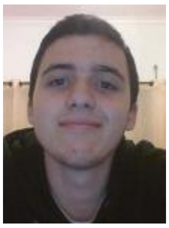
Universidade do Minho
Escola de Engenharia
Departamento de Informática
Mestrado Integrado em Engenharia Informática

Modelos Determinísticos de Investigação Operacional

Trabalho Prático Nº. 3

Braga, 24 de Novembro de 2015

Pedro Vieira Fortes
Nº64309



João Pedro Pereira Fontes
Nº71184

Resumo

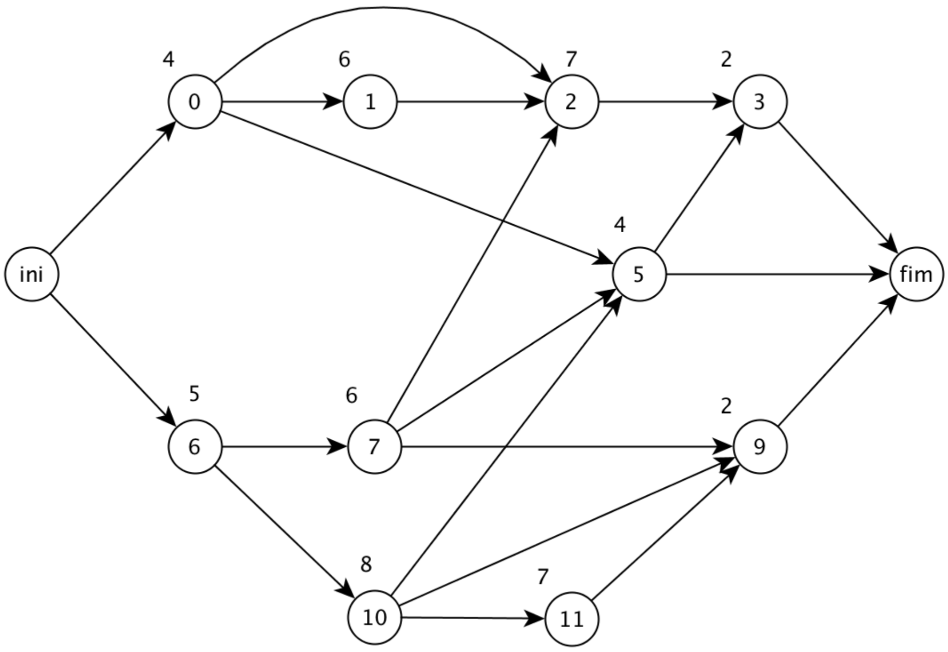
Este relatório tem como intuito apresentar a análise e resolução do trabalho que nos foi proposto na unidade curricular de Modelos Determinísticos de Investigação Operacional pelo professor Valério Carvalho. Neste trabalho deveríamos resolver um caso de gestão de projetos com a ajuda da ferramenta LPSolve e aplicando os conhecimentos adquiridos na unidade curricular.

Índice

Parte 1

1.1. Formulação do Problema

Através das instruções para a determinação da nossa rede em estudo, obtemos o seguinte resultado:



Esta rede é resultante da rede original do enunciado retirando o nodo 4 e 8 que corresponde aos últimos 2 dígitos do aluno com maior número de inscrição (71184).

Em tudo, o objetivo deste modelo é igual ao adotado na Parte II do Trabalho 1, ou seja, minimizar o tempo de execução total do projeto através da imposição de restrições nos tempos de início de cada atividade. A única diferença em relação à Parte II do trabalho 1 é que escolhidas 3 atividades que fossem possíveis correr em paralelo, estas fossem executadas em série. Observando o diagrama de Gantt da parte II do Trabalho 1 escolhemos as atividades 1, 7 e 10 para a experiência.

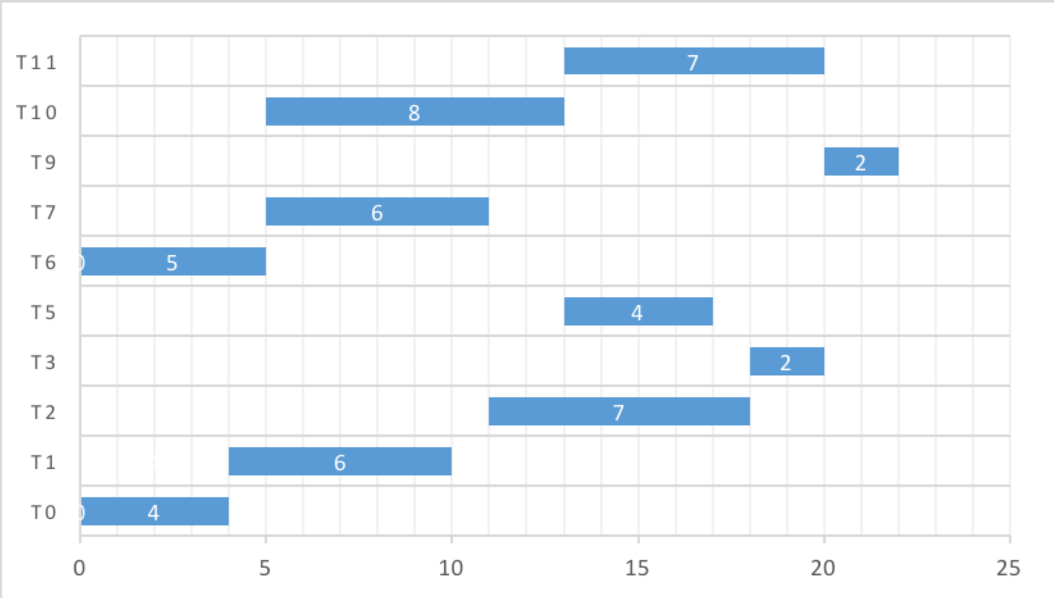


Figura 1- Diagrama de Gantt da parte II do 1º Trabalho

Com estas formalizações temos o problema formulado da seguinte forma:

- **Variáveis de Decisão**

  As nossas variáveis de decisão são:
  - ✓ **ti** (variáveis inteiras), no qual representam o tempo acumulado até o nodo i (tal que i ϵ {0,1,2,3,5,6,7,9,10,11,f}). Assim, ti representa o tempo mínimo decorrido até se poder iniciar a atividade do nodo i em causa.
  - ✓ **y** (variável binária), no qual indica a ativação ou não das restrições de serialização (que serão explicadas no tópico das restrições desta parte).

- **Função Objetivo**

  Tem-se então a seguinte função objetivo:

$$min: tf;$$

  O tf representa o tempo total de execução das atividades pertencentes ao caminho crítico, respeitando as suas devidas precedências.

- **Restrições**

  O problema em questão exigia restrições que contemplam para cada ti (tal que i ϵ {0,1,2,3,4,5,6,7,8,9,10,11,f}) os tempos ti dos nodos precedentes, mais o tempo de execução desse mesmos nodos i. Já t0 e t6 são inicializados a 0, pois são os nodos adjacentes ao nodo inicial, logo podem ser todos iniciados no instante de tempo 0 representando a inicialização do projeto.

$$
\begin{aligned}
t0 &= 0;\\
t6 &= 0;\\
t1 &>= t0 + 4;\\
t2 &>= t0 + 4;\\
t2 &>= t1 + 6;\\
t2 &>= t7 + 6;\\
t3 &>= t2 + 7;\\
t3 &>= t5 + 4;\\
t5 &>= t0 + 4;\\
t5 &>= t7 + 6;\\
t5 &>= t10 + 8;\\
t7 &>= t6 + 5;\\
t9 &>= t7 + 6;\\
t9 &>= t10 + 8;\\
t9 &>= t11 + 7;\\
t10 &>= t6 + 5;\\
t11 &>= t10 + 8;\\
tf &>= t3 + 2;\\
tf &>= t5 + 4;\\
tf &>= t9 + 2;
\end{aligned}
$$

As restrições novas adicionadas em relação à parte II do 1º trabalho são as responsáveis por serializar as tarefas 1, 7 e 10. Pelo o diagrama de Gantt mostrado, é fácil identificar que a tarefa t1 garantidamente será a primeira a ser executada, logo as outra duas tarefas terão que ser executadas só depois de t1 terminar (F1 e F2). Para a decisão entre as outras duas tarefas, fazendo uso de uma variável binária, esta tomará o valor 1 se a tarefa 10 for escolhida para ser realizada primeiro que a tarefa 7 ou então tomará o valor 0 seja a tarefa 5 a escolhida para ser realizada em primeiro. Estas restrições (F3 e F4) são possíveis através de uma implicação lógica que só pode ser introduzida com programação linear inteira, ou seja, se a variável binária y tomar o valor 1 então a tarefa 10 é efetuada antes da tarefa 7, caso contrário a tarefa 5 será executada antes da tarefa 2.

$$
\begin{aligned}
&t7 >= t1 + 6; && \text{F1}\\
&t10 >= t1 + 6; && \text{F2}\\
&t10 + 8 <= t7 + 50 - 50y; && \text{F3}\\
&t7 + 6 <= t10 + 50y; && \text{F4}
\end{aligned}
$$

Não adicionamos restrições de não-negatividade porque o LPSOLVE assume implicitamente que as variáveis não são negativas.

1.2. Input no LP Solve

Este foi o input realizado no programa LP Solve, no qual trata de todos os aspetos falados na formulação do problema, incluindo alguns comentários colocados no input:

```
/* Objective function */

min: tf;

/* Variable bounds */

t0 = 0;
t6 = 0;

t1 >= t0 + 4;

t2 >= t0 + 4;
t2 >= t1 + 6;
t2 >= t7 + 6;

t3 >= t2 + 7;
t3 >= t5 + 4;

t5 >= t0 + 4;
t5 >= t7 + 6;
t5 >= t10 + 8;

t7 >= t6 + 5;

t9 >= t7 + 6;
t9 >= t10 + 8;
t9 >= t11 + 7;

t10 >= t6 + 5;

t11 >= t10 + 8;

tf >= t3 + 2;
tf >= t5 + 4;
tf >= t9 + 2;

t1 + 6 <= t7;
t1 + 6 <= t10;

t7 >= t1 + 6;
t10 >= t1 + 6;

t10 + 8 <= t7 + 50 - 50y;
t7 + 6 <= t10 + 50y;

Int t0, t1, t2, t3, t5, t6, t7, t9, t10, t11, tf;
Bin y;
```

1.3. Output do LP Solve

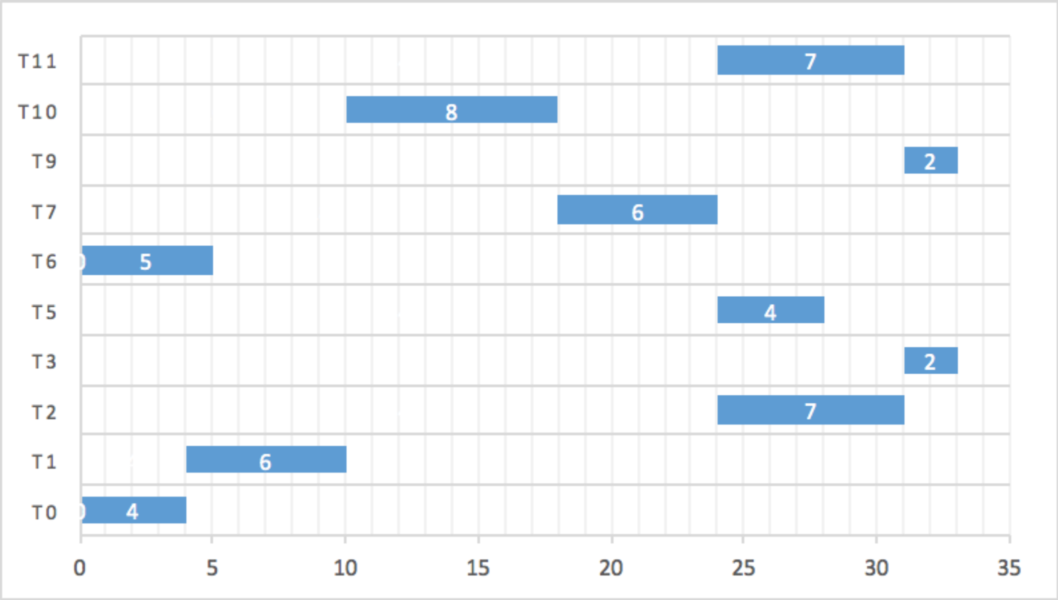
Este foi o output resultante do input inserido no LP Solve:

Variables	MILP ...	result
	33	33
t0	0	0
t1	4	4
t10	10	10
t11	24	24
t2	24	24
t3	31	31
t5	24	24
t6	0	0
t7	18	18
t9	31	31
tf	33	33
y	1	1

Como podemos ver, foi possível obter um output do programa, ou seja, o problema é possível tendo uma solução ótima de 33 U.T.

1.4. Diagrama de Gantt

Com ajuda do output obtido no LP Solve foi possível realizar um plano de execução do projeto através do seguinte diagrama de Gantt:



Cada linha representa uma atividade ti (tal que i ϵ {0,1,2,3,4,5,6,7,8,9,10,11,f}) que é iniciada só quando todas as atividades precedentes ou atividades prioritárias a estas acabam. As últimas tarefas acabam às 33 U.T. o que comprova que a solução ótima é de 33 U.T.

1.5. Observações Finais

Comparando estes resultados aos resultados da parte II do 1º Trabalho, podemos concluir que houve um aumento da duração total do projeto de 11 U.T. (33 - 22) sendo a solução ótima igual a 33 U.T. (tf = 33). Como se pode observar no diagrama de Gantt, isto ocorreu devido ao facto das atividades escolhidas (1, 7 e 10) deixarem de ser realizadas em paralelo, passando a serem serializadas o que implica um atraso nas atividades posteriores a estas. Neste caso, a atividade 1 é realizada em primeiro, logo depois desta terminar a tarefa 10 é realizada e por fim a atividade 7 é realizada.

Parte 2

2.1. Formulação do Problema

O objetivo deste modelo é minimizar o tempo global de execução do projeto na mesma situação que na parte 1 deste trabalho, mas desta vez com a oportunidade de realizar uma das tarefas em paralelo com uma empresa externa. No entanto essa escolha implica que a atividade escolhida tenha uma duração adicional de 1 U.T. As outras duas atividades não poderão ser realizadas em paralelo.

Com estas formalizações temos o problema formulado da seguinte forma:

- **Variáveis de Decisão**

  As nossas variáveis de decisão são:
  - ✓ **ti** (variáveis inteiras), no qual representam o tempo acumulado até o nodo i (tal que i ϵ {0,1,2,3,5,6,7,9,10,11,f}). Assim, ti representa o tempo mínimo decorrido até se poder iniciar a atividade do nodo i em causa.
  - ✓ **Yi_j** (variáveis binárias), no qual indica a ativação ou não das restrições de serialização (que serão explicadas no tópico de restrição desta parte), tal que i ,j ϵ {1,7,10})
  - ✓ **txi** (variáveis binárias), no qual indica a escolha ou não da atividade que será realizada por uma empresa exterior (tal que i ϵ {1,7,10}).

- **Função Objetivo**

  Tem-se então a seguinte função objetivo:

$$min: tf;$$

  O tf representa o tempo total de execução das atividades pertencentes ao caminho crítico, respeitando as suas devidas precedências.

- **Restrições**

  A restrição abaixo é a decisão de qual das 3 atividades será realizada por uma empresa exterior. As variáveis binárias txi tomam o valor 1 se a atividade do respetivo índice j = (1,7,10) for realizada externamente e o valor 0 caso contrário.

$$tx1 + tx7 + tx10 <= 1;$$

  Cada par de restrições abaixo representam as restrições não-simultaneidade, ou seja, quando uma restrição é ativada a outra será desativada. As variáveis binárias yi_j correspondem a cada par de restrições tal que toma o valor 1 se a atividade do respetivo índice i = (1,7,10) for realizada antes da atividade do respetivo índice j =(0,7,10); Toma o valor 0 caso contrário.

$$t1 + 6 <= t7 + 50 - 50x + 50\,tx7;$$
$$t7 + 6 <= t1 + 50x + 50tx1;$$

$$t1 + 6 <= t10 + 50 - 50y + 50tx10;$$
$$t10 + 8 <= t1 + 50y + 50tx1;$$

$$t7 + 6 <= t10 + 50 - 50z + 50tx10;$$
$$t10 + 6 <= t7 + 50z + 50\,tx7;$$

As restrições abaixo são as restrições de atribuição do tempo de execução das tarefas apresentadas na parte II do Trabalho 1 com a seguinte alteração: todas as atividades que possam ser realizadas por uma empresa externa, têm a adição de 1 U.T. caso estas sejam concretizadas por essa empresa exterior.

$$t0 = 0;$$
$$t6 = 0;$$

$$t1 >= t0 + 4;$$

$$t2 >= t0 + 4;$$
$$t2 >= t1 + 6 + tx1;$$
$$t2 >= t7 + 6 + tx7;$$

$$t3 >= t2 + 7;$$
$$t3 >= t5 + 4;$$

$$t5 >= t0 + 4;$$
$$t5 >= t7 + 6 + tx7;$$
$$t5 >= t10 + 8 + tx10;$$

$$t7 >= t6 + 5;$$

$$t9 >= t7 + 6 + tx7;$$
$$t9 >= t10 + 8 + tx10;$$
$$t9 >= t11 + 7;$$

$$t10 >= t6 + 5;$$

$$t11 >= t10 + 8 + tx10;$$

$$tf >= t3 + 2;$$
$$tf >= t5 + 4;$$
$$tf >= t9 + 2;$$

Não adicionamos restrições de não-negatividade porque o LPSOLVE assume implicitamente que as variáveis não são negativas.

2.2. Input no LP Solve

Este foi o input realizado no programa LP Solve, no qual trata de todos os aspetos falados na formulação do problema, incluindo alguns comentários colocados no input:

```
/* Objective function */

min: tf;

/* Variable bounds */

t0 = 0;
t6 = 0;

t1 >= t0 + 4;

t2 >= t0 + 4;
t2 >= t1 + 6 + tx1;
t2 >= t7 + 6 + tx7;

t3 >= t2 + 7;
t3 >= t5 + 4;

t5 >= t0 + 4;
t5 >= t7 + 6 + tx7;
t5 >= t10 + 8 + tx10;

t7 >= t6 + 5;

t9 >= t7 + 6 + tx7;
t9 >= t10 + 8 + tx10;
t9 >= t11 + 7;

t10 >= t6 + 5;

t11 >= t10 + 8 + tx10;

tf >= t3 + 2;
tf >= t5 + 4;
tf >= t9 + 2;

t1 + 6 <= t7 + 50 - 50y1_7 + 50 tx7;
t7 + 6 <= t1 + 50y1_7 + 50tx1;

t1 + 6 <= t10 + 50 - 50y1_10 + 50tx10;
t10 + 8 <= t1 + 50y1_10 + 50tx1;

t7 + 6 <= t10 + 50 - 50y7_10 + 50tx10;
t10 + 6 <= t7 + 50y7_10 + 50 tx7;
```

```
tx1 + tx7 + tx10 <= 1;

Int t0, t1, t2, t3, t5, t6, t7, t9, t10, t11, tf;
Bin y1_7, y1_10, y7_10, tx1, tx7, tx10;
```

2.3. Output do LP Solve

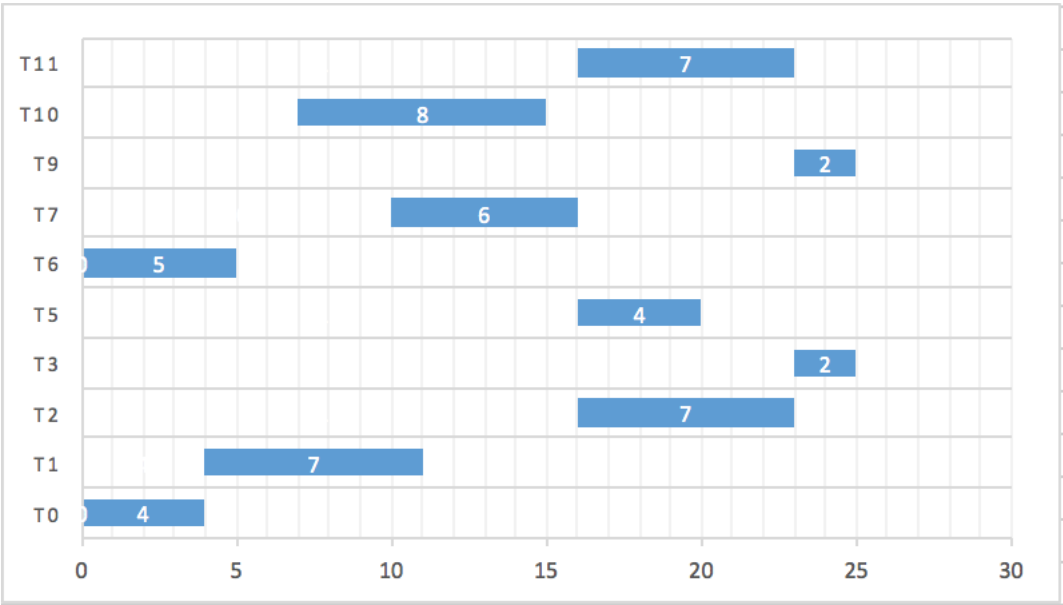
Este foi o output resultante do input inserido no LPSolve:

Variables	MILP ...	MILP ...	MILP ...	result
	28	26	25	25
t0	0	0	0	0
t1	4	4	4	4
t10	11	5	7	7
t11	19	17	16	16
t2	12	17	16	16
t3	23	24	23	23
t5	19	17	16	16
t6	0	0	0	0
t7	5	11	10	10
t9	26	24	23	23
tf	28	26	25	25
tx1	1	1	0	0
tx10	0	0	1	1
tx7	0	0	0	0
y1_10	0	0	1	1
y1_7	0	0	1	1
y7_10	1	0	1	1

Como podemos ver, foi possível obter um output do programa, ou seja, o problema é possível tendo uma solução ótima de 25 U.T.

2.4. Diagrama de Gantt

Com ajuda do output obtido no LP Solve foi possível realizar um plano de execução do projeto através do seguinte diagrama de Gantt:



Cada linha representa uma atividade ti (tal que i ϵ {0,1,2,3,4,5,6,7,8,9,10,11,f}) que é iniciada só quando todas as atividades precedentes ou atividades prioritárias a estas acabam. As últimas tarefas acabam às 25 U.T. o que comprova que a solução ótima é de 25 U.T.

2.5. Observações Finais

Comparando estes resultados aos resultados da parte II do 1º Trabalho, podemos concluir que houve um aumento da duração total do projeto de 3 U.T. (25 - 22) sendo a solução ótima igual a 25 U.T. (tf = 25). Como se pode observar no diagrama de Gantt, ao contrário do que aconteceu na parte I deste trabalho, desta vez é possível executar uma das atividades escolhidas em paralelo fora da empresa com uma penalização de 1 U.T. Apesar desta penalização, em relação à parte 1, esta utilização da empresa externa permite melhorar a nossa solução ótima, sendo a atividade 10 a atividade escolhida para ser realizada externamente. Entre as outras atividades que foram realizadas internamente na empresa, a atividade 1 foi a 1ª a ser executado e depois foi a atividade 7.

Parte 3

3.1. Formulação do Problema

O objetivo deste modelo é minimizar o custo suplementar, acrescentando reduções nos nodos, sendo que em cada nodo estão associados dois tempos máximos de redução, cada um com diferentes custos, sendo esses custos c1 e C2. É de notar que a redução com custo c2 só pode ser executada após a redução máxima do custo c1 ser aplicada. A diferença em relação à parte V do 1º Trabalho é que com as restrições aplicadas no mesmo (exemplo abaixo), não precisávamos de preocupar com a ordem com que as reduções eram aplicadas, pois as reduções com custo C1 eram sempre aplicada em 1º que as reduções com custo C2, pois C1 era sempre menor que C2.

$ti >= tj + 5 - rj\_1 - rj\_2;$ → este é um exemplo das restrições aplicadas na parte V do 1º Trabalho com os custos associados (tal que i ϵ {0,1,2,3,5,6,7,9,10,11,f})

Já neste caso, temos situações em que C2 é menor que C1, o que implica controlo das escolhas dos custos pela ordem correta.

Com estas formalizações temos o problema formulado da seguinte forma:

- **Variáveis de Decisão**

  As nossas variáveis de decisão são:
  - ✓ **ti**, no qual representam o tempo acumulado até o nodo i (tal que i ϵ {0,1,2,3,5,6,7,9,10,11,f}). Assim, ti representa o tempo mínimo decorrido até se poder iniciar a atividade do nodo i em causa.
  - ✓ **ri_j,** no qual representam a redução efetuada pelo custo associado j (tal que j ϵ {1,2}) na atividade i (tal que i ϵ {0,1,2,3,5,6,7,9,10,11,f}).
  - ✓ **yi** (variável binária), no qual indica se a redução associada ao custo 1 da atividade i foi totalmente utilizada (yi = 1) ou não (yi = 0) (tal que i ϵ {0,1,2,3,5,6,10,11}).

- **Função Objetivo**

  Tem-se então a seguinte função objetivo:

$$\begin{aligned} min: 400 &+ 200\,r0\_1 + 100\,r0\_2 + 1000 + 600\,r1\_1 + 300\,r1\_2 \\ &+ 1400 + 1000\,r2\_1 + 500\,r2\_2 + 300 + 200\,r3\_1 \\ &+ 100\,r3\_2 + 1000 + 1600\,r5\_1 + 800\,r5\_2 + 800 \\ &+ 180\,r6\_1 + 90\,r6\_2 + 900 + 300 + 1600 \\ &+ 1000\,r10\_1 + 500\,r10\_2 + 1400 + 600\,r11\_1 \\ &+ 300\,r11\_2; \end{aligned}$$

O min representa o custo mínimo total obtido derivado das reduções feitas para cumprir com as restrições dadas, ou seja, temos que determinar quais as reduções mais rentáveis de modo a diminuir as 4 unidades de tempo na execução total do projeto obtido na parte 2 do 1º Trabalho. Na função objetivo, todas as parcelas que não têm nenhum coeficiente ri_j associado são os custos constantes ou normais de se realizar uma determinada atividade (para este caso são as atividade 7 e 9). Os coeficientes dos ri_ j são o custo suplementar de se reduzir por U.T. no nodo i com o custo j.

- **Restrições**

As restrições abaixo representam as reduções máximas permitidas para cada variável ri_j, (tal que j ϵ {1,2} e i ϵ {0,1,2,3,5,6,10,11}). As reduções associadas ao custo 2 só serão ativadas se as reduções associadas ao custo 1 atingirem o seu máximo, ou seja, se a variável binária yi é ativada.

$$r0\_1 \ <= \ 0.5;$$
$$r0\_2 \ <= \ 0.5y0;$$

$$r1_1 <= \ 1;$$
$$r1\_2 \ <= \ 1y1;$$

$$r2\_1 \ <= \ 3;$$
$$r2\_2 \ <= \ 1y2;$$

$$r3\_1 \ <= \ 0.5;$$
$$r3\_2 \ <= \ 0.5y3;$$

$$r5\_1 \ <= \ 0.5;$$
$$r5\_2 \ <= \ 0.5y5;$$

$$r6\_1 \ <= \ 1;$$
$$r6\_2 \ <= \ 1y6;$$

$$r10\_1 \ <= \ 0.5;$$
$$r10\_2 \ <= \ 0.5y10;$$

$$r11\_1 \ <= \ 1;$$
$$r11\_2 \ <= \ 1y11;$$

As restrições abaixo determinam se é necessário usar totalmente as reduções associadas ao custo 1 (yi = 1) ou não (yi=0)

$$r0\_1 \ >= \ 0.5\,y0;$$
$$r1\_1 \ >= \ 1\,y1;$$
$$r2\_1 \ >= \ 3y2;$$
$$r3\_1 \ >= \ 0.5y3;$$
$$r5\_1 \ >= \ 0.5y5;$$
$$r6\_1 \ >= \ 1y6;$$
$$r10\_1 \ >= \ 0.5y10;$$

$$r11\_1 >= 1y11;$$

As restrições abaixo são as restrições de atribuição do tempo de execução das tarefas apresentadas na parte II do Trabalho 1 com a possibilidade de poder reduzir o tempo de duração de algumas das atividades com as variáveis de redução, segunda a tabela apresentada no enunciado.

$$t0 = 0;$$
$$t6 = 0;$$
$$t1 >= t0 + 4 - r0\_1 - r0\_2;$$
$$t2 >= t0 + 4 - r0\_1 - r0\_2;$$
$$t2 >= t1 + 6 - r1\_1 - r1\_2;$$
$$t2 >= t7 + 6;$$
$$t3 >= t2 + 7 - r2\_1 - r2\_2;$$
$$t3 >= t5 + 4 - r5\_1 - r5\_2;$$
$$t5 >= t0 + 4 - r0\_1 - r0\_2;$$
$$t5 >= t7 + 6;$$
$$t5 >= t10 + 8 - r10\_1 - r10\_2;$$
$$t7 >= t6 + 5 - r6\_1 - r6\_2;$$
$$t9 >= t7 + 6;$$
$$t9 >= t10 + 8 - r10\_1 - r10\_2;$$
$$t9 >= t11 + 7 - r11\_1 - r11\_2;$$
$$t10 >= t6 + 5 - r6\_1 - r6\_2;$$
$$t11 >= t10 + 8 - r10\_1 - r10\_2;$$
$$tf >= t3 + 2 - r3\_1 - r3\_2;$$
$$tf >= t5 + 4 - r5\_1 - r5\_2;$$
$$tf >= t9 + 2;$$

As restrições abaixo representam o tempo máximo pedido para que o projeto acabe.

$$tf = 22 - y;$$
$$y = 4;$$

Não adicionamos restrições de não-negatividade porque o LPSOLVE assume implicitamente que as variáveis não são negativas.

3.2. Input no LP Solve

Este foi o input realizado no programa LP Solve, no qual trata de todos os aspetos falados na formulação do problema, incluindo alguns comentários colocados no input:

```
/* Objective function */
min: 400 + 200 r0_1 + 100 r0_2
    + 1000 + 600 r1_1 + 300 r1_2
    + 1400 + 1000 r2_1 + 500 r2_2
```

```
    + 300 + 200 r3_1 + 100 r3_2
    + 1000 + 1600 r5_1 + 800 r5_2
    + 800 + 180 r6_1 + 90 r6_2
    + 900 + 300
    + 1600 + 1000 r10_1 + 500 r10_2
    + 1400 + 600 r11_1 + 300 r11_2;

/* Variable bounds */

r0_1 <= 0.5;
r0_1 >= 0.5 y0;
r0_2 <= 0.5y0;

r1_1 <= 1;
r1_1 >= 1 y1;
r1_2 <= 1y1;

r2_1 <= 3;
r2_1 >= 3y2;
r2_2 <= 1y2;

r3_1 <= 0.5;
r3_1 >= 0.5y3;
r3_2 <= 0.5y3;

r5_1 <= 0.5;
r5_1 >= 0.5y5;
r5_2 <= 0.5y5;

r6_1 <= 1;
r6_1 >= 1y6;
r6_2 <= 1y6;

r10_1 <= 0.5;
r10_1 >= 0.5y10;
r10_2 <= 0.5y10;

r11_1 <= 1;
r11_1 >= 1y11;
r11_2 <= 1y11;

t0 = 0;
t6 = 0;

t1 >= t0 + 4 - r0_1 - r0_2;

t2 >= t0 + 4 - r0_1 - r0_2;
t2 >= t1 + 6 - r1_1 - r1_2;
t2 >= t7 + 6;

t3 >= t2 + 7 - r2_1 - r2_2;
```

```
t3 >= t5 + 4 - r5_1 - r5_2;

t5 >= t0 + 4 - r0_1 - r0_2;
t5 >= t7 + 6;
t5 >= t10 + 8 - r10_1 - r10_2;

t7 >= t6 + 5 - r6_1 - r6_2;

t9 >= t7 + 6;
t9 >= t10 + 8 - r10_1 - r10_2;
t9 >= t11 + 7 - r11_1 - r11_2;

t10 >= t6 + 5 - r6_1 - r6_2;

t11 >= t10 + 8 - r10_1 - r10_2;

tf >= t3 + 2 - r3_1 - r3_2;
tf >= t5 + 4 - r5_1 - r5_2;
tf >= t9 + 2;

tf = 22 - y;
y = 4;

Bin y0, y1, y2, y3, y5, y6, y10, y11;
```

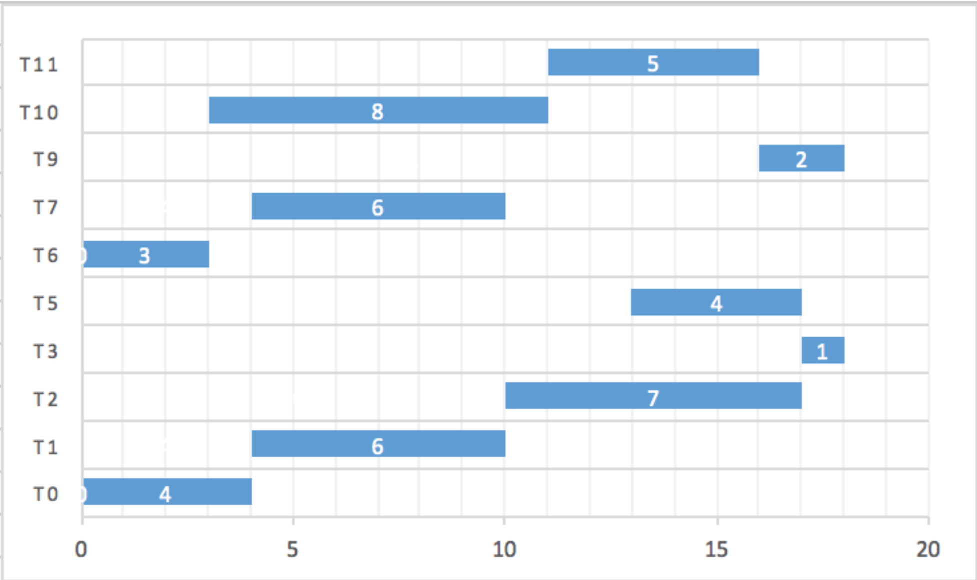
3.3. Output do LP Solve

A imagem abaixo é o output resultante do input inserido no LPSolve. Observando-o, podemos ver que foi possível obter um output do programa, ou seja, o problema é possível tendo uma solução ótima de 10420 U.M.

Variables	MILP ...	result
	10420	10420
r0_1	0	0
r0_2	0	0
r1_1	0	0
r1_2	0	0
r2_1	0	0
r2_2	0	0
r3_1	0.499...	0.499...
r3_2	0.499...	0.499...
r5_1	0	0
r5_2	0	0
r6_1	1	1
r6_2	1	1
r10_1	0	0
r10_2	0	0
r11_1	1	1
r11_2	1	1
y0	0	0
y1	0	0
y2	0	0
y3	1	1
y5	0	0
y6	1	1
y10	0	0
y11	1	1
t0	0	0
t6	0	0
t1	4	4
t2	10	10
t7	4	4
t3	17	17
t5	13	13
t10	3	3
t9	16	16
t11	11	11
tf	18	18
y	4	4

3.4. Diagrama de Gantt

Com ajuda do output obtido no LP Solve foi possível realizar um plano de execução do projeto através do seguinte diagrama de Gantt:



Cada linha representa uma atividade ti (tal que i ϵ {0,1,2,3,4,5,6,7,8,9,10,11,f}) que é iniciada só quando todas as atividades precedentes ou atividades prioritárias a estas acabam.

Como resultado final, as atividades que sofreram uma redução foram:

- ✓ Atividade 3, que sofreu uma redução tanto do custo 1 (0.5 U.T.), como do custo 2 (0.5 U.T.), passando a ter uma duração de 1 U.T. (2-0.5-0.5 = 1), por 150 U.M. adicionais ao custo normal da atividade (200*0.5 + 100*0.5 = 150);
- ✓ Atividade 6, que sofreu uma redução tanto do custo 1 (1 U.T.), como do custo 2 (1 U.T.) passando a ter uma duração de 3 U.T. (5 – 1 – 1 = 3), por 270 U.M. adicionais ao custo normal da atividade (180*1 + 90*1 = 270);
- ✓ Atividade 11, que sofreu uma redução tanto do custo 1 (1 U.T.) como do custo 2 (1 U.T.), passando a ter uma duração de 5 U.T. (7 – 1 – 1 = 5).

No final, como comprovado no diagrama de gantt, foi possível reduzir o plano de execução para 18 U.T. com um custo normal de 9100 U.M. mais um custo adicional de reduções de 1320 U.M., dando um total de custos de 10420 U.M., comprovando que o output do LPSolve está correto.