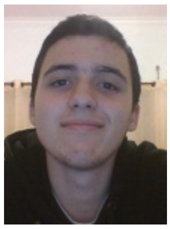
Universidade do Minho
Escola de Engenharia
Departamento de Informática
Mestrado Integrado em Engenharia Informática

Modelos Determinísticos de Investigação Operacional

Trabalho Prático Nº. 1

Braga, 12 de Outubro de 2015

Pedro Vieira Fortes
Nº64309



João Pedro Pereira Fontes
Nº71184

Resumo

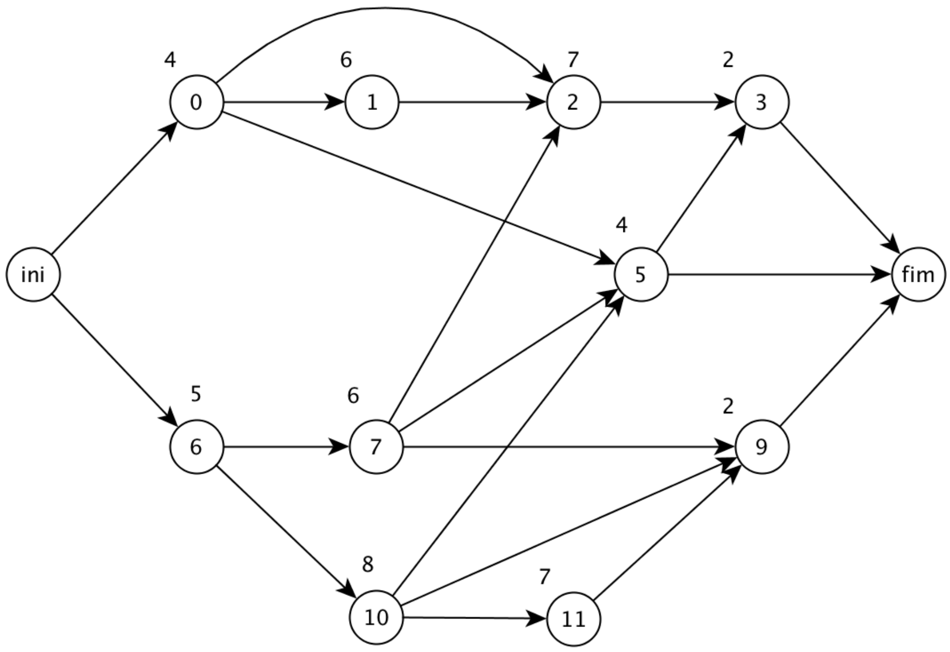
Este relatório tem como intuito apresentar a análise e resolução do trabalho que nos foi proposto na unidade curricular de Modelos Determinísticos de Investigação Operacional pelo professor Valério Carvalho. Neste trabalho deveríamos resolver um caso de gestão de projetos com a ajuda da ferramenta LP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

Índice

Parte 1

1.1. Rede do Projeto

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).

1.2. Formulação do Problema

- **Objetivo**

  O objetivo deste modelo é encontrar o caminho mais longo (caminho crítico) entre os nodos início e fim, escolhendo os nodos (de 1 a 11, exceto 4 e 8) que resultam um maior peso no final. Este modelo foi realizado baseado na rede apresentada em cima.

- **Variáveis de Decisão**

  As nossas variáveis de decisão são binárias, ou seja, será atribuído o valor 0 ou 1 aos vértices **vi_ j** sendo i o nodo origem e j o nodo destino (tal que i, j ϵ {0,1,2,3,4,5,6,7,8,9,10,11} e ainda i pode ser igual a i e j pode ser igual f)
  Assim, para que o modelo seja resolvido é atribuído o valor 0 ou 1 às arestas do caminho, conforme estas constam ou não para um caminho mais crítico.

- **Função Objetivo**

  Tem-se então a seguinte função objetivo:

$$\begin{aligned} max{:}\ 4\,vi\_0 \ +\ 5\,vi\_6 +\ 6\,v0\_1 \ +\ 7\,v0\_2 \ +\ 4\,v0\_5 +\ 7\,v1\_2 +\ 2\,v2\_3 \\ +\ 2\,v5\_3 +\ 6\,v6\_7 \ +\ 8\,v6\_10 +\ 7\,v7\_2 \ +\ 4\,v7\_5 \\ +\ 2\,v7\_9 +\ 4\,v10_5 +\ 2\,v10\_9 \ +\ 7\,v10\_11 +\ 2\,v11\_9; \end{aligned}$$

  Cada parcela da função objetivo representa o custo em tempo de uma determinada aresta. Por exemplo, 4*vi_0 representa o custo de 4 U.T em escolher o caminho que se inicia no nodo i e termina no nodo 0.

- **Restrições**

  O problema em questão exigia as seguintes restrições:

  - para decidir um só caminho crítico, em cada nodo só poderá ser escolhido 1 dos vértices vi_j, ou seja, no nodo i só será possível escolher um único destino *j*. Por exemplo, usando a primeira equação abaixo, do nodo origem ini só é possível escolher o vértice para o nodo 0 ou para o nodo 6. Nestas restrições temos duas equações que representam a obrigatoriedade de iniciar e finalizar o projeto, e as restantes restrições são inequações que representam a possível passam por esses nodos.

$$vi\_0 \ +\ vi\_6 \ =\ 1;$$
$$v0\_1 \ +\ v0\_2 \ +\ v0\_5 \ <=\ 1;$$
$$v1\_2 \ <=\ 1;$$

$$v2\_3 \ <= \ 1;$$
$$v5\_6 \ <= \ 1;$$
$$v6\_7 \ + \ v6\_10 \ <= \ 1;$$
$$v7\_2 \ + \ v7\_5 \ + \ v7\_9 \ <= \ 1;$$
$$v10\_5 \ + \ v10\_9 \ + \ v10\_11 \ <= \ 1;$$
$$v11\_9 \ <= \ 1;$$
$$v3\_f \ + \ v5\_f \ + \ v9\_f \ = \ 1;$$

- para obrigar que o fluxo de entrada dos nodos seja igual ao fluxo de saída. Por exemplo, para o nodo 5, o fluxo de entrada, ou seja, as variáveis vi_ j em que j = 5 (destino), deverá ser igual ao fluxo de saída, ou seja, as variáveis vi_ j em que i=5 (origem):

$$vi\_0 \ = \ v0\_1 \ + \ v0\_2 \ + \ v0\_5;$$
$$v0\_1 \ = \ v1\_2;$$
$$v0\_2 \ + \ v1\_2 \ + \ v7\_2 \ = \ v2\_3;$$
$$v2\_3 \ + \ v5\_3 \ = \ v3\_f;$$
$$v0\_5 \ + \ v7\_5 \ + \ v10\_5 \ = \ v5\_3 \ + \ v5\_f;$$
$$vi\_6 \ = \ v6\_7 \ + \ v6\_10;$$
$$v6\_7 \ = \ v7\_2 \ + \ v7\_5 \ + \ v7\_9;$$
$$v7\_9 \ + \ v10\_9 \ + \ v11\_9 \ = \ v9\_f;$$
$$v6\_10 \ = \ v10\_5 \ + \ v10\_9 \ + \ v10\_11;$$
$$v10\_11 \ = \ v11\_9;$$

1.3. Input no LP Solve

Este foi o input realizado no programa LP Solve, no qual inclui a declaração das variáveis binárias (como mencionado anteriormente), a função objetivo, as restrições do problema e alguns comentários sobre o próprio input:

```
/* Função Objetivo */
max: 4 vi_0 + 5 vi_6
    + 6 v0_1 + 7 v0_2 + 4 v0_5
    + 7 v1_2
    + 2 v2_3
    + 2 v5_3
    + 6 v6_7 + 8 v6_10
    + 7 v7_2 + 4 v7_5 + 2 v7_9
    + 4 v10_5 + 2 v10_9 + 7 v10_11
    + 2 v11_9;

/*Restrições*/
vi_0 + vi_6 = 1;
v0_1 + v0_2 + v0_5 <= 1;
v1_2 <= 1;
v2_3 <= 1;
v5_6 <= 1;
v6_7 + v6_10 <= 1;
```

```
v7_2 + v7_5 + v7_9 <= 1;
v10_5 + v10_9 + v10_11 <= 1;
v11_9 <= 1;
v3_f + v5_f + v9_f = 1;

vi_0 = v0_1 + v0_2 + v0_5;
v0_1 = v1_2;
v0_2 + v1_2 + v7_2 = v2_3;
v2_3 + v5_3 = v3_f;
v0_5 + v7_5 + v10_5 = v5_3 + v5_f;
vi_6 = v6_7 + v6_10;
v6_7 = v7_2 + v7_5 + v7_9;
v7_9 + v10_9 + v11_9 = v9_f;
v6_10 = v10_5 + v10_9 + v10_11;
v10_11 = v11_9;

/* Variáveis de Decisão */
/*vi_j - peso para chegar do vertice i ao vertice j*/
Bin vi_0, vi_6,
v0_1, v0_2, v0_5,
v1_2,
v2_3,
v3_f,
v5_3,v5_f,
v6_7, v6_10,
v7_2, v7_5, v7_9,
v9_f,
v10_5, v10_9, v10_11
v11_9;
```

1.4. Output do LP Solve

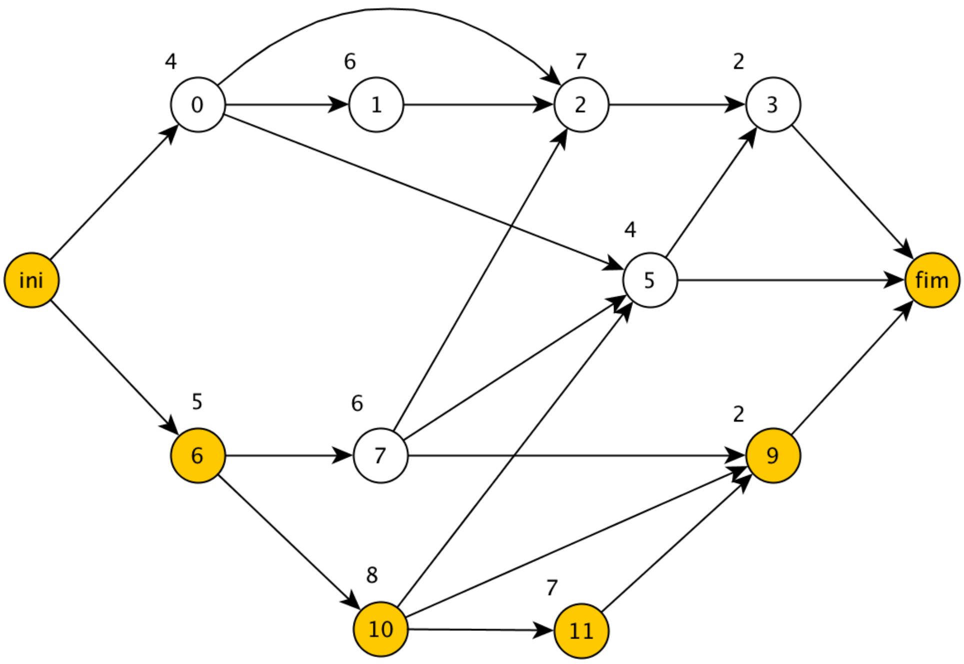
Este foi o output resultante do input inserido no LP Solve:

Parte 1	
vi_0	0
vi_6	1
v0_1	0
v0_2	0
v0_5	0
v1_2	0
v2_3	0
v5_3	0
v5_6	0
v6_7	0
v6_10	1
v7_2	0
v7_5	0
v7_9	0
v10_5	0
v10_9	0
v10_11	1
v11_9	1
v3_f	0
v5_f	0
v9_f	1
	22

Como podemos ver, foi possível obter um output do programa, ou seja, o problema é possível tendo uma solução ótima de 22 U.T.

1.5. Caminho Crítico

Através do output obtido no LP Solve podemos concluir que o tempo máximo obtido é através da realização da atividade 6, 10, 11 e 9. Em seguida temos uma apresentação em rede do caminho crítico assinalado a amarelo:



Parte 2

2.1. Formulação do Problema

- **Objetivo**

  O objetivo deste modelo é minimizar o tempo de execução total do projeto através da imposição de restrições nos tempos de início de cada atividade.

- **Variáveis de Decisão**

  As nossas variáveis de decisão são inteiras, **ti,** no qual representa 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.

- **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.

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

$$t9 >= t10 + 8;$$
$$t9 >= t11 + 7;$$
$$t10 >= t6 + 5;$$
$$t11 >= t10 + 8;$$
$$tf >= t3 + 2;$$
$$tf >= t5 + 4;$$
$$tf \geq t9 + 2;$$

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 incluí a declaração das variáveis inteiras (como mencionado anteriormente), a função objetivo, as restrições do problema e alguns comentários sobre o próprio 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;

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

2.2. Output do LP Solve

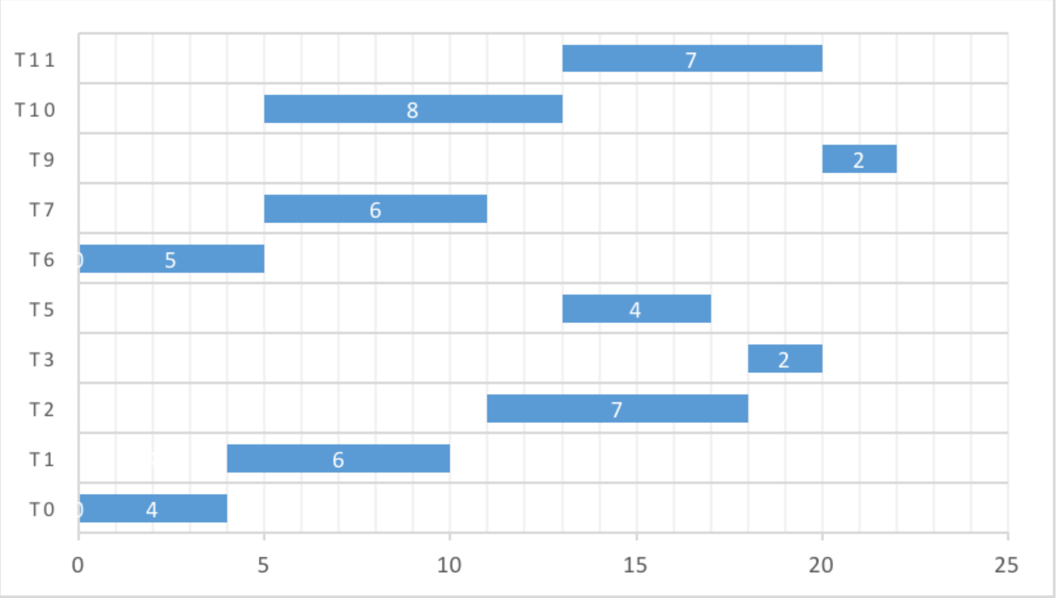
Este foi o output resultante do input inserido no LP Solve:

Parte 2	
t0	0
t1	5
t2	11
t3	20
t5	16
t6	0
t7	5
t9	20
t10	5
t11	13
tf	22
	22

Como podemos ver, foi possível obter um output do programa, ou seja, o problema é possível tendo uma solução ótima de 22 U.T. Com isto podemos concluir que com diferentes modelos de programação podemos representar o mesmo problema e chegar ao mesmo resultado.

2.4. Diagrama de Gantt

Com ajuda do output obtido no LP Solve foi possível realizar um plano de execução do projeto através de um 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 acabam.

2.5. Atividade pertencente ao Caminho Crítico

Para análise deste caso, escolhemos a atividade 9. Assim, através do diagrama de Gantt podemos concluir que essa mesma atividade só pode ser iniciada no instante de tempo 20 U.T., ou seja, após os seus precedentes 7, 10 e 11 acabarem.

2.6. Atividade não pertencente ao Caminho Crítico

a) Escolhendo a atividade 5, podemos concluir que essa mesma atividade deve ser iniciada no mínimo no instante 13 U.T., pois esta está dependente das atividades 0, 7 e 10, e apesar de, por exemplo, a atividade 7 terminar no instante de tempo 11, a atividade 10 só termina nas 13 U.T, ou seja, por depender também da atividade 10, a atividade 5 só inicia depois de todas as atividades precedentes tiverem acabado.
b) Mantendo na atividade 5, esta tarefa pode ser inicia no máximo nas 16 U.T, pois como esta tem uma duração de 4 U.T, e é predescendente da atividade 3 se for iniciada no instante 16, acaba no instante 20, dando tempo ainda de iniciar a atividade 3 que tem uma duração de 2 U.T. acabando no instante 22 não atrasando assim o projeto.

Parte 3

3.1. Formulação do Problema

- **Objetivo**

  O objetivo deste modelo é minimizar o custo suplementar de redução dos tempos nos nodos obedecendo às novas restrições postas. Cada nodo está associado a um tempo máximo de redução e um custo suplementar.

- **Variáveis de Decisão**

  As nossas variáveis de decisão são inteiras: **ti,** no qual representa o tempo acumulado até o nodo i (tal que i ϵ {0,1,2,3,5,6,7,9,10,11,f}), representando o tempo mínimo decorrido até se poder iniciar a atividade do nodo i em causa; tf representa o tempo total de execução das atividades pertencentes ao caminho crítico, respeitando as suas devidas precedências; ri é o número U.T. reduzidas no nodo i (tal que i ϵ {0,1,2,3,5,6,10,11,f});

- **Função Objetivo**

  Tem-se então a seguinte função objetivo:

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

  Na função objetivo, todas as parcelas que não têm nenhum coeficiente ri associado são os custos constantes ou normais de se realizar uma determinada atividade. Os coeficientes dos ri são o custo suplementar de se reduzir por U.T. no nodo i. É de notar que as atividades 7 e 9 não têm qualquer custo suplementar associado.

- **Restrições**

  O problema em questão exigia as seguintes restrições:

  - Limitar as reduções em cada nodo segundo a tabela dada no enunciado:

$$
\begin{aligned}
r0 \;&<=\; 1; \\
r1 \;&<=\; 2; \\
r2 \;&<=\; 4; \\
r3 \;&<=\; 1; \\
r5 \;&<=\; 1; \\
r6 \;&<=\; 2;
\end{aligned}
$$

$$r10 <= 1;$$
$$r11 <= 2;$$

- Tal como na parte II, para representar as precedências dos nodos, utilizamos a variável ti nas restrições, onde ti representa o tempo mínimo decorrido até se poder iniciar a atividade do nodo i em causa. A diferença nestas restrições em relação à parte 2 é a possibilidade de reduzir o tempo normal de cada atividade por ri, exceto nas atividades 7 e 9 que não têm ri associado. 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.

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

Sabemos também que, pelo enunciado, o tempo total obtido na parte 1, ou seja, tf, deve ser reduzido em 3 U.T. sendo essa redução representada por y.

$$tf = 22 - y;$$
$$y = 3;$$

- 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 incluí a declaração das variáveis inteiras (como mencionado anteriormente), a função objetivo, as restrições do problema e alguns comentários sobre o próprio input:

```
/* Objective function */
min: 400 + 100 r0 + 1000 + 300 r1 + 1400 + 500 r2
     + 300 + 100 r3 + 1000 + 800 r5 + 800 + 90 r6
     + 900 + 300 + 1600 + 500 r10 + 1400 + 300 r11;

/* Variable bounds */

r0 <= 1;
r1 <= 2;
r2 <= 4;
r3 <= 1;
r5 <= 1;
r6 <= 2;
r10 <= 1;
r11 <= 2;


t0 = 0;
t6 = 0;

t1 >= t0 + 4 - r0;

t2 >= t0 + 4 - r0;
t2 >= t1 + 6 - r1;
t2 >= t7 + 6;

t3 >= t2 + 7 - r2;
t3 >= t5 + 4 - r5;

t5 >= t0 + 4 - r0;
t5 >= t7 + 6;
t5 >= t10 + 8 - r10;

t7 >= t6 + 5 - r6;

t9 >= t7 + 6;
t9 >= t10 + 8 - r10;
t9 >= t11 + 7 - r11;

t10 >= t6 + 5 - r6;

t11 >= t10 + 8 - r10;

tf >= t3 + 2 - r3;
```

```
tf >= t5 + 4 - r5;
tf >= t9 + 2;

tf = 22 - y;
y = 3;

x = min - 100;

Int y, t0, t1, t2, t3, t5, t6, t7, t9, t10, t11, tf,
    r0, r1, r2, r3, r5, r6, r10, r11;
```

3.3. Output do LP Solve

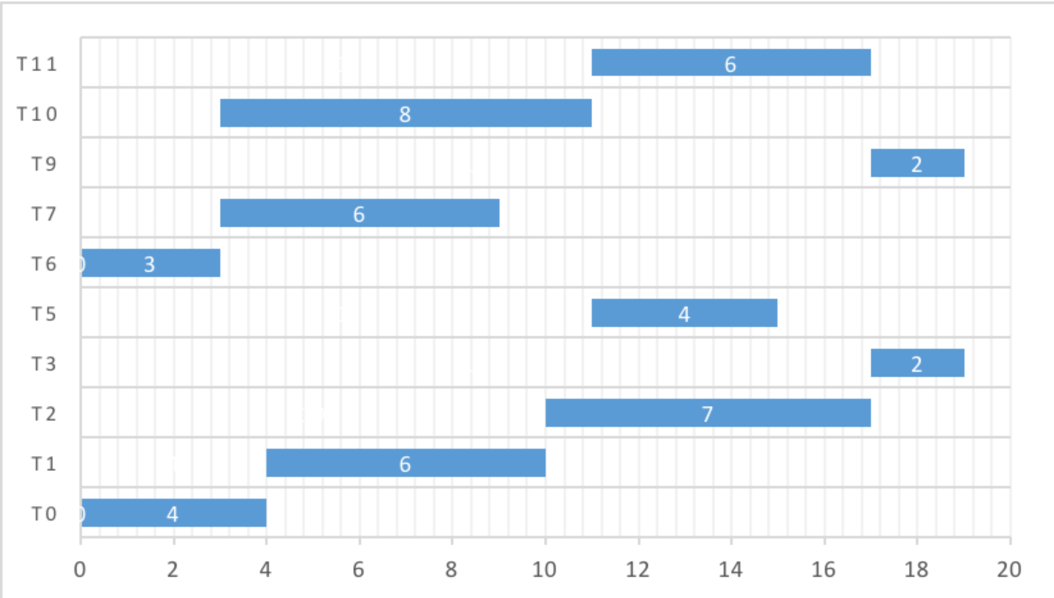
Este foi o output resultante do input inserido no LP Solve:

Parte 3	
t0	0
t1	4
t2	10
t3	17
t5	13
t6	0
t7	4
t9	17
t10	3
t11	11
tf	19
r0	0
r1	0
r2	0
r3	0
r5	0
r6	2
r10	0
r11	1
y	3
	9580

Como podemos ver, foi possível obter um output do programa, ou seja, o problema é possível tendo uma solução ótima com um custo mínimo de 9580 U.M.

3.4. Reduções

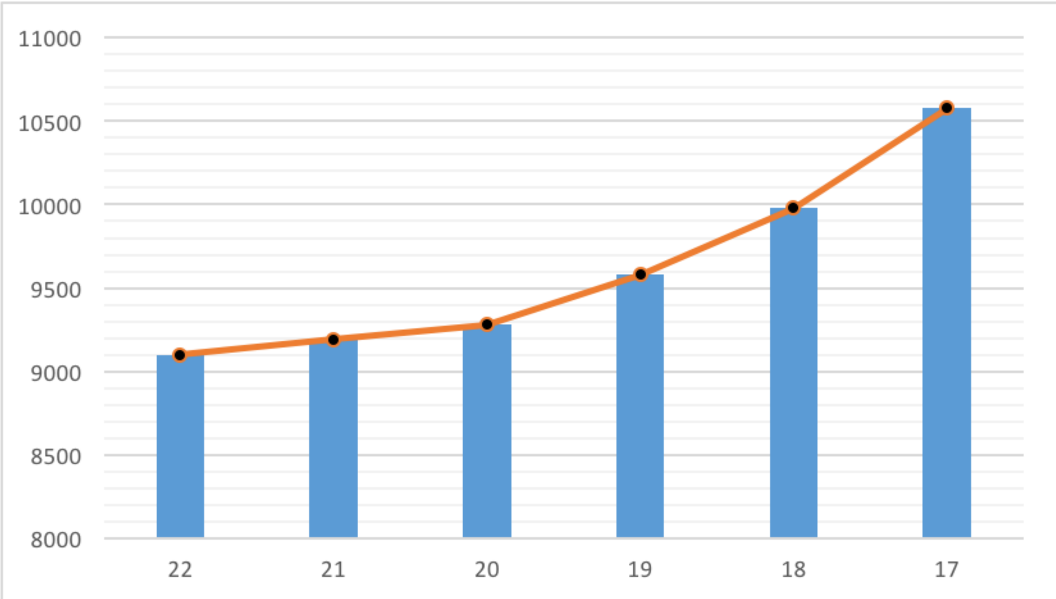
Como se pode observar pelo output do LP SOLVE, a solução consiste em reduzir 2 U.T. na atividade 6, e 1 U.T. na atividade 10, dando um tempo total de duração de 19 U.T, como especificado nas restrições. Com estas reduções teremos um custo suplementar para atividade 6 de 180 U.M. (90 * r6) e para atividade 11 de 300 U.M. (300 * r11), o que resulta de um custo suplementar total de 480 U.M. Assim o custo final será de 9580 U.M. (Custo normal + custo suplementar = 9100 + 480). É de notar que estas reduções foram realizadas apenas no caminho crítico, não sendo necessário alterar em outras atividades do projeto. Podemos confirmar esse mesmo resultado com o seguinte diagrama de Gantt com as novas durações:



Como podemos observar, em comparação com o diagrama da parte 2, este tem uma redução nas atividades 6 que permite que as atividades 7 e 10 comecem mais cedo e também houve a necessidade de reduzir na atividade 11 para que a atividade 9 pudesse começar mais cedo e assim acabar no tempo estipulado nas restrições.

Parte 4

Após uma analise dos vários resultados obtidos alterando a variável y da parte 3, obtemos o seguinte gráfico:



Como podemos observar, à medida que é exigido menos U.T. para a conclusão do projeto, mais U.M. é necessário gastar, crescendo de uma forma exponencial. Para valores inferiores a 16 U.T., inclusive, o modelo é inviável, não sendo possível reduzir o tempo das atividades para que possa cumprir o tempo total do projeto. Então, com estas condições, o projeto no mínimo pode ser terminado a 17 U.T. com um custo suplementar de 1480 U.M. e um custo total do projeto de 10580 U.M.

Parte 5

5.1. Formulação do Problema

- **Alterações**

Em relação à parte 3, as alterações realizadas para a parte 5 foram:

  - em todo o projeto, em vez de 1 variável de redução, teremos 2 variáveis de redução por cada nodo: uma associada ao custo 1 e outra associada ao custo 2.
  - nas restrições, em vez de subtrair apenas 1 variável de redução ao tempo da atividade associada, iremos subtrair duas variáveis de redução ao tempo da atividade associado.
  - na função objetivo, em vez de um nodo ter só um custo de redução associado, terá agora 2

- **Objetivo**

O objetivo deste modelo nesta parte é bastante semelhante ao da parte 3: minimizar o custo suplementar de redução dos tempos nos nodos obedecendo às novas restrições postas. No entanto, em vez de cada nodo estar associado a um tempo máximo de redução e um custo suplementar, estará associado a 2 custos suplementares destintos e seus tempos máximos de redução.

- **Variáveis de Decisão**

As nossas variáveis de decisão são inteiras: **ti,** no qual representa o tempo acumulado até o nodo i (tal que i ∊ {0,1,2,3,4,5,6,7,8,9,10,11,f}), representando o tempo mínimo decorrido até se poder iniciar a atividade do nodo i em causa; tf representa o tempo total de execução das atividades pertencentes ao caminho crítico, respeitando as suas devidas precedências; ri_j é o número U.T. reduzidas no nodo i com o custo j associado.

- **Função Objetivo**

Tem-se então a seguinte função objetivo:

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

Na função objetivo, todas as parcelas que não têm nenhum coeficiente ri associado são os custos constantes ou normais de se realizar uma determinada atividade. Os coeficientes dos ri são o custo suplementar de se reduzir por U.T. no nodo i. É de notar que as atividades 7 e 9 não têm qualquer custo suplementar associado.

- **Restrições**

O problema em questão exigia as seguintes restrições:

- Limitar as reduções em cada nodo segundo a tabela dada no enunciado:

$$r0 <= 1;$$
$$r1 <= 2;$$
$$r2 <= 4;$$
$$r3 <= 1;$$
$$r5 <= 1;$$
$$r6 <= 2;$$
$$r10 <= 1;$$
$$r11 <= 2;$$

- Tal como na parte II, para representar as precedências dos nodos, utilizamos a variável ti nas restrições, onde ti representa o tempo mínimo decorrido até se poder iniciar a atividade do nodo i em causa. A diferença nestas restrições em relação à parte 2 é a possibilidade de reduzir o tempo normal de cada atividade por ri, exceto nas atividades 7 e 9 que não têm ri associado. 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.

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

Sabemos também que, pelo enunciado, o tempo total obtido na parte 1, ou seja, tf, deve ser reduzido em 3 U.T. sendo essa redução representada por y.

$$tf = 22 - y;$$
$$y = 3;$$

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

5.2. Input do LP Solve

Este foi o input realizado no programa LPSolve, no qual incluí a declaração das variáveis inteiras (como mencionado anteriormente), a função objetivo, as restrições do problema e alguns comentários sobre o próprio input:

```
/* Objective function */
min: 400 + 100 r0_1 + 200 r0_2 + 1000 + 300 r1_1 + 600 r1_2
    + 1400 + 500 r2_1 + 1000 r2_2 + 300 + 100 r3_1 + 200 r3_2
    + 1000 + 800 r5_1 + 1600 r5_2 + 800 + 90 r6_1 + 180 r6_2
    + 900 + 300 + 1600 + 500 r10_1 + 1000 r10_2
    + 1400 + 300 r11_2 + 600 r11_2;

/* Variable bounds */

r0_1 <= 0.5;
r0_2 <= 0.5;
r1_1 <= 1;
r1_2 <= 1;
r2_1 <= 1;
r2_2 <= 3;
r3_1 <= 0.5;
r3_2 <= 0.5;
r5_1 <= 0.5;
r5_2 <= 0.5;
r6_1 <= 1;
r6_2 <= 1;
r10_1 <= 0.5;
r10_2 <= 0.5;
r11_1 <= 1;
r11_2 <= 1;

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;

/*Int y, t0, t1, t2, t3, t5, t6, t7, t9, t10, t11, tf,
   r0_1, r0_2, r1_1, r1_2, r2_1, r2_2, r3_1, r3_2,
   r5_1, r5_2, r6_2, r10_1, r10_2, r11_1, r11_2, r6_1;
```

5.3. Output do LP Solve

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

Parte 5	
t0	0
t1	3,5
t2	9,5
t3	16,5
t5	12,5
t6	0
t7	3,5
t9	16
t10	3
t11	10,5
tf	18
r0_1	0,5
r0_2	0
r1_1	0
r1_2	0
r2_1	0
r2_2	0
r3_1	0,5
r3_2	0
r5_1	0
r5_2	0
r6_1	1
r6_2	1
r10_1	0,5
r10_2	0
r11_1	1
r11_2	0,5
y	4
	10170

Como podemos ver, foi possível obter um output do programa, ou seja, o problema é possível tendo uma solução ótima com um custo mínimo de 10170 U.M.

3.4. Reduções

Como se pode observar pelo output do LP SOLVE, a solução consiste em reduzir 0.5 U.T. na atividade 1 (com custo de 50 U.M. do custo 1), 0.5 U.T. na atividade 3 (com um custo de 50 U.M. do custo 1), 2 U.T na atividade 6 (com o custo de 90 U.M. do custo 1 mais 180 U.M. do custo 2), 0.5 U.T. na atividade 10 (com o custo de 250 U.M. do custo 1) e 1.5 U.M. na atividade 11 (com o custo de 500 U.M. do custo 1 mais 500 U.M. do custo 2), dando um tempo total de duração do projeto de 18 U.T, como especificado nas restrições. O custo suplementar total é de 1070 U.M. Assim o custo final será de 10170 U.M. (Custo normal + custo suplementar = 9100 + 1070). É de notar que desta vez foi também necessário efetuar reduções em atividades que não pertencem ao caminho crítico, pois o tempo total pedido era menor. Podemos confirmar esse mesmo resultado com o seguinte diagrama de Gantt com as novas durações: