Modelos Determinísticos de Investigação Operacional

Trabalho 1 - Grupo 19

William Sousa 61029; Jose Gonçalves 58657; João Carlos 53690; Axel Silva 53064; Andreia Martins 54759.

12 de Outubro de 2015

Conteúdo

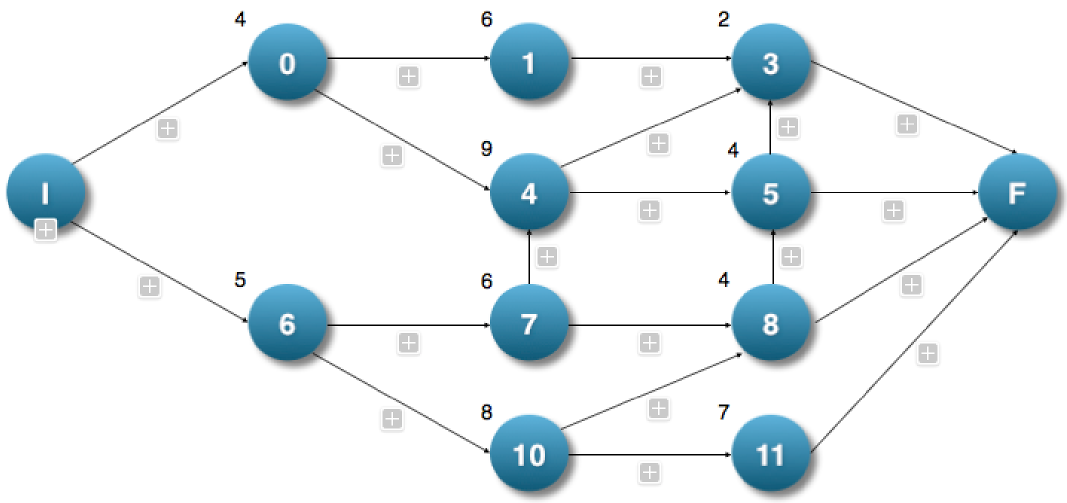
1 Introdução

Este relatório tem como objetivo a analise do primeiro trabalho da unidade curricular de Modelos Determinísticos de Investigação Operacional. Neste trabalho foi nos apresentado o problema do caminho mais longo, também conhecido de caminho critico, tendo como contexto as tarefas necessárias para a conclusão de um projeto fictício que vai ser o caso de estudo, as tarefas do projeto têm relações de precedência e durações determinísticas conhecidas formando assim um grafo acíclico dirigido que é uma das bases para todos os modelos aqui apresentados. Numa primeira parte iremos modelar e procurar uma solução para o problema apresentado tirando partido do principio de conservação de fluxo, no entanto está solução apenas nos fornece a informação do caminho critico e nada mais. Numa segunda parte é sugerido que passemos a tratar do problema tirando proveito da noção de precedência temporal entre as atividades e formando assim um modelo mais robusto que permite apurar e acomodar outros tipos de restrições, tais como o custo associado a cada atividade e uma possível redução no tempo de atividades especificas. Por fim esperamos conseguir não só responder e modelar o sistema de forma correta e precisa como acumular conhecimentos tanto ao nível operacional das ferramentas, nomeadamente do LPSolve, como ao nível científico que nos permita ganhar alguma sensibilidade sobre as dificuldades que estão associadas a modelação de sistemas complexos em problemas “simples” de programação linear.

2 Parte I

2.1 Questão 1

A rede correspondente a remoção das tarefas 2 e 9 do projeto é o seguinte:



2.2 Questão 2

O objetivo deste exercício é a determinação de um modelo em programação linear que determine o caminho mais longo, ou caminho crítico, da rede de um projeto. Seguindo o enunciado, as variáveis de decisão para este modelo serão descritas da forma $X_{[origem][destino]}$. cada uma associada a um arco do grafo e cada arco deverá tomar o valor de 1 unidade de fluxo, caso faça parte do caminho crítico, ou 0 unidade de fluxo caso contrário.

A função objetivo é constituída pelo somatório de todos os arcos, i.e. as variáveis de decisão, cada um multiplicado pela duração associada ao nodo de origem do arco. E uma vez que estamos à procura de um extremo condicional e queremos o caminho mais longo do projeto é natural que a escolha seja um máximo, portanto temos que a função objetivo é representada por:

```
max:4x01+4x04+6x13+2x3f++9x43+9x45+4x53+4x5f+5x67+5x610+6x78+
6x74+4x85+4x8f+8x108+8x1011+7x11f;
```

Na determinação das restrições, foi seguido o princípio da conservação de fluxo em um grafo acíclico dirigido. Começando por injetar uma unidade de fluxo no nodo ”início”, ação essa representada na restrição $x_{i0}+x_{i6} = 1$; as seguintes restrições obrigam a que o fluxo em todos os nodos seja igual a zero. É importante ressaltar que decidimos manter a última restrição que indica a saída de uma unidade de fluxo do grafo, apenas por uma questão estética e para manter a coerência sobre a preservação do fluxo, mas de facto conseguimos verificar após alguma analise do modelo e interações com o mesmo que essa restrição é redundante, uma vez que não há outro caminho possível para que a unidade de fluxo possa “prosseguir” após ter passado pelos nodos que representam o fim do projeto. Portanto uma vez que esse é um problema académico e tem dimensões e um numero de restrições que são computacionalmente "simples" decidimos, tal como referido anteriormente, manter a última restrição que representa a saída da unidade de fluxo e consequentemente o fim do projeto.

2.3 Questão 3

Input utilizado no LPSolve:

```
/* Objective function */
max:4x01+4x04+6x13+2x3f++9x43+9x45+4x53+4x5f+5x67+5x610+6x78+
6x74+4x85+4x8f+8x108+8x1011+7x11f;

/* Restrictions - Princípio da conservação de fluxo */

//início do fluxo
xi0+xi6=1;

//conservação do fluxo
-xi0+x01+x04=0; // nodo 0
-x01+x13=0; // nodo 1
-x13-x43-x53+x3f=0 ;// nodo 3
-x04-x74+x43+x45=0; // nodo 4
-x45-x85+x53+x5f=0; // nodo 5
-xi6+x67+x610=0; // nodo 6
-x67+x74+x78=0; // nodo 7
-x78-x108+x85+x8f=0;// nodo 8
-x610+x108+x1011=0; // nodo 10
-x1011+x11f=0; // nodo 11

// fim do fluxo
-x3f-x5f-x8f-x11f=-1;
```

2.4 Questão 4

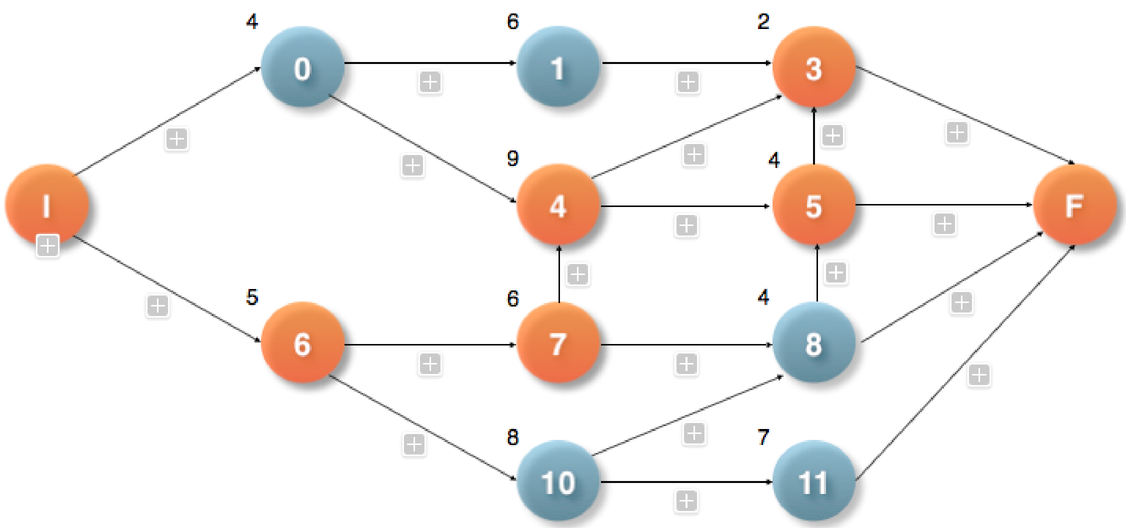
Output do LPSolve:

```
Variables        result
        26
x01        0
x04        0
x13        0
x3f        1
x43        0
x45        1
x53        1
x5f        0
x67        1
x610        0
x78        0
```

```
x74         1
x85         0
x8f         0
x108         0
x1011         0
x11f         0
xi0         0
xi6         1
```

2.5 Questão 5

A rede abaixo representa o caminho critico, i.e. os nodos preenchidos com a cor laranja, que consiste no caminho formado pelas tarefas 1, 6, 7, 4, 5 e 3.



3 Parte II

3.1 Questão 1

Para esta segunda parte o objetivo é encontrar o tempo mínimo de execução total do projeto através de um modelo em programação linear. As variáveis de decisão são descritas como $t_{[atividade]}$ e representam o tempo de início da atividade a que se referem.

A função objetiva é trivial pois apenas constituí a variável tf, que representa o tempo de início da atividade do nodo "fim", ou seja, a duração total do projeto, que deve ser minimizada:

```
min:tf;
```

As restrições do modelo devem especificar as precedências temporais a que o projeto está sujeito. Logo o tempo de início de uma dada atividade deverá ser maior ou igual do que a soma da duração da atividade anterior com o tempo de início desta, pois uma atividade que tenha uma precedência só pode iniciar no momento em que a anterior termina, aplicado essas restrições a todas as tarefas e as suas respetivas precedências o modelo fica formulado.

3.2 Questão 2

Input utilizado no LPSolve:

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

//* Restrictions - Precedências Temporais */
t0>=0;
t1>=4+t0;
t3>=6+t1;
t3>=9+t4;
t3>=4+t5;
t4>=4+t0;
t4>=6+t7;
t5>=9+t4;
t5>=4+t8;
t6>=0;
t7>=5+t6;
t8>=6+t7;
t8>=8+t10;
t10>=5+t6;
t11>=8+t10;
tf>=2+t3;
tf>=4+t5;
tf>=4+t8;
tf>=7+t11;
```

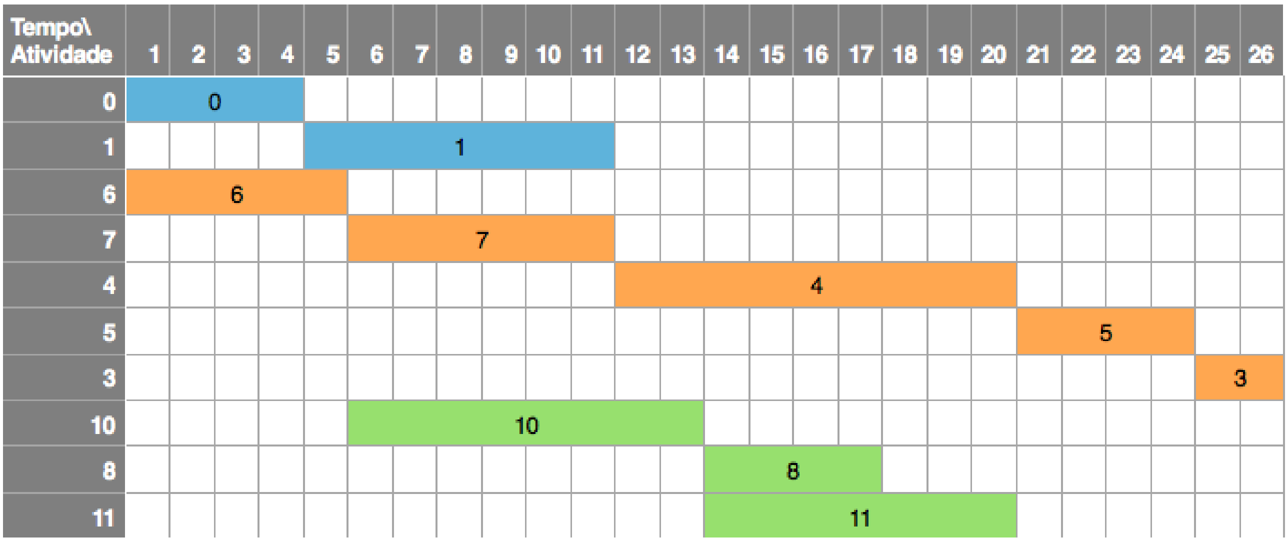
3.3 Questão 3

Ouput do LPSolve:

```
Variables        result
        26
tf        26
t0        0
t1        4
t3        24
t4        11
t5        20
t7        5
t8        13
t6        0
t10        5
t11        13
```

3.4 Questão 4

No seguinte diagrama de Gantt é possível analisar o comportamento do projeto face ao caminho critico, verificar o tempo mínimo necessário em que uma tarefa pode ser iniciada e como o atraso em uma tarefa pode ou não contribuir para um atraso no projeto.

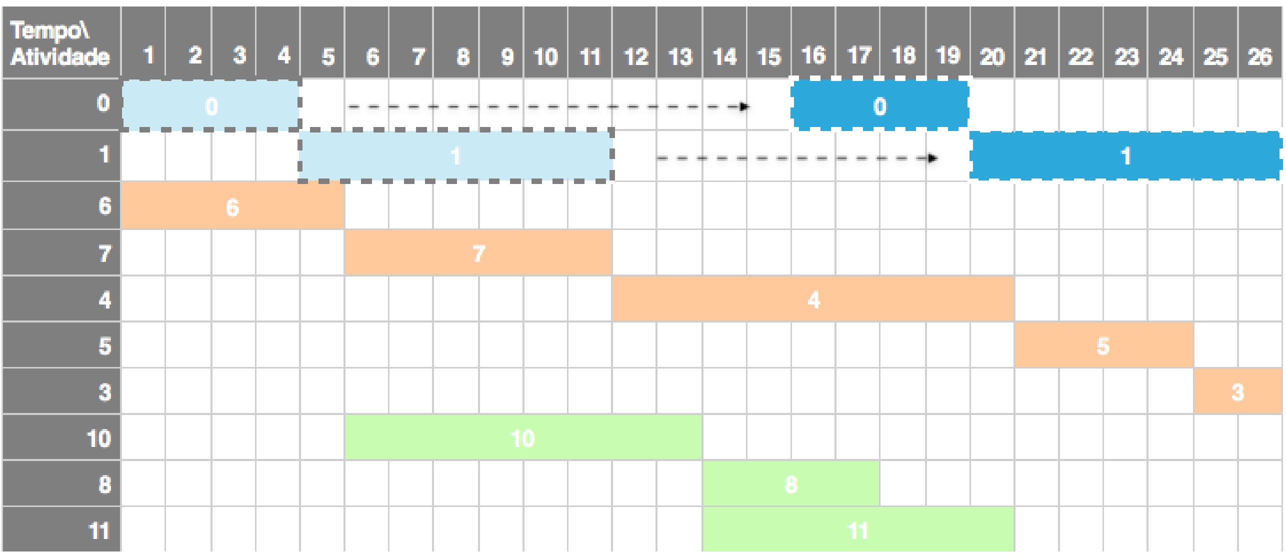


3.5 Questão 5

Com base no diagrama apresentado acima, e escolhendo a tarefa sete do caminho crítico, verifica-se que a mesma deverá ser iniciada 5 unidades de tempo após o início do projeto. Para além do diagrama, pode-se verificar o mesmo com base no valor da variável t_7 que toma o valor de 5 unidades de tempo no ficheiro de output do LPSolve.

3.6 Questão 6

Passemos agora a análise de uma tarefa que não faz parte do caminho crítico, por exemplo, tomemos como objeto de estudo a tarefa zero que não possui nenhuma precedência e faz parte da precedência apenas da tarefa um. A tarefa zero uma vez que não possui nenhuma precedência pode ser iniciado, no mínimo, no instante inicial do projeto; e como a tarefa zero tem uma duração de 4 unidades de tempo e a tarefa um uma duração de 6 unidades de tempo, a tarefa zero poderá ser iniciada, no máximo, no instante de tempo 15 sem causar atrasos no projeto que tem uma duração mínima de 26 unidades de tempo. Essa janela de tempo $[0, 15]$ é trivialmente verificada no diagrama de Gantt abaixo representado:



4 Parte III

4.1 Questão 1

Nesta parte é associado a cada atividade um custo monetário de execução, e a possibilidade de reduzir a duração de cada uma destas. No entanto, o número de reduções da duração de uma atividade tem um limite e um custo por unidade de tempo associado. O objetivo é reduzir o tempo de execução encontrado na Parte I (que também coincide com a Parte II) do projeto por 3 unidades de temporais, minimizando o custo suplementar causado pelas reduções. Como variáveis de decisão temos $r_{[atividade]}$ que representa o número de reduções efetuadas na duração da atividade correspondente. A função objetivo será constituída pelo somatório dos valores de custo normal de cada atividade, com o produto da redução e o custo suplementar correspondente. Esta função representa o custo total do projeto, cujo objetivo é minimizar, temos então que a função objetivo é:

```
min:400+100r0+1000+300r1+300+100r3+2000+400r4+1000+800r5+800+90r6+900+600+100r8+
1600+500r10+1400+300r11+r7+ri;
```

que pode ser simplificada como:

```
min: 10000+100r0+300r1+100r3+400r4+800r5+90r6+100r8+500r10+300r11+r7+ri;
```

Neste modelo a primeira restrição diz respeito a variável t_f e representa o tempo mínimo pretendido para o projeto que será menor ou igual a 23; as restantes restrições dizem respeito limite de reduções permitida em cada atividade; E por fim temos que o tempo de início de uma dada atividade deverá ser maior ou igual do que a duração da atividade anterior somada com o tempo de início desta e subtraída com o número de unidades temporais de redução da mesma. Este modelo é uma variação do modelo da Parte II, incluindo apenas as possíveis reduções da duração de cada atividade, mas é fácil verificar que dado a relações de precedência e ao conceito de caminho critico tais reduções iram afetar a duração do projeto de uma forma dinâmica.

4.2 Questão 2

Input do LPSolve:

```
/* Objective function */
min: 10000+100r0+300r1+100r3+400r4+800r5+90r6+100r8+500r10+300r11+r7+ri;

/* Restrictions - Precedências Temporais + Reduções (c/ Custo Associado) */

//Factor de Optimização
tf<=23;

//Limite superior das reduções
ri=0;
r7=0;
r0<=1;
r1<=2;
r3<=1;
r4<=3;
r5<=1;
```

```
r6<=2;
r8<=1;
r10<=1;
r11<=2;

//precedências temporais
t0>=0;
t1>=4-r0+t0;
t3>=6-r1+t1;
t3>=9-r4+t4;
t3>=4-r5+t5;
t4>=4-r0+t0;
t4>=6-r7+t7;
t5>=9-r4+t4;
t5>=4-r8+t8;
t6>=0;
t7>=5-r6+t6;
t8>=6-r7+t7;
t8>=8-r10+t10;
t10>=5-r6+t6;
t11>=8-r10+t10;
tf>=2-r3+t3;
tf>=4-r5+t5;
tf>=4-r8+t8;
tf>=7-r11+t11;
```

4.3 Questão 3

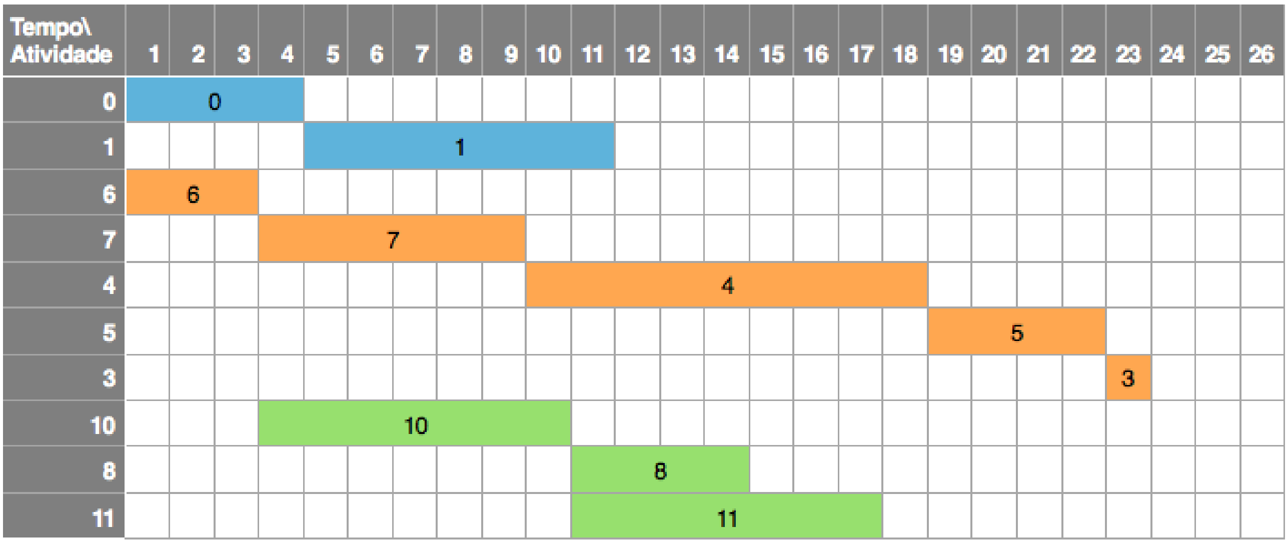
Output do LPSolve:

```
Variables        result
        10280
r0        0
r1        0
r3        1
r4        0
r5        0
r6        2
r8        0
r10        0
r11        0
r7        0
ri        0
tf        23
t0        0
t1        16
t3        22
t4        9
t5        18
```

```
t7         3
t8         11
t6         0
t10         3
t11         16
```

4.4 Questão 4

Resolvido o problema apresentado e obtido o output com auxilio do LPSolve, verifica-se que a duração da atividade "três"deve ser reduzida por uma unidade de tempo, com um custo associado de 100 U.M., e que a duração da atividade "seis"deve ser reduzida por 2 unidades de tempo, com um custo de 180 U.M.



Somando os valores dos custos normais de cada atividade com o custo das reduções, temos:

$$10000 + 100 * 1 + 90 * 2 = 10280 \;\; U.M.$$

Que é igual ao resultado da função objetivo apresentado pelo output do programa.

5 Parte V

5.1 Questão 1

O problema apresentado é uma variação daquele presente na parte III, mas com um conjunto maior de reduções e custos que resulta de aproximação linear de custo e reduções não-linear, neste caso, cada atividade tem um custo de redução c_1 com um número máximo de reduções associado; e um custo de redução c_2 e correspondente número máximo de reduções. Este último tipo de redução só pode ser aplicado se o número máximo de reduções c1 for atingido. De facto, para garantir essa restrição seria necessário ter algo como:

$$c_1 \geq c_2$$

No entanto, uma vez que os custos de reduções por unidade de tempo associados as reduções do tipo c_1 são sempre inferiores que os do tipo c_2, para todas as atividades do projeto, tal restrição não é necessária já que a função objetivo é minimizante. E caso tivéssemos custos de reduções com um "pior" comportamento seria necessário utilizar tal restrição para manter a mesma lógica de negocio em que apenas se utiliza as reduções do tipo c_2 após ter se esgotado as reduções do tipo c_1. Além das variáveis que representam o tempo de início de uma atividade usadas tal como anteriormente variáveis descritas como:

$$r_{[atividade][tipo]}$$

Que representam o número de reduções de cada atividade é o tipo correspondente. O número máximo de reduções das atividades no projeto é de 4 unidades temporais tendo em conta obviamente o custo suplementar de cada redução. A função objetivo será semelhante á usada em III, com o somatório dos valores de custo normal de cada atividade e o produto de cada variável $r_{[atividade][tipo]}$ com o custo suplementar correspondente. Mais uma vez, esta função representa o custo a minimizar:

```
min:400+100r01+200r02+1000+300r11+600r12+300+100r31+200r32+2000+400r41+
800r42+1000+800r51+1600r52+800+90r61+180r62+900+600+100r81+200r82+1600+
500r101+1000r102+1400+300r111+600r112;
```

Em relação ao modelo da parte III, as restrições foram alteradas de forma a acomodar a nova representação das reduções. Cada variável $r_{[atividade][tipo]}$ é limitada por o seu respetivo número máximo de reduções, e na representação das precedências temporais subtrai-se o número de reduções possíveis de cada tipo ao tempo de início da atividade precedente somado com a duração da atividade precedente. Por fim, estará presente que $t_f \leqslant 22$, representando a redução de 4 unidades temporais á duração do projeto. E de notar que de forma sistemática poderiam ser utilizados aproximações mais precisas, aumentado o número de variáveis de decisão, para a função não-linear que representa o custo da redução, bastando apenas repetir o processo apresentado aqui em relação ao modelo base (apresentado na Parte III).

5.2 Questão 2

```
/*Objective function */
min:400+100r01+200r02+1000+300r11+600r12+300+100r31+200r32+2000+400r41+
800r42+1000+800r51+1600r52+800+90r61+180r62+900+600+100r81+200r82+1600+
500r101+1000r102+1400+300r111+600r112;

/* Restrictions - precedências Temporais + Reduções (c/ Custo Associado) */

//factor de optimização
tf<=22;

//Limite superior das reduções
ri=0;
r71=0;
r72=0;

r01<=0.5;
r11<=1;
r31<=0.5;
r41<=1;
r51<=0.5;
r61<=1;
r81<=0.5;
r101<=0.5;
r111<=1;
```

```
r02<=0.5;
r12<=1;
r32<=0.5;
r42<=2;
r52<=0.5;
r62<=1;
r82<=0.5;
r102<=0.5;
r112<=1;

//precedências temporais
t0>=0;
t1>=4-r01-r02+t0;
t3>=6-r11-r12+t1;
t3>=9-r41-r42+t4;
t3>=4-r51-r52+t5;
t4>=4-r01-r02+t0;
t4>=6-r71-r72+t7;
t5>=9-r41-r42+t4;
t5>=4-r81-r82+t8;
t6>=0;
t7>=5-r61-r62+t6;
t8>=6-r71-r72+t7;
t8>=8-r101-r102+t10;
t10>=5-r61-r62+t6;
t11>=8-r101-r102+t10;
tf>=2-r31-r32+t3;
tf>=4-r51-r52+t5;
tf>=4-r81-r82+t8;
tf>=7-r111-r112+t11;
```

5.3 Questão 3

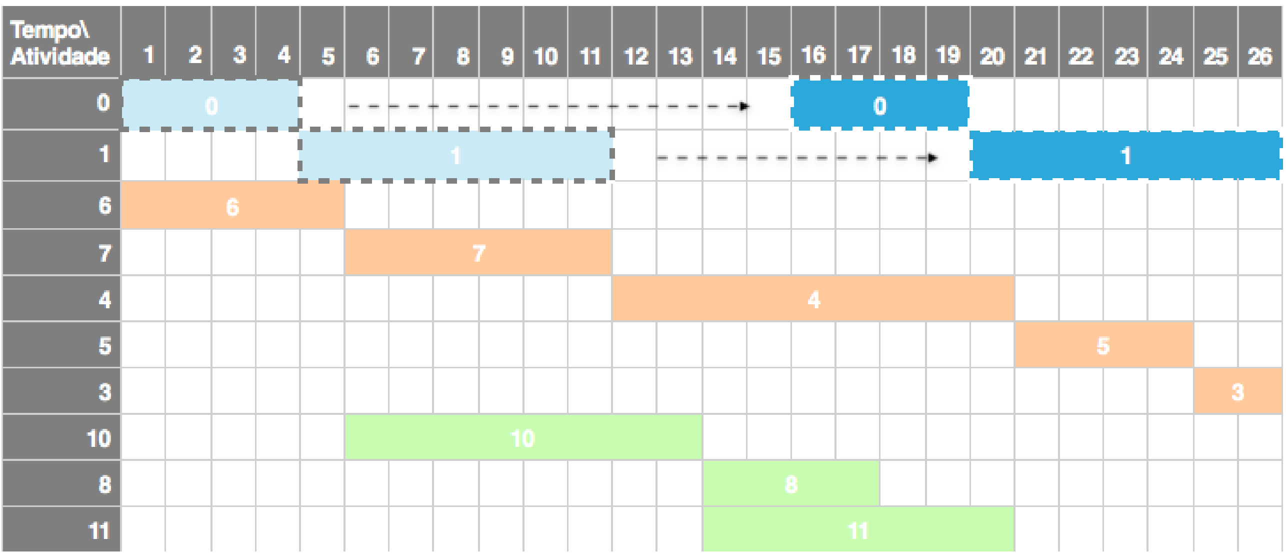
Output do LPSolve:

```
Variables        result
        10820
r01        0
r02        0
r11        0
r12        0
r31        0,5
r32        0,5
r41        1
r42        0
r51        0
r52        0
r61        1
r62        1
```

```
r81         0
r82         0
r101         0
r102         0
r111         0
r112         0
tf          22
ri          0
r71          0
r72          0
t0          0
t1          4
t3          21
t4          9
t5          17
t7          3
t8          11
t6          0
t10          3
t11          11
```

5.4 Questão 4

Resolvido o problema apresentado e obtido o output com auxílio do LPSolve, verifica-se que a duração da atividade "três"deve ser reduzida por 0.5 unidades de tempo a custo c_1 e 0.5 unidades de tempo a custo c_2, com um custo total de 150 U.M., a duração da atividade "quatro"deve ser reduzida por 1 unidade de tempo a custo c_1, com um custo total 400, e a duração da atividade "seis"deve ser reduzida por 1 unidade de tempo a custo c_1 e 1 unidade de tempo a custo c_2, com um custo total de 270 U.M.



Somando os valores dos custos normais de cada atividade com o custo das reduções, temos

$$10000 + 100 * 0.5 + 200 * 0.5 + 400 * 1 + 90 * 1 + 180 * 1 = 10820 \ \ U.M.$$

Que é igual ao resultado da função objetivo apresentado pelo output do programa.