLUTION FOUND
Time used: 0.0 secs
Memory used: 0.3 Mb (294772 bytes)
----- PERCURSOS -----
1..2..3..4..5..6..7..8..9..10..11..12.
Corredor 1: 2435 3375 2700 4200 2700 3000 2640 3120 4971 3000 1980 2340
Corredor 2: 2580 3720 3300 4620 3180 3900 3360 3900 6300 3900 2400 3000
Corredor 3: 2400 3600 3000 4200 3000 3120 3000 3600 5400 3600 2100 2100
Corredor 4: 2400 3000 3000 3660 2700 2700 2520 2880 4440 2880 1800 2100
Corredor 5: 2400 3600 4200 3900 2700 3000 3000 3120 5280 3300 2340 2520
Corredor 6: 1950 2700 2460 4140 3000 2565 2460 2580 4453 2663 1623 1865
Corredor 7: 2280 2460 2400 3840 2040 2340 2040 2340 3960 2640 1560 1800
Corredor 8: 2160 3120 2640 3900 2700 3000 2580 3000 4320 3000 1920 2100
Corredor 9: 2280 3360 2700 4500 2700 2760 2640 3360 5400 3420 2100 2220
Corredor 10: 1945 2480 2065 3660 2215 2670 2050 2650 4440 2540 1700 1940
Corredor 11: 1502 2196 1792 2714 1740 1924 1740 2100 3384 2200 1320 1440
Corredor 12: 1700 2400 2000 3100 2150 2150 2050 2350 3700 2500 1550 1650
----- PERCURSOS -----
1..2..3..4..5..6..7..8..9..10..11..12.
Corredor 1: ..... 3000 .....
Corredor 2: 2580 ..... 3000 .....
Corredor 3: ..... 2700 ..... 2100 .....
Corredor 4: ..... 2700 ..... 1800 .....
Corredor 5: ..... 2700 .....
Corredor 6: ..... 2700 .....
Corredor 7: ..... 2340 .....
Corredor 8: ..... 2580 .....
Corredor 9: ..... 2760 .....
Corredor 10: ..... 2065 .....
Corredor 11: ..... 2714 .....
Corredor 12: ..... 3700 .....
Tempo de Conclusão da Prova:
8 horas
37 minutos
19 segundos
Model has been successfully processed
>Exit code: 0 Time: 0.356
```

GLPK Param: --cover --clique --gomory --mir -d Doze.dat

DISCUSSÃO E CONCLUSÕES

O presente trabalho destaca a importância da Pesquisa Operacional (PO) como ramo interdisciplinar da matemática aplicada que faz uso de modelos matemáticos e algoritmos fornecendo ferramentas quantitativas como forma de apoiar o processo de tomada de decisões.

A Programação Linear (PL) é uma das técnicas de sucesso da PO, que utiliza um conjunto de equações/inequações para representar relacionamentos de estruturas reais. Em um modelo matemático de PL, são incluídos três conjuntos principais de elementos: **(i)** as variáveis de decisão e parâmetros; **(ii)** um conjunto de restrições; e, **(iii)** uma função objetivo, ou função de avaliação do sistema.

Destacou-se o uso do ambiente GUSEK, um *software* livre destinado à modelagem e resolução de modelos de PL, na implementação da solução para o problema da **Designação de Corredores em Percursos de uma Corrida de Revezamento**.

MAGATÃO, Leandro. **Otimização de Sistemas**. Notas de aulas da disciplina. Curitiba-PR, agosto/2013. Disponível em <<http://pessoal.utfpr.edu.br/magatao/osi/>>. Acesso em 24/jun./2013.

8.2 Modelo com Variáveis Binárias (pág. 61)

8.8 Notação de Conjuntos para Elaboração de Modelos de PO (pág. 70)

9. Exemplos de Modelos com Grandezas Discretas (pág. 72)

BERTOL, Omero Francisco; DOSCIATTI, Eden Ricardo; MAGATÃO, Leandro. **Solução de um Problema de Programação Linear Usando o Ambiente de Modelagem Gusek**. Universidade Tecnológica Federal do Paraná - UTFPR, Câmpus Pato Branco. Pato Branco-PR, agosto/2013. Disponível em: <<http://www.pb.utfpr.edu.br/secompb/artigos.html>>. Acesso em 25/ago./2013.

SMELJ. **Secretaria Municipal do Esporte, Lazer e Juventude.** Curitiba-PR. Disponível em <http://www.curitiba.pr.gov.br/conteudo/equipe-smelj-secretaria-municipal-do-esporte-lazer-e-juventude/110>>. Acesso em 24/ago./2013.

GUSEK. **Página Oficial do Ambiente de Modelagem GUSEK.** Disponível em http://gusek.sourceforge.net/gusek_ptbr.html>. Acesso em 24/jun./2013.

SCITE. **SCIntilla Text Editor.** Disponível em <http://www.scintilla.org/SciTE.html>>. Acesso em 24/jun./2013.

GLPK. **Solver GNU Linear Programming Kit (GLPK).** Disponível em <http://www.gnu.org/software/glpk/>>. Acesso em 24/jun./2013.

<http://www.utfpr.edu.br/curitiba/estrutura-universitaria/diretorias/dirppg/programas/cpgei/inicio>

DESIGNAÇÃO DE CORREDORES EM PERCURSOS DE UMA CORRIDA DE REVEZAMENTO

Disciplina:

Otimização de Sistemas

Professor:

Leandro Magatão (magatao@utfpr.edu.br)



De 28/10 a 01/11

UENP - CAMPUS LUIZ MENEGHEL
CENTRO DE CIÊNCIAS TECNOLÓGICAS

VII SEMINÁRIO DE INFORMÁTICA
E TECNOLOGIA

DESIGNAÇÃO DE CORREDORES EM PERCURSOS DE UMA CORRIDA DE REVEZAMENTO

Obrigado

Contato:

Omero Francisco Bertol (omero@utfpr.edu.br)