Relatório do 3º Trabalho de Métodos Determinísticos de Investigação Operacional

João Silva A81761

Davide Matos A80970

Pedro Medeiros A80580

Bernardo Viseu A74618

8 de Janeiro de 2020

1 Introdução

Como é dito no próprio enunciado, o objetivo deste trabalho é desenvolvermos a nossa capacidade de analisar sistemas, criar modelos que os descrevam e, com a ajuda de programas adequados, validar estes modelos. Portanto, o que se segue é um estudo do caso proposto pelo enunciado, consoante as perguntas que nos são dirigidas.

2 Parte 0

2.1 Pergunta 1:

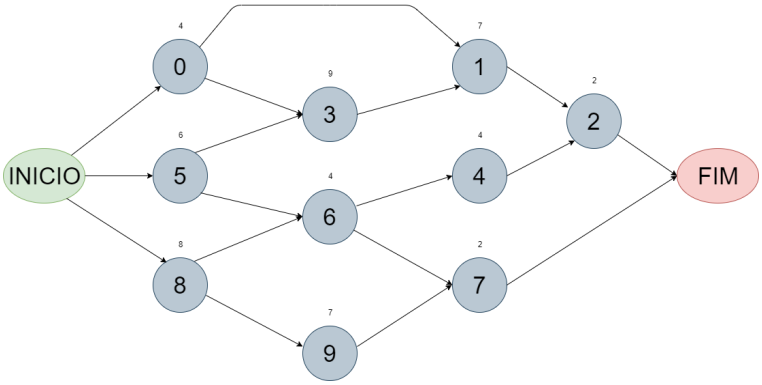

Figure 1: Grafo do projeto

```
/* Objective function */
min: tf ;

/* Variable bounds */
t1 >= t0 + 4;
t1 >= t3 + 9;
t3 >= t0 + 4;
t3 >= t5 + 6;
t6 >= t5 + 6;
t6 >= t8 + 8;
t9 >= t8 + 8;
t4 >= t6 + 4;
t4 >= t3 + 9;
t7 >= t6 + 4;
t7 >= t9 + 7;
t2 >= t1 + 7;
t2 >= t4 + 4;
tf >= t2 + 2;
tf >= t7 + 2;
t0 >= ti + 0;
t5 >= ti + 0;
t8 >= ti + 0;
```

Figure 2: Ficheiro .lp com restrições e função objetivo

Variables	result
	24
tf	24
t1	15
t0	0
t3	6
t5	0
t6	11
t8	0
t9	8
t4	18
t7	15
t2	22
ti	0

Figure 3: Output

Com este modelo, sabemos o instante de começo de cada atividade, ti, e a duração total do projeto representado por tf.

```
/* Objective function */
max: 4*xi0 + 6*xi5 + 8*xi8 + 9*x03 + 7*x01 + 4*x56 + 4*x86 + 7*x89
           + 2*x67 + 2*x97 + 2*x12 + 2*x42 + 0*x7f + 0*x2f + 7*x31 + 9*x53;

/* Variable bounds */
xi0 + xi5 + xi8 = 1;
xi0 = x03 + x01;
xi5 = x56 + x53;
xi8 = x86 + x89;
x03 + x53 = x31;
x56 + x86 = x64 + x67;
x89 = x97;
x01 + x31 = x12;
x64 = x42;
x57 + x97 = x7f;
x12 + x42 = x2f;

bin xi0,xi5,xi8,x03,x01,x56,x86,x89,x64,x67,x97,x31,x12,x42,x7f,x2f;
```

Figure 4: Ficheiro .lp com restrições e função objetivo

Variables	MILP ...	result
	24	24
xi0	0	0
xi5	1	1
xi8	0	0
x03	0	0
x01	0	0
x56	0	0
x86	0	0
x89	0	0
x67	0	0
x97	0	0
x12	1	1
x42	0	0
x7f	0	0
x2f	1	1
x31	1	1
x53	1	1
x64	0	0
x57	0	0

Figure 5: Resultado

Após determinar as atividades do projeto ficamos com o grafo representado na Figure 1, que é representado pelo modelo de PL da Figure 2, com a solução presente no Output, explicito na Figure 3.

2.2 Pergunta 2:

Podemos observar tanto pelo Output do lpSolve (Figure 3), como pela análise do Diagrama de Gantt (Figure 5) que a duração é de 24 dias.

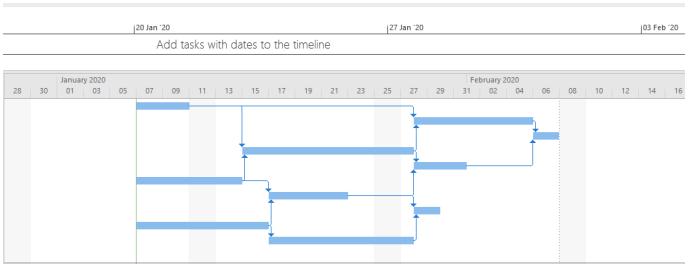

Figure 6: Diagrama de Gantt

3 Parte I

3.1 Pergunta 1

Para a resolução deste problema decidimos implementar 1 máquina para realizar as atividades 0, 5 e 8, passando a ser atividades não-simultaneas. Para isso aplicamos as restrições de não-simultaneidade no modelo PL, sendo estas da linha 23 à linha 29 do modelo PL apresentado na Figure 6.

3.2 Pergunta 2

```
1  /* Objective function */
2  min: tf ;
3
4  /* Variable bounds */
5  t1 >= t0 + 4;
6  t1 >= t3 + 9;
7  t3 >= t0 + 4;
8  t3 >= t5 + 6;
9  t6 >= t5 + 6;
10 t6 >= t8 + 8;
11 t9 >= t8 + 8;
12 t4 >= t6 + 4;
13 t7 >= t6 + 4;
14 t7 >= t9 + 7;
15 t2 >= t1 + 7;
16 t2 >= t4 + 4;
17 tf >= t2 + 2;
18 tf >= t7 + 2;
19 t0 >= ti + 0;
20 t5 >= ti + 0;
21 t8 >= ti + 0;
22
23 t0 + 4 <= t5 + 100 - 100*y05;
24 t5 + 6 <= t0 + 100*y05;
25
26 t0 + 4 <= t8 + 100 - 100*y08;
27 t8 + 8 <= t0 + 100*y08;
28
29 t5 + 6 <= t8 + 100 - 100*y58;
30 t8 + 8 <= t5 + 100*y58;
31
32
33 bin y05, y08, y58;
```

Figure 7: Ficheiro .lp com restrições e função objetivo

3.3 Pergunta 3

Variables	MILP Feasible	result
	28	28
tf	28	28
t1	19	19
t0	0	0
t3	10	10
t5	4,00000000...	4,000...
t6	18	18
t8	10	10
t9	19	19
t4	22	22
t7	26	26
t2	26	26
ti	0	0
y05	1	1
y08	1	1
y58	1	1

Figure 8: Output do ficheiro .lp da Figure 6

3.4 Pergunta 4

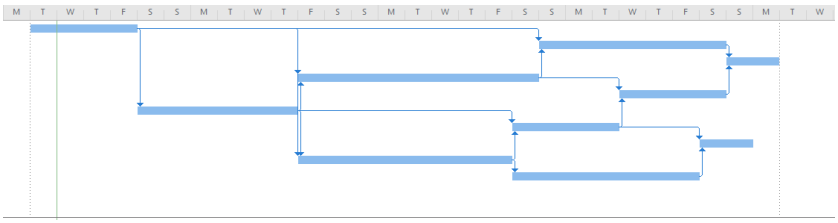

Figure 9: Diagrama de Gantt

4 Parte II

4.1 Pergunta 1

Nesta segunda parte, tinhamos que analisar as atividades nas quais iríamos aplicar uma redução do custo temporal, de forma a reduzir o custo temporal do projeto inteiro em 3 U.T. e minimizar o custo. Para tal, adicionámos novas restrições relativas ao limite de reduções de cada atividade e atualizamos as restrições do tempo de cada atividade consoante as reduções que nela se aplicariam. Desta forma, criou-se uma variável binária, yi, sendo 0 caso não se tenha obtido uma redução máxima na atividade i, e 1 caso contrário, podendo assim aplicar reduções extra a essa atividade. Contudo, tivemos dificuldade na implementação deste raciocínio em programação inteira. Apresentamos na mesma o nosso ficheiro input que traduz o nosso raciocínio, apesar de estar incompleto.

4.2 Pergunta 2

```
/* Objective function */
min: 200*r10 + 1000*r11 + 200*r12 + 800*r13 + 1600*r14 + 200*r16
             + 1000*r18 + 600*r19 + 100*r20 + 500*r21 + 100*r22
             + 400*r23 + 800*r24 + 100*r26 + 500*r28 + 300*r29 ;
```

Figure 10: Input

```
/* Variable bounds */
tf <= 21;

t1 >= t0 - r10 - r20 + 4;
t1 >= t3 - r13 - r23 + 9;
t3 >= t0 - r10 - r20 + 4;
t6 >= t5 + 6;
t3 >= t5 + 6;
t6 >= t8 - r18 - r28 + 8;
t9 >= t8 - r18 - r28 + 8;
t4 >= t6 - r16 - r26 + 4;
t7 >= t6 - r16 - r26 + 4;
t7 >= t9 - r19 - r29 + 7;
t2 >= t1 - r11 - r21 + 7;
t2 >= t4 - r14 - r24 + 4;
tf >= t2 - r12 - r22 + 2;
tf >= t7 + 2;
t0 >= ti + 0;
t5 >= ti + 0;
t8 >= ti + 0;
```

Figure 11: Input

```
r10 <= 0.5;
r20 <= 0.5*y0;
r11 <= 3;
r21 <= 2*y1;
r12 <= 0.5;
r22 <= 0.5*y2;
r13 <= 2;
r23 <= 1*y3;
r14 <= 0.5;
r24 <= 0.5*y4;
r16 <= 0.5;
r26 <= 0.5*y6;
r18 <= 0.5;
r28 <= 0.5*y8;
r19 <= 1;
r29 <= 1*y9;
```

Figure 12: Input

```
0.5 >= r10 - 100 + 100*y0;
0.5 <= r10 + 100*y0;
3 >= r11 - 100 + 100*y1;
3 <= r11 + 100*y1;
0.5 >= r12 - 100 + 100*y2;
0.5 <= r12 + 100*y2;
2 >= r13 - 100 + 100*y3;
2 <= r13 + 100*y3;
0.5 >= r14 - 100 + 100*y4;
0.5 <= r14 + 100*y4;
0.5 >= r16 - 100 + 100*y6;
0.5 <= r16 + 100*y6;
0.5 >= r18 - 100 + 100*y8;
0.5 <= r18 + 100*y8;
1 >= r19 - 100 + 100*y9;
1 <= r19 + 100*y9;

bin y0, y1, y2, y3 ,y4 ,y6 ,y8 , y9;
```

Figure 13: Input