



Escola de Engenharia
Universidade do Minho

DEPARTAMENTO DE PRODUÇÃO E SISTEMAS
Mestrado Integrado em Engenharia Informática
Métodos Determinísticos de Investigação Operacional

TRABALHO 1 - GESTÃO DE PROJETO

Grupo 25

Bruno Pereira
Aluno nº 72628

André Santos
Aluno nº 61778

Jéssica Pereira
Aluno nº 71164

Braga, 12 de Outubro de 2015

Resumo

Este relatório tem como objetivo apresentar a experiência de modelação e resolução dos casos propostos na realização do 1º trabalho prático da unidade curricular de Modelos Determinísticos de Investigação Operacional. Além da apresentação dos modelos, procuram-se justificar detalhadamente todas as decisões tomadas.

O relatório encontra-se dividido por capítulos, em que cada capítulo corresponde a uma parte do trabalho.

Conteúdo

1	Parte 1	3
1.1	Análise do problema	3
1.2	Modelo	4
1.2.1	Parâmetros	4
1.2.2	Variáveis de decisão	4
1.2.3	Função objetivo	4
1.2.4	Restrições	5
1.3	Ficheiro <i>Input</i>	6
1.4	<i>Output</i> produzido pelo <code>lp_solve</code>	8
1.5	Resultado	9
1.6	Validação do Modelo	9
1.6.1	Variáveis de Decisão	9
1.6.2	Função objetivo	9
1.6.3	Restrições	10
2	Parte 2	12
2.1	Análise Problema	12
2.2	Modelo	12
2.2.1	Parâmetros	12
2.2.2	Variáveis de decisão	12
2.2.3	Função Objetivo	12
2.2.4	Restrições	12
2.3	Ficheiro <i>input</i>	13
2.4	<i>Output</i> produzido pelo <code>lp_solve</code>	15
2.5	Resultado	16
2.6	Validação do modelo	16
2.6.1	Variáveis de decisão	17
2.6.2	Função objetivo	17
2.6.3	Restrições	17
2.7	Atividade no caminho crítico	19
2.8	Atividade fora do caminho crítico	19
3	Parte 3	20
3.1	Análise do problema	20
3.2	Modelo	20
3.2.1	Parâmetros	20
3.2.2	Variáveis de decisão	20
3.2.3	Função Objetivo	21

3.2.4	Restrições	21
3.3	Ficheiro de <i>input</i>	23
3.4	<i>Output</i> produzido pelo <code>lp_solve</code>	24
3.5	Resultado	26
3.6	Validação do modelo	27
3.6.1	Variáveis de decisão	28
3.6.2	Função objetivo	28
3.6.3	Restrições	28
4	Parte 4	32
4.1	Análise do problema	32
4.2	Modelo	32
4.2.1	Parâmetros	32
4.2.2	Variáveis de decisão	32
4.2.3	Função Objetivo	32
4.2.4	Restrições	33
4.3	<i>Input</i> do <i>script</i> em <i>BASH</i>	34
4.4	Resultado	35
5	Parte 5	38
5.1	Análise do problema	38
5.2	Modelo	38
5.2.1	Parâmetros	38
5.2.2	Variáveis de decisão	38
5.2.3	Função objetivo	39
5.2.4	Restrições	40
5.3	Ficheiro <i>Input</i>	41
5.4	<i>Output</i> produzido pelo <code>lp_solve</code>	43
5.5	Resultado	44
5.6	Validação do Modelo	45
5.6.1	Variáveis de Decisão	45
5.6.2	Função objetivo	45
5.6.3	Restrições	45

1. Parte 1

1.1 Análise do problema

Nesta 1ª parte do trabalho, pretende-se criar um modelo que permita descobrir o caminho mais longo de um grafo orientado acíclico. Visto que neste grafo em particular os nós correspondem a atividades e as arestas unidirecionais traduzem as precedências entre as atividades, a rede pode ser entendida como um projeto, cujas atividades devem ser realizadas obedecendo à ordem das suas precedências. Neste contexto, encontrar o caminho mais longo significa por isso encontrar a duração mínima do projeto.

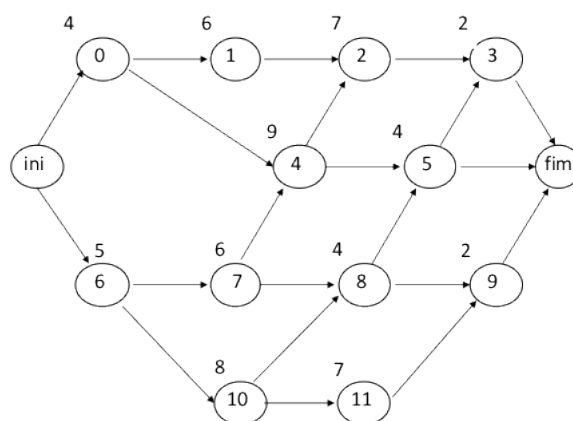


Figura 1.1: Grafo Inicial do enunciado

Antes de partir para a formulação do modelo, foi necessário saber qual a rede a considerar. À rede fornecida no enunciado (figura 1.1) foi necessário retirar dois nós, de acordo com a metodologia apresentada na secção *Determinação da Lista de Atividades* presente no final do enunciado. Os números de aluno dos autores deste relatório são 61778, 71164 e 72628. Como o número mais alto é 72628, então $D=2$ e $E=8$, sendo por isso os nós 2 e 8 a ser retirados da rede. A rede resultante da remoção destes dois nós tem a representação gráfica mostrada na figura 1.2:

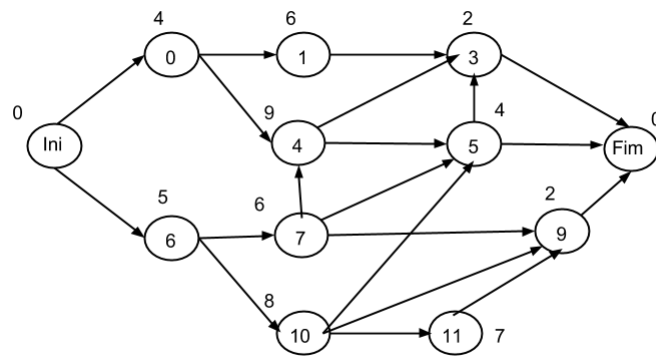


Figura 1.2: Grafo resultante da remoção das atividades 2 e 8, com indicação da duração de cada atividade (em unidades de tempo arbitrárias)

1.2 Modelo

1.2.1 Parâmetros

Os parâmetros deste modelo são as precedências e as durações de cada atividade.

1.2.2 Variáveis de decisão

O modelo desenvolvido nesta primeira parte tem como objetivo achar o caminho mais longo da rede. Visto que se pretende achar um caminho, cada arco do grafo terá associada uma variável de decisão, que tomará o valor de 1 ou 0, caso esse arco faça ou não parte do caminho mais longo, respectivamente. As variáveis de decisão serão por isso binárias. Embora a restrição relativamente ao fato das variáveis serem binárias não se encontrar explícita no modelo, ela é garantida por características específicas deste modelo que serão detalhadas na secção 1.2.4.

Relativamente à nomenclatura das variáveis, usou-se X_{I-J} para representar a aresta que vai da atividade I para a atividade J. Assim, X_{2-4} representa a aresta que vai da atividade 2 para a atividade 4.

1.2.3 Função objetivo

Como se pretende achar o caminho mais longo, a função objetivo terá que ser uma expressão que indique a duração de um caminho, que queremos que tome o maior valor possível. Trata-se por isso de um problema de maximização.

Visto que os valores das variáveis de decisão indicam os arcos que fazem ou não parte de um caminho, para se construir a função objetivo com essas variáveis, foi necessário associar um custo a cada um dos arcos. A decisão tomada foi a de assumir que cada arco tem um custo correspondente à duração da atividade de onde esse arco é originado. Por exemplo, o arco X_{0-1} tem origem na atividade 0 e destino na atividade 1, e terá um custo de 4, visto ser essa a duração da atividade 0. O sentido em termos práticos de assumir que a duração não está na atividade mas sim nas arestas do grafo que dele saem corresponde a dizer que a atividade em si não tem duração, o que tem duração é sim a passagem dessa atividade para uma outra. Embora isto possa ser pouco intuitivo em termos reais, esta consideração revela-se extremamente útil para a resolução do modelo como veremos de seguida.

Em qualquer solução admissível, o custo de um arco só deverá ter influencia no valor da função objetivo se o arco fizer parte do caminho. Uma vez que as variáveis de decisão traduzem

com 1 ou 0 o facto de o arco fazer ou não parte do caminho, então para saber o custo efetivo de um arco numa determinada solução, basta multiplicar o seu custo pela variável de decisão associada. Designando por C_I esse custo efetivo, temos a seguinte expressão:

$$C_I = C_{I_J} \times X_{I_J}$$

Onde:

C_{I_J} Custo associado ao arco que vai de I para J - parâmetro do problema

X_{I_J} Variável de decisão indicativa se o arco faz ou não parte do caminho, conforme detalhado na secção 1.2.2.

Esta multiplicação faz sentido, uma vez que tanto as variáveis de decisão como as durações das atividades estão associadas aos arcos. Tal mostra a utilidade em termos de resolução de modelo de anteriormente se ter associado a duração das atividades aos arcos.

A função objetivo será o somatório de todos esses custos. Em termos genéricos, pode ser escrita como:

$$\max z = \sum C_I$$

Expandindo a expressão e substituindo os valores de C_{I_J} pelos valores de custos do enunciado, juntamente com as variáveis de decisão, temos a seguinte expressão:

$$\begin{aligned} \max: & 0 \text{ Xini}_0 + 0 \text{ Xini}_6 + 4 \text{ X0}_1 + 4 \text{ X0}_4 \\ & + 6 \text{ X1}_3 + 2 \text{ X3}_{\text{fim}} + 9 \text{ X4}_3 + 9 \text{ X4}_5 \\ & + 4 \text{ X5}_3 + 4 \text{ X5}_{\text{fim}} + 5 \text{ X6}_7 + 5 \text{ X6}_{10} \\ & + 6 \text{ X7}_4 + 6 \text{ X7}_5 + 6 \text{ X7}_9 + 2 \text{ X9}_{\text{fim}} \\ & + 8 \text{ X10}_5 + 8 \text{ X10}_9 + 8 \text{ X10}_{11} \\ & + 7 \text{ X11}_9; \end{aligned}$$

1.2.4 Restrições

Com as restrições pretende-se indicar o espaço de possíveis soluções. Qualquer caminho no grafo corresponde a uma solução admissível do problema. A forma encontrada de representar um caminho em termos de restrições, foi a de assumir que em cada nó o *fluxo de entrada* = *fluxo de saída*. Com esta restrição, se nada entrar num nó, então nada sairá, o que corresponde a dizer que o nó não faz parte do caminho. Por outro lado, se alguma coisa entrar, ou seja, se o nó fizer parte do caminho, então queremos que o mesmo fluxo saia do nó. Isto corresponde a dizer que um caminho tem que começar e terminar, o fluxo não pode ficar “parado” no meio da rede.

Embora sejam um bom ponto de partida, as restrições de conservação do fluxo só por si são insuficientes, uma vez que permitem soluções em que entrem num nó 2 ou mais unidades de fluxo, e que as mesmas unidades saiam. Tais soluções não são admissíveis no modelo, pois não traduzem um caminho. Para resolver este problema foi necessário assumir que se injeta na rede 1 unidade de fluxo no nodo inicial, e que essa unidade de fluxo sai no nodo final. A injeção de uma unidade de fluxo, em conjunto com as restrições de conservação do fluxo resolvem o problema, uma vez que se entrar uma unidade de fluxo num nó, então essa unidade (e só essa) também sairá do nó. Por outras palavras, tem-se a definição de um caminho.

As restrições do modelo correspondem por isso a dizer que em cada nó:

$$\text{fluxo entrada} = \text{fluxo saida} \Leftrightarrow \text{fluxo entrada} - \text{fluxo saida} = 0$$

Ao ter a equação escrita da segunda forma, considera-se implicitamente o fluxo de entrada como sendo positivo e o fluxo de saída como negativo.

As variáveis do modelo são binárias, podendo tomar por isso apenas o valor de 0 ou 1. Tais restrições às variáveis são implícitas a este modelo particular, e por isso não foram incluídas no ficheiro *lp_solve* e são apenas referidas aqui. A restrição é implícita ao modelo pois como estamos a injetar uma unidade de fluxo num nó e ao mesmo tempo a dizer que o que entra em cada nó tem que ser o mesmo que sai, então todas as variáveis forçosamente terão apenas o valor de 0 ou 1. As restrições completas do modelo podem ser vistas na secção 1.3.

1.3 Ficheiro *Input*

O ficheiro de *input* é constituído pela função objetivo e restrições, detalhadas em secções anteriores.

```
/* === FUNCAO objetivo === */

max: 0 Xini_0 + 0 Xini_6 + 4 X0_1 + 4 X0_4
+ 6 X1_3 + 2 X3_fim + 9 X4_3 + 9 X4_5
+ 4 X5_3 + 4 X5_fim + 5 X6_7 + 5 X6_10
+ 6 X7_4 + 6 X7_5 + 6 X7_9 + 2 X9_fim
+ 8 X10_5 + 8 X10_9 + 8 X10_11
+ 7 X11_9;

/* === RESTRICOES === */

//Nodo Inicio
1 - Xini_6 - Xini_0 = 0;
//Nodo 0
Xini_0 - X0_1 - X0_4 = 0;
//Nodo 1
X0_1 - X1_3 = 0;
//Nodo 3
X1_3 + X4_3 + X5_3 - X3_fim = 0;
//Nodo 4
X0_4 + X7_4 - X4_3 - X4_5 = 0;
//Nodo 5
X4_5 + X7_5 + X10_5 - X5_3 - X5_fim = 0;
//Nodo 6
Xini_6 - X6_7 - X6_10 = 0;
//Nodo 7
X6_7 - X7_4 - X7_5 - X7_9 = 0;
//Nodo 9
X7_9 + X10_9 + X11_9 - X9_fim = 0;
//Nodo 10
```



```
X6_10 - X10_5 - X10_9 - X10_11 = 0;  
//Nodo 11  
X10_11 - X11_9 = 0;  
//Nodo Fim  
X3_fim + X5_fim + X9_fim - 1 = 0;
```

1.4 *Output* produzido pelo `lp_solve`

O output apresentado a seguir foi obtido por *copy-paste* direto resultante da execução do `lp_solve` num sistema linux para o ficheiro de input apresentado anteriormente:

Value of objective function: 26

Actual values of the variables:

Xini_0	0
Xini_6	1
X0_1	0
X0_4	0
X1_3	0
X3_fim	1
X4_3	0
X4_5	1
X5_3	1
X5_fim	0
X6_7	1
X6_10	0
X7_4	1
X7_5	0
X7_9	0
X9_fim	0
X10_5	0
X10_9	0
X10_11	0
X11_9	0

Actual values of the constraints:

R1	-1
R2	0
R3	0
R4	0
R5	0
R6	0
R7	0
R8	0
R9	0
R10	0
R11	0
R12	1

1.5 Resultado

De acordo com o ficheiro de *output* obtido, o caminho mais longo tem a duração de 26 unidades de tempo e é o que passa pelas arestas X_{ini_6} , X_{6_7} , X_{7_4} , X_{4_5} , X_{5_3} e X_{3_fim} . Em termos gráficos, o resultado é o apresentado na figura 1.3. As setas de linha cheia indicam as arestas que fazem parte do caminho mais longo, e os nós por onde esse caminho passa foram colocados a verde.

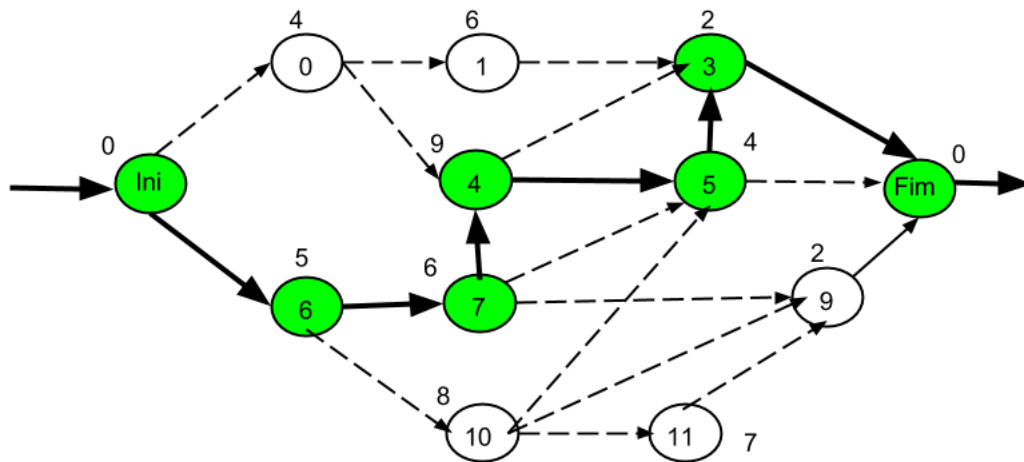


Figura 1.3: Grafo com indicação do caminho crítico obtido. Os valores em cada nó representam a duração (em unidades de tempo) da respetiva atividade

Este resultado indica que as atividades 6,7,4,5 e 3 devem ser vigiadas de perto e deve-se tentar garantir que são executadas nos tempos previstos, sem atrasos, caso contrário todo o projeto será atrasado..

1.6 Validação do Modelo

Para validar os resultados, tanto na função objetivo como nas restrições, substituímos os valores das variáveis de decisão pelo valor que estas tomam na solução que o *lp_solve* indica como ótima. A ideia é verificar que os valores das variáveis de decisão obtidos confirmam o valor da função objetivo obedecendo a todas as restrições.

Para evitar ao máximo o erro humano, a substituição de variáveis foi feita recorrendo a ferramentas que auxiliaram a substituição automática das variáveis pelo seu valor.

1.6.1 Variáveis de Decisão

No resultado obtido todas as variáveis são de facto binárias, tomam apenas o valor de 0 ou 1, tal como esperado.

1.6.2 Função objetivo

Depois da substituição das variáveis pelo seu valor, a função objetivo fica:

$$0 * 0 + 0 * 1 + 4 * 0 + 4 * 0 + 6 * 0 + 2 * 1 + 9 * 0 + 9 * 1 + 4 * 1 + 4 * 0 + 5 * 1 + 5 * 0 + 6 * 1 + 6 * 0 + 6 * 0 + 2 * 0 + 8 * 0 + 8 * 0 + 8 * 0 + 7 * 0 = 26$$

Inserindo a expressão numa calculadora verifica-se que a expressão é igual a 26, o que confirma o resultado obtido com o *lp_solve*.

1.6.3 Restrições

- Nodo Inicio

$$1 - X_{ini_6} - X_{ini_0} = 0$$

$$1 - 1 - 0 = 0$$

- Nodo 0

$$X_{ini_0} - X_{0_1} - X_{0_4} = 0$$

$$0 - 0 - 0 = 0$$

- Nodo 1

$$X_{0_1} - X_{1_3} = 0$$

$$0 - 0 = 0$$

- Nodo 3

$$X_{1_3} + X_{4_3} + X_{5_3} - X_{3_fim} = 0$$

$$0 + 0 + 1 - 1 = 0$$

- Nodo 4

$$X_{0_4} + X_{7_4} - X_{4_3} - X_{4_5} = 0$$

$$0 + 1 - 0 - 1 = 0$$

- Nodo 5

$$X_{4_5} + X_{7_5} + X_{10_5} - X_{5_3} - X_{5_fim} = 0$$

$$1 + 0 + 0 - 1 - 0 = 0$$

- Nodo 6

$$X_{ini_6} - X_{6_7} - X_{6_10} = 0$$

$$1 - 1 - 0 = 0$$

- Nodo 7

$$X_{6_7} - X_{7_4} - X_{7_5} - X_{7_9} = 0$$

$$1 - 1 - 0 - 0 = 0$$

- Nodo 9

$$X_{7_9} + X_{10_9} + X_{11_9} - X_{9_fim} = 0$$

$$0 + 0 + 0 - 0 = 0$$

- **Nodo 10**

$$X_{6_{10}} - X_{10_5} - X_{10_9} - X_{10_{11}} = 0$$

$$0 - 0 - 0 - 0 = 0$$

- **Nodo 11**

$$X_{10_{11}} - X_{11_9} = 0$$

$$0 - 0 = 0$$

- **Nodo Fim**

$$X_{3_{fim}} + X_{5_{fim}} + X_{9_{fim}} - 1 = 0$$

$$1 + 0 + 0 - 1 = 0$$

Assim conclui-se que todas as restrições são respeitadas.

2. Parte 2

2.1 Análise Problema

Na Parte I o problema tratava de achar o caminho mais longo de uma rede. Nesta parte, partindo da mesma rede, pretende-se descobrir o tempo em que cada atividade é iniciada, sabendo que todas as atividades são realizadas.

2.2 Modelo

2.2.1 Parâmetros

À semelhança da Parte I, também aqui os parâmetros do problema são a duração de cada atividade e as suas precedências.

2.2.2 Variáveis de decisão

As variáveis de decisão correspondem ao tempo em que cada atividade é iniciada. Assim, a cada atividade está associada uma variável de decisão. Relativamente ao nome, a opção tomada foi a de considerar T_i como o tempo de início da atividade i (em unidades de tempo arbitrárias), em que i corresponde ao número da atividade. Uma vez que apenas se pretende conhecer os tempos de início de cada atividade, estas são as únicas variáveis deste modelo.

2.2.3 Função Objetivo

Neste modelo, quer-se minimizar o tempo de execução total do projeto. Isso corresponde a dizer que queremos que a atividade final seja iniciada o mais cedo possível. A atividade final é na verdade “fictícia”, pois não corresponde a uma atividade que tenha de ser efetivamente realizada. No entanto para efeitos de modelação, é útil considerá-la, assumindo que é realizada após todas as outras da rede terem terminado e que tem duração de 0 unidades de tempo. Nestas condições, o tempo inicial da atividade final indica a duração do projeto.

Uma vez que a variável T_{fim} indica a duração do projeto, a função objetivo fica simplesmente:

$$\min z = T_{fim}$$

2.2.4 Restrições

Com as restrições pretende-se indicar o espaço de possíveis soluções. Sabe-se que uma atividade não pode começar sem que as que lhe precedem tenham terminado. Qualquer solução

que obedeça a este princípio é uma solução admissível para o problema. Para escrever as restrições é por isso necessário saber quando uma atividade termina. Ora, sabendo que as nossas variáveis de decisão indicam o tempo em que cada atividade se inicia e que temos a duração das mesmas como parâmetro do modelo, podemos dizer que o tempo final de uma atividade corresponde a somar o seu tempo de início com a sua duração. Ou seja:

$$Tf_i = T_i + D_i$$

Onde:

Tf_i Tempo em que a atividade i termina

T_i Tempo em que a atividade i começa (variável de decisão)

D_i Duração da atividade i

Dizer que uma atividade não pode começar sem que as que lhe precedem tenham terminado é o mesmo que dizer que o tempo inicial da atividade tem que ser maior que o tempo final de todas as atividades que lhe precedem. Assumindo que se tem uma atividade j que precede uma atividade i , podemos escrever que:

$$T_i \geq Tf_j + D_j$$

O modelo terá por isso uma restrição deste tipo por cada nodo e por cada atividade precedente ao nodo. Ou seja, um nó que tenha apenas 1 precedência, apenas originará uma restrição, enquanto que se o nodo tiver por exemplo 3 precedências, dará origem a 3 restrições — uma restrição para cada precedência do nodo. As restrições completas podem ser consultadas na secção 2.3

Visto que os tempos não podem ser negativos, neste modelo tem-se ainda restrições de não-negatividade:

$$T_i \geq 0, \forall i \in \{ini, 0, 1, 3, 4, 5, 6, 7, 9, 10, 11, fim\}$$

2.3 Ficheiro *input*

O ficheiro de *input* é constituído pela função objetivo e restrições, detalhadas em secções anteriores.

```
=== FUNCAO objetivo ===
```

```
min: Tfim;
```

```
=== RESTRICOES ===
```

```
//Nodo Inicial
```

```
Tini >= 0 + 0;
```

```
//Nodo 0
```

```

T0 >= Tini + 0;

//Nodo 1
T1 >= T0 + 4;

//Nodo 3
T3 >= T1 + 6;
T3 >= T5 + 4;
T3 >= T4 + 9;

//Nodo 4
T4 >= T0 + 4;
T4 >= T7 + 6;

//Nodo 5
T5 >= T4 + 9;
T5 >= T7 + 6;
T5 >= T10 + 8;

//Nodo 6
T6 >= Tini + 0;

//Nodo 7
T7 >= T6 + 5;

//Nodo 9
T9 >= T7 + 6;
T9 >= T11 + 7;
T9 >= T10 + 8;

//Nodo 10
T10 >= T6 + 5;

//Nodo 11
T11 >= T10 + 8;

//Nodo final
Tfim >= T3 + 2;
Tfim >= T5 + 4;
Tfim >= T9 + 2;

```


2.4 *Output* produzido pelo `lp_solve`

O output apresentado a seguir foi obtido por *copy-paste* direto resultante da execução do `lp_solve` num sistema linux para o ficheiro de input apresentado anteriormente:

Value of objective function: 26

Actual values of the variables:

Tfim	26
Tini	0
T0	0
T1	4
T3	24
T5	20
T4	11
T7	5
T10	5
T6	0
T9	20
T11	13

Actual values of the constraints:

R1	0
R2	4
R3	20
R4	4
R5	13
R6	11
R7	6
R8	9
R9	15
R10	15
R11	0
R12	5
R13	15
R14	7
R15	15
R16	5
R17	8
R18	2
R19	6
R20	6

2.5 Resultado

O resultado obtido indica uma duração total do projeto de 26 unidades de tempo. Os tempos iniciais de cada atividade representados graficamente podem ser vistos na figura 2.1 :

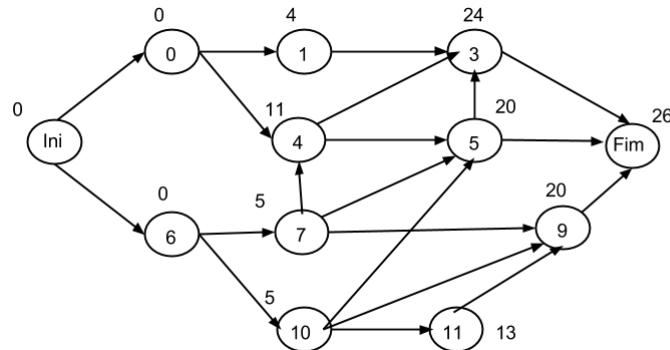


Figura 2.1: Grafo com tempo de início de cada atividade (em unidades de tempo arbitrárias)

O Diagrama de Gantt correspondente pode ser visto na figura 2.2:

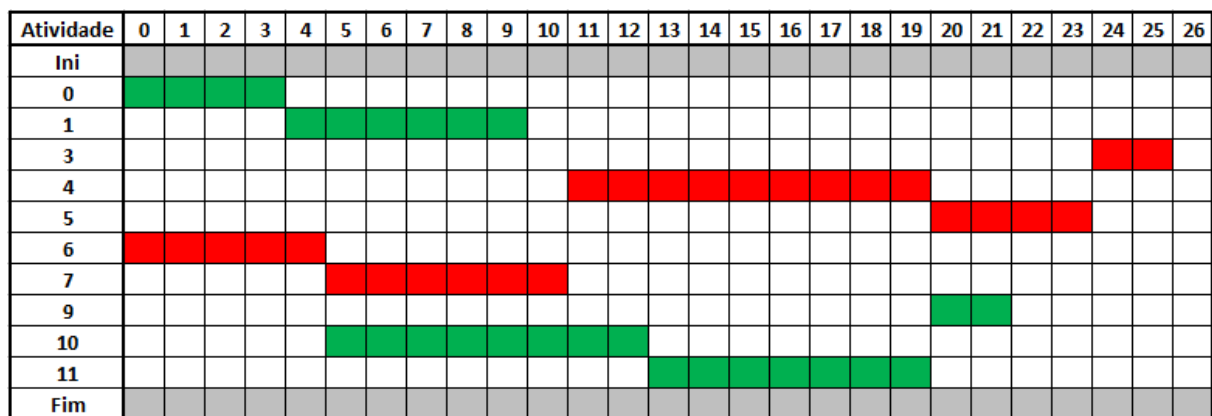


Figura 2.2: Diagrama de Gantt com indicação do caminho crítico a vermelho

Visto serem atividades fictícias de duração nula, representam-se atividades iniciais e finais a cinzento. As barras a vermelho indicam o caminho crítico, enquanto que as barras a verde representam atividades não pertencentes ao caminho crítico.

2.6 Validação do modelo

Para validar os resultados, tanto na função objetivo como nas restrições, substituímos os valores das variáveis de decisão pelo valor que estas tomam na solução que o `lp_solve` indica como ótima. A ideia é verificar que os valores das variáveis de decisão obtidos confirmam o valor da função objetivo obedecendo a todas as restrições.

Para evitar ao máximo o erro humano, a substituição de variáveis foi feita recorrendo a ferramentas que auxiliaram a substituição automática das variáveis pelo seu valor.

2.6.1 Variáveis de decisão

No resultado obtido, todas as variáveis tomam um valor maior ou igual a 0, tal como seria de esperar.

2.6.2 Função objetivo

Neste modelo a função objetivo consiste apenas no valor de uma variável, T_{fim} , que vale 26 unidades de tempo na solução ótima. Por motivos que não fazem parte do âmbito deste trabalho, o valor esperado para a duração mínima do projeto deverá ser o mesmo valor de duração encontrado no caminho crítico da Parte I. O valor obtido corresponde de facto ao esperado, uma vez que a duração do caminho crítico obtido na Parte I foi também igual a 26 unidades de tempo.

2.6.3 Restrições

```
//Nodo Inicial
Tini >= 0 + 0;
0 >= 0 + 0;

//Nodo 0
T0 >= Tini + 0;
0 >= 0 + 0;

//Nodo 1
T1 >= T0 + 4;
4 >= 0 + 4;

//Nodo 3
T3 >= T1 + 6;
24 >= 4 + 6;

T3 >= T5 + 4;
24 >= 20 + 4;

T3 >= T4 + 9;
24 >= 11 + 9;

//Nodo 4
T4 >= T0 + 4;
11 >= 0 + 4;

T4 >= T7 + 6;
11 >= 5 + 6;

//Nodo 5
T5 >= T4 + 9;
20 >= 11 + 9;
```

```

T5 >= T7 + 6;
20 >= 5 + 6;

T5 >= T10 + 8;
20 >= 5 + 8;

//Nodo 6
T6 >= Tini + 0;
0 >= 0 + 0;

//Nodo 7
T7 >= T6 + 5;
5 >= 0 + 5;

//Nodo 9
T9 >= T7 + 6;
20 >= 5 + 6;

T9 >= T11 + 7;
20 >= 13 + 7;

T9 >= T10 + 8;
20 >= 5 + 8;

//Nodo 10
T10 >= T6 + 5;
5 >= 0 + 5;

//Nodo 11
T11 >= T10 + 8;
13 >= 5 + 8;

//Nodo final
Tfim >= T3 + 2;
26 >= 24 + 2;

Tfim >= T5 + 4;
26 >= 20 + 4;

Tfim >= T9 + 2;
26 >= 20 + 2;

```

Assim conclui-se que todas as restrições são respeitadas.

2.7 Atividade no caminho crítico

Considere-se a atividade 7, que faz parte do caminho crítico. Por análise do resultado do modelo verifica-se que esta atividade deve começar em $T = 5 \text{ u.t.}$. Como é facilmente visível no Diagrama de *Gantt* apresentado na figura 2.2, caso a atividade comece um pouco mais tarde, irá atrasar o começo da atividade que lhe sucede, neste caso a atividade 4. Se a atividade 4 começar mais tarde, por sua vez vai atrasar o início da atividade 5, que lhe sucede. Seguindo a mesma lógica a atividade 5 começando mais tarde atrasa o início da atividade 3. Visto a atividade 3 ser a última atividade a ser realizada, tal atraso implica um atraso no tempo de execução total do projeto.

Obtém-se resultados semelhantes caso sejam provocados atrasos em outras atividades pertencentes ao caminho crítico. Com isto mostra-se o impacto que um atraso na execução de uma atividade no caminho crítico tem no projeto total.

2.8 Atividade fora do caminho crítico

Considere-se a atividade 1, não pertencente ao caminho crítico. De acordo com os resultados do modelo, esta atividade deve ter início em $T_1 = 4 \text{ u.t.}$. Por análise do Diagrama de *Gantt* da figura 2.2, verifica-se que no entanto, esta pode começar um pouco mais tarde. A atividade 1 apenas é sucedida pela atividade 3, sendo que a atividade 3 apenas deve ter início em $T_3 = 24 \text{ u.t.}$. Como restrição do modelo, sabemos que a soma do tempo de início de uma atividade com a sua duração não deve ser maior que o tempo de início de qualquer atividade sucessora. No limite, uma atividade acaba imediatamente antes de uma sua sucessora começar. Traduzindo esse facto em termos matemáticos, no caso da atividade 1 temos que:

$$T_1 + D_1 = T_3$$

Onde:

T_1 Tempo de início da atividade 1

D_1 Duração da atividade 1

T_3 Tempo de início da atividade 3

Resolvendo a equação em ordem a T_1 temos:

$$T_1 + 6 = 24 \text{ u.t.} \Leftrightarrow T_1 = 18 \text{ u.t.}$$

Isto significa que no limite, a atividade 1 pode começar em $T_1 = 18 \text{ u.t.}$ sem que haja atrasos no projeto, embora os resultados do modelo sugiram um tempo inicial $T_1 = 4 \text{ u.t.}$

A situação descrita aqui para a atividade 1 acontece também com outras atividades que não pertencem ao caminho crítico. A conclusão que se tira é que pode haver atrasos (dentro de certos limites) nas atividades que não pertencem ao caminho crítico sem que isso tenha influência no tempo total do projeto.

3. Parte 3

3.1 Análise do problema

Na Parte II, a solução do modelo indica o tempo de início de cada atividade, sabendo que todas as atividades demoram um certo tempo a serem executadas. Nesta parte assume-se que é possível reduzir a duração de certas atividades a um dado custo suplementar. Com isto pretende-se determinar quais as atividades em que aplicar reduções de tempo de modo a que a realização do projeto seja efetuada numa duração menor em 3 unidades de tempo relativamente ao previsto na Parte I e Parte II, com um custo suplementar mínimo possível.

3.2 Modelo

3.2.1 Parâmetros

À semelhança da Parte I e Parte II, também aqui se tem como parâmetros do problema a duração de cada atividade e as suas precedências. Os novos parâmetros são os custos suplementares de redução de cada atividade e a quantidade máxima de redução permitida para cada uma, em unidades de tempo.

Embora os custos normais das atividades sejam um parâmetro do problema, ele não foi considerado na construção do modelo visto o objetivo deste problema ser o de decidir que reduções de tempo se aplicam às atividades, a um custo suplementar mínimo. Os únicos custos relevantes neste contexto são por isso os custos suplementar de redução apenas, não sendo necessário considerar os custos normais de cada atividade na construção do modelo. Os custos normais da atividade foram apenas considerados na análise dos resultados para o cálculo do custo total do projeto (ver secção 3.5).

3.2.2 Variáveis de decisão

O modelo desta parte pretende reduzir o tempo de execução do projeto da Parte I em 3 unidades de tempo. Para isso, assume-se que é possível aplicar unidades de redução à duração das atividades, com um custo suplementar associado. É importante saber que reduções se deve aplicar e em que atividades. Por isso, tem-se como variáveis de decisão a redução (em unidades de tempo) a aplicar a cada umas das atividades. Essas reduções serão representadas por R_i . Por exemplo, R_5 representa a redução de tempo efetuada à atividade 5. Tal como nas partes anteriores, nodos finais e iniciais foram considerados como atividades “fictícias”, existindo por isso as variáveis R_{ini} e R_{fim} .

Neste modelo, como a duração de cada atividade pode ser reduzida, o tempo de início das atividades pode também ser alterado. Assim, além das variáveis de decisão apresentadas anteriormente, manteve-se as mesmas variáveis correspondentes aos tempos iniciais das atividades

usadas na resolução do modelo da Parte II. Decidiu-se também manter a mesma nomenclatura para essas variáveis: T_i .

3.2.3 Função Objetivo

Sabendo que o objetivo deste modelo é saber quais as atividades em que se deve reduzir o tempo de duração, a um determinado custo suplementar, a função objetivo deverá ser uma expressão que indique o custo suplementar total inerente à redução de tempos aplicados a estas atividades. Este custo deve tomar o menor valor possível. Trata-se por isso de um problema de minimização.

Os valores das variáveis de decisão R_i indicam a redução (em unidades de tempo) que a atividade i sofreu. Cada atividade tem um custo suplementar de redução associado, por unidade de tempo. Saber o custo suplementar das reduções aplicadas a uma atividade corresponde a multiplicar as unidades de tempo da redução pelo custo suplementar unitário de redução associado a essa atividade. Ou seja, o custo suplementar das reduções à duração de uma atividade, Cr_i é dado por:

$$Cr_i = R_i \times C_i$$

Onde:

R_i Variável de decisão, indica a redução (em unidades de tempo) aplicada à atividade i

C_i Parâmetro do problema, indica o custo de redução (por unidade de tempo) associado à atividade i

O custo suplementar de redução do projeto todo será o somatório dos custos suplementares de cada atividade. Ou seja, a função objetivo, z , pode ser escrita de forma genérica como:

$$\min z = \sum Cr_i$$

Substituindo na fórmula os custos unitários de redução dados no enunciado, fica-se com a seguinte função objetivo:

$$\min z = 100 R_0 + 300 R_1 + 100 R_3 + 400 R_4 + 800 R_5 + 90 R_6 + 500 R_{10} + 300 R_{11}$$

3.2.4 Restrições

Com as restrições pretende-se indicar o espaço de possíveis soluções. Em primeiro lugar, sabe-se que há limites relativamente às unidades de tempo para as reduções à duração que cada atividade pode sofrer. Assim, cada variável de decisão do modelo terá um limite superior. Conforme visto anteriormente, tem-se uma variável de decisão correspondente às unidades de tempo reduzidas em cada atividade. Cada atividade i , gera por isso uma restrição do tipo:

$$R_i \leq M_i$$

Onde:

R_i Variável de decisão que indica a redução (em unidades de tempo) aplicada à atividade i .

M_i Valor máximo de redução que é possível aplicar à atividade i

Analisando as reduções de tempo máximas permitidas para cada atividade, vemos que a atividade zero só pode ser reduzida em 1 unidade de tempo, por exemplo, enquanto que as atividades 7 e 9 não podem ser reduzidas em nenhuma unidade de tempo, pois o valor máximo de redução é zero. Traduzindo esta informação em termos matemáticos temos que $R_0 \leq 2$, $R_7 \leq 0$ e $R_9 \leq 0$. A exceção à regra são as atividades finais e iniciais, que por serem atividades “fictícias” não irão ter qualquer redução de tempo, não gerando por isso nenhuma restrição deste tipo.

Em segundo lugar, tem-se restrições relativamente aos tempos de início de cada atividade, semelhantes às da Parte II. Relembrando as restrições da parte II:

$$T_i \geq T_j + D_j$$

Sendo a atividade i realizada após a atividade j , esta expressão indica que tempo de início da atividade i tem obrigatoriamente de ser maior ou igual à soma do tempo de início da atividade j com a duração da atividade j pois é a atividade j que precede à atividade i . Por exemplo, sendo a atividade 0 aquela que precede a atividade 1, tínhamos que:

$$T_1 \geq T_0 + D_0 \Leftrightarrow T_1 \geq T_0 + 4$$

A diferença neste modelo é que a duração da atividade precedente, a atividade j , pode ter sofrido ou não uma redução no seu tempo de duração. Assim, na expressão é necessário subtrair a redução de tempo à duração inicial da atividade. Fica-se com:

$$T_i \geq T_j + (D_j - R_j)$$

Onde:

T_i Tempo em que a atividade i inicia

T_j Tempo em que a atividade j inicia

D_j Duração da atividade j

R_j Redução de tempo a aplicar na atividade j

Existe uma restrição deste tipo para cada uma das atividades existentes.

Por último, tem-se uma restrição para forçar o tempo do projeto. Sabe-se que o tempo de início da atividade final corresponde à duração do projeto (visto na Parte II). Neste modelo, pretende-se reduzir a duração do projeto em 3 unidades de tempo relativamente ao tempo obtido na Parte I e Parte II. A duração do projeto nessas partes foi de 26 unidades de tempo. Pretende-se por isso que a nova duração seja de $26 - 3 = 23$ unidades de tempo. A duração total do projeto é dada por T_{fim} , tempo inicial da atividade final. Assim, forçar a que o projeto tenha uma duração específica corresponde a forçar essa variável a ter o valor de duração pretendido. Por esse motivo, no modelo tem-se a restrição:

$$T_{fim} = 23.$$

Visto que os tempos e as reduções não podem ser negativos, neste modelo tem-se ainda restrições de não-negatividade:

$$T_i \geq 0; R_i \geq 0, \forall i \in \{ini, 0, 1, 3, 4, 5, 6, 7, 9, 10, 11, fim\}$$

3.3 Ficheiro de *input*

O ficheiro de *input* é constituído pela função objetivo e restrições, detalhadas em secções anteriores.

```
=== FUNCAO OBJECTIVO ===
```

```
min: 100 R0 + 300 R1 + 100 R3 +  
400 R4 + 800 R5 + 90 R6 + 500 R10 + 300 R11;
```

```
=== RESTRICOES ===
```

```
//Nodo inicial  
Rini = 0;
```

```
//Nodo 0  
R0 <= 1;  
//Nodo 1  
R1 <= 2;  
//Nodo 3  
R3 <= 1;  
//Nodo 4  
R4 <= 3;  
//Nodo 5  
R5 <= 1;  
//Nodo 6  
R6 <= 2;  
//Nodo 7  
R7 <= 0;  
//Nodo 9  
R9 <= 0;  
//Nodo 10  
R10 <= 1;  
//Nodo 11  
R11 <= 2;  
//Nodo final  
Rfim <= 0;
```

```
//Nodo Inicial  
Tini >= 0 + 0;  
//Nodo 0  
T0 >= Tini + 0 - Rini;  
//Nodo 1  
T1 >= T0 + 4 - R0;  
//Nodo 3  
T3 >= T1 + 6 - R1;  
T3 >= T5 + 4 - R5;  
T3 >= T4 + 9 - R4;
```

```

//Nodo 4
T4 >= T0 + 4 - R0;
T4 >= T7 + 6 - R7;
//Nodo 5
T5 >= T4 + 9 - R4;
T5 >= T7 + 6 - R7;
T5 >= T10 + 8 - R10;
//Nodo 6
T6 >= Tini + 0 - Rini;
//Nodo 7
T7 >= T6 + 5 - R6;
//Nodo 9
T9 >= T7 + 6 - R7;
T9 >= T11 + 7 - R11;
T9 >= T10 + 8 - R10;
//Nodo 10
T10 >= T6 + 5 - R6;
//Nodo 11
T11 >= T10 + 8 - R10;
//Nodo final
Tfim >= T3 + 2 - R3;
Tfim >= T5 + 4 - R5;
Tfim >= T9 + 2 - R9;

Tfim = 26-3;

```

3.4 *Output* produzido pelo `lp_solve`

O output apresentado a seguir foi obtido por *copy-paste* direto resultante da execução do `lp_solve` num sistema linux para o ficheiro de input apresentado anteriormente:

Value of objective function: 280

Actual values of the variables:

R0	0
R1	0
R3	1
R4	0
R5	0
R6	2
R7	0
R9	0
R10	0
R11	0
Rini	0
Rfim	0

Tini	0
T0	0
T1	4
T3	22
T5	18
T4	9
T7	3
T10	3
T6	0
T9	18
T11	11
Tfim	23

Actual values of the constraints:

R1	0
R2	4
R3	18
R4	4
R5	13
R6	9
R7	6
R8	9
R9	15
R10	15
R11	0
R12	5
R13	15
R14	7
R15	15
R16	5
R17	8
R18	2
R19	5
R20	5

3.5 Resultado

Os resultados do lpsolve mostram que se consegue reduzir a duração total do projeto em 3 unidades de tempo, passando o projeto a demorar 23 unidades de tempo, com um custo suplementar de 280 unidades monetárias.

O custo total do projeto é dado pela soma dos custos normais das atividades com os custos suplementares de redução às suas durações, sendo este último valor o resultado obtido na função objetivo.

O custo normal das atividades é um parâmetro do problema. Somando o custo normal de todas as atividades temos:

$$800 + 900 + 2000 + 1000 + 300 = 5000 \text{ u.m.}$$

O custo total do projeto é por isso $5000 + 280 = 5280 \text{ u.m.}$

Na figura 3.1 mostra-se o caminho crítico obtido bem como as durações das atividades da Parte I.

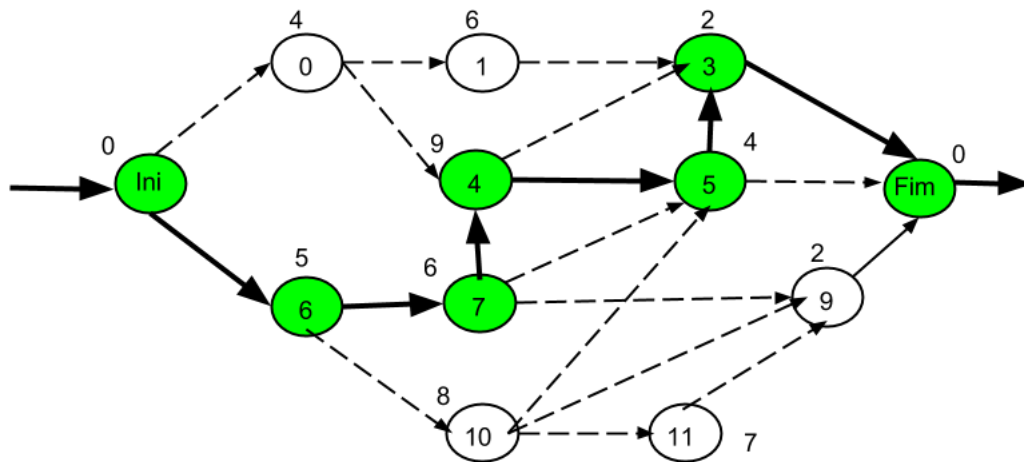


Figura 3.1: Grafo da Parte I, com indicação do caminho crítico e duração (em unidades de tempo) de cada atividade

Agora a duração da atividade 3 é menor em 1 unidade, redução que custa 100 unidades monetárias, e a duração da atividade 6 é menor em duas unidades, redução que custa 180 unidades monetárias, tal como revelado no resultado *output* do *lp_solve*. Na figura 3.2 encontram-se representadas essas reduções.

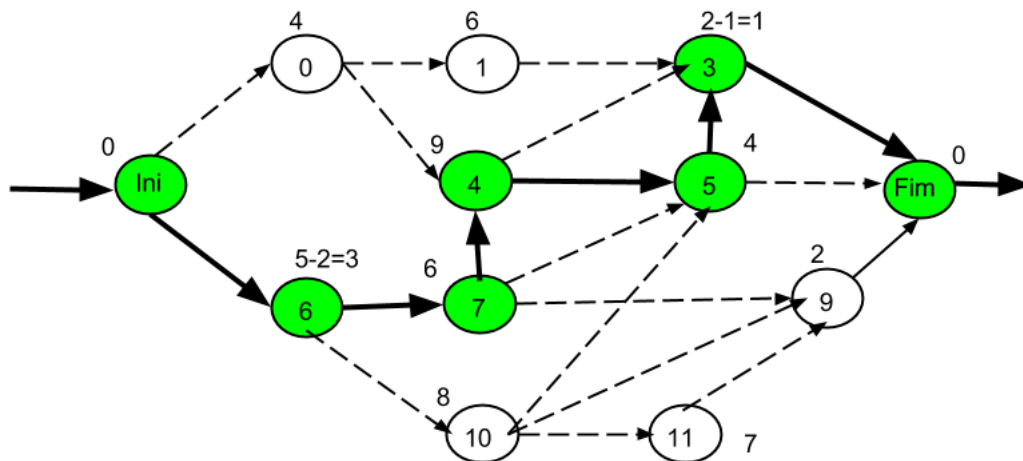


Figura 3.2: Grafo da parte I com as novas durações de cada atividade

Como se obteve reduções de tempo em atividades pertencentes ao caminho crítico da Parte I, conclui-se a solução do modelo da Parte I é única, e que por isso só existia um caminho crítico na rede. Além disso, somando as novas durações das atividades do caminho crítico da Parte I, vemos que o caminho tem a duração de 23 unidades de tempo e o resultado do modelo indica também uma duração total do projeto de 23 unidades de tempo. Ou seja, o caminho crítico da Parte I também continua a ser caminho crítico neste modelo, embora não haja a garantia que seja único.

A figura 3.3 representa o Diagrama de Gantt de acordo com os resultados do modelo. As atividades que sofreram uma redução de tempo estão representadas a amarelo enquanto que as atividades cuja duração não se alterou estão representadas a verde. O caminho crítico está apresentado com preenchimento padrão preto.

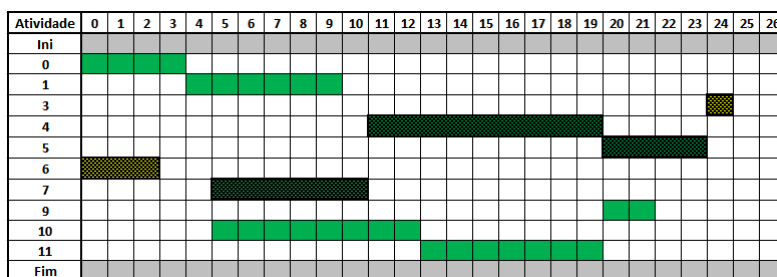


Figura 3.3: Diagrama de Gantt com indicação do caminho crítico

3.6 Validação do modelo

Para validar os resultados, tanto na função objetivo como nas restrições, substituímos os valores das variáveis de decisão pelo valor que estas tomam na solução que o `lp_solve` indica como ótima. A ideia é verificar que os valores das variáveis de decisão obtidos confirmam o valor da função objetivo obedecendo a todas as restrições.

Para evitar ao máximo o erro humano, a substituição de variáveis foi feita recorrendo a ferramentas que auxiliaram a substituição automática das variáveis pelo seu valor.

3.6.1 Variáveis de decisão

No resultado obtido, todas as variáveis tomam um valor maior ou igual a 0, tal como seria de esperar.

3.6.2 Função objetivo

Após a substituição das variáveis pelo valor obtido na solução, a função objetivo fica:

$$100*0+300*0+100*1+400*0+800*0+90*2+500*0+300*0 = 100*1+90*2 = 280$$

Logo confirma-se que o custo total suplementar é de 280 unidades monetárias, tal como indica a solução obtida com o *lp_solve*.

3.6.3 Restrições

```
//Nodo inicial  
Rini = 0;  
0 = 0;
```

```
//Nodo 0  
R0 <= 1;  
0 <= 1;
```

```
//Nodo 1  
R1 <= 2;  
0 <= 2;
```

```
//Nodo 3  
R3 <= 1;  
1 <= 1;
```

```
//Nodo 4  
R4 <= 3;  
0 <= 3;
```

```
//Nodo 5  
R5 <= 1;  
0 <= 1;
```

```
//Nodo 6  
R6 <= 2;  
2 <= 2;
```

```
//Nodo 7
```

```

R7 <= 0;
0 <= 0;

//Nodo 9
R9 <= 0;
0 <= 0;

//Nodo 10
R10 <= 1;
0 <= 1;

//Nodo 11
R11 <= 2;
1 <= 2;

//Nodo final
Rfim <= 0;
0 <= 0;

//Nodo Inicial
Tini >= 0 + 0;
0 >= 0 + 0;

//Nodo 0
T0 >= Tini + 0 - Rini;
0 >= 0 + 0 - 0;

//Nodo 1
T1 >= T0 + 4 - R0;
4 >= 0 + 4 - 0;

//Nodo 3
T3 >= T1 + 6 - R1;
22 >= 4 + 6 - 0;

T3 >= T5 + 4 - R5;
22 >= 18 + 4 - 0;

T3 >= T4 + 9 - R4;
22 >= 9 + 9 - 0;

//Nodo 4
T4 >= T0 + 4 - R0;
9 >= 0 + 4 - 0;

T4 >= T7 + 6 - R7;
9 >= 3 + 6 - 0;

```

```

//Nodo 5
T5 >= T4 + 9 - R4;
18 >= 9 + 9 - 0;

T5 >= T7 + 6 - R7;
18 >= 3 + 6 - 0;

T5 >= T10 + 8 - R10;
18 >= 40 + 8 - 0;

//Nodo 6
T6 >= Tini + 0 - Rini;
0 >= 0 + 0 - 0;

//Nodo 7
T7 >= T6 + 5 - R6;
3 >= 0 + 5 - 2;

//Nodo 9
T9 >= T7 + 6 - R7;
18 >= 3 + 6 - 0;

T9 >= T11 + 7 - R11;
18 >= 41 + 7 - 1;

T9 >= T10 + 8 - R10;
18 >= 40 + 8 - 0;

//Nodo 10
T10 >= T6 + 5 - R6;
40 >= 0 + 5 - 2;

//Nodo 11
T11 >= T10 + 8 - R10;
41 >= 40 + 8 - 0;

//Nodo final
Tfim >= T3 + 2 - R3;
23 >= 22 + 2 - 1;

Tfim >= T5 + 4 - R5;
23 >= 18 + 4 - 0;

Tfim >= T9 + 2 - R9;
23 >= 18 + 2 - 0;

Tfim = 23;

```


$$23 = 23;$$

Assim conclui-se que todas as restrições são respeitadas.

4. Parte 4

4.1 Análise do problema

O problema deste capítulo baseia-se no modelo apresentado no Capítulo anterior, e pretende-se encontrar todas as soluções possíveis, a partir da duração inicial do projeto com várias reduções, de forma a poder relacionar os custos de redução com o custo do projeto. Desta forma, é possível fazer um balanço de quanto é que o projeto se pode reduzir e o seu custo de redução. Ou seja, se os custos de redução justificam a redução pretendida, qual é o máximo que se pode reduzir e quanto custa.

4.2 Modelo

4.2.1 Parâmetros

À imagem do capítulo anterior, considerou-se a duração de cada atividade e a suas precedências como parâmetros, bem como os custos de redução e normais daquelas. Ao contrário das partes anteriores, usou-se os custos normais na construção do modelo. A inclusão deste custos no modelo nesta parte justifica-se pela conveniência no cálculo dos custos totais em cada unidade que se pretende decrementar, que está explicado sucintamente mais abaixo neste capítulo.

4.2.2 Variáveis de decisão

As variáveis de decisão são as mesmas que as descritas no Capítulo 3, secção 3.2.2. Ou seja, pretende-se decidir qual as reduções necessárias em cada atividade, com o menor custo total. As reduções são descritas como anteriormente:

$$R_i, \forall i \in \{ini, 0, 1, 3, 4, 5, 6, 7, 9, 10, 11, fim\}$$

De igual modo, se inclui os tempos de inicio de atividade, à semelhança do capítulo anterior:

$$T_i, \forall i \in \{ini, 0, 1, 3, 4, 5, 6, 7, 9, 10, 11, fim\}$$

O significado destas últimas variáveis mantém-se (ver Capítulo 3, secção 3.2.2).

4.2.3 Função Objetivo

A função objetivo sofreu uma modificação em relação ao modelo do capítulo anterior. Nesta parte, acrescentou-se o custo normal da cada atividade à função objetivo. Assim, a função objetivo agora não representa o custo total suplementar das reduções e passa a representar o

custo total do projeto, que engloba os custos normais e os custos suplementares. Continua-se a querer minimizar os custos de redução de forma ótima, apenas se inclui o custo normal da cada atividade. Note-se que, a soma de todos os custos normais não influencia o valor da variável de decisão (ao contrário dos custos de redução), uma vez que todas as atividades têm que ser realizadas e os seus custos normais são fixos. Assim, o valor da função objetivo permite saber o custo total (custos normais mais custos de redução) para redução de tempo que se pretende efetuar ao projeto.

A função objetivo, para cada atividade $i \in \{ini, 0, 1, 3, 4, 5, 6, 7, 9, 10, 11, fim\}$, é que figura a seguir:

$$\min z = \sum C_{\text{Normal } i} + C_{\text{Redução } i} \times R_i$$

Onde:

- $C_{\text{Normal } i}$ — Custo normal da atividade i ;
- $C_{\text{Redução } i}$ — Custo de redução da atividade i ;
- R_i — Redução de tempo da atividade i

Concretizando os valores na função:

$\min z :$	400	+100 R_0	+1000	+300 R_1
	+300	+100 R_3	+2000	+400 R_4
	+1000	+800 R_5	+800	+90 R_6
	+900	+0 R_7	+300	+0 R_9
	+1600	+500 R_{10}	+1400	+300 R_{11}

4.2.4 Restrições

As restrições são iguais às da Parte III, com a exceção da restrição do tempo final do projeto, onde, ao invés de subtrair 3 unidades de tempo ao tempo do caminho crítico do projeto, considerou-se uma redução gradual (de 1 unidade de tempo) em cada execução do modelo no `lp_solve`, de forma a obter um custo total para cada valor de redução, e por isso de cada tempo total de execução do projeto. O ponto de partida foi o tempo do projeto obtido na Parte I, de 26 unidades de tempo e a partir daí analisou-se todas as possibilidades redução de tempo até uma redução máxima de 26 unidades de tempo, que corresponde a uma duração nula do projeto. Reduções muito elevadas são impossíveis como seria de esperar, no entanto por uma questão de completude, consideraram-se todas as possibilidades. Um *script* em BASH foi criado para o efeito, para executar todos os modelos. Assim sendo, para cada execução do modelo a restrição do tempo final fica:

$$T_{fim} = T_{\text{caminho crítico}} - R_{\text{total desejada}}$$

- T_{fim} — Tempo final;
- $T_{\text{caminho crítico}}$ — Tempo total do caminho crítico;

- $R_{\text{total desejada}}$ — Redução total de tempo desejada no projeto, em cada execução do modelo no `lp_solve`.

Assim:

$$\begin{aligned} T_{fim} &= 26 - 0 = 26 \\ T_{fim} &= 26 - 1 = 25 \\ T_{fim} &= 26 - 2 = 24 \\ T_{fim} &= 26 - 3 = 23 \\ &\dots \\ T_{fim} &= 26 - 26 = 0 \end{aligned}$$

4.3 *Input do script em BASH*

O *script* para a execução de cada redução manipula o ficheiro do modelo do `lp_solve`, removendo a última linha onde se encontra a última restrição, para em seguida adicionar uma nova linha com uma nova restrição com o novo valor a reduzir. O *output* apresentado corresponde à primeira linha de cada execução do modelo. O código do *script* foi o seguinte:

```
#!/bin/bash

#Nome do ficheiro passado como parâmetro de entrada
file="$1"

#Valor do tempo do caminho crítico
CAM_CRIT=26

#Redução em cada execução
REDUCAO=0
for i in $(seq $CAM_CRIT -1 0)
do
    #Formata string com última restrição com valor da redução
    var='Tfim ='$i';'
    #Elimina a última string do ficheiro
    sed -i '$ d' $file
    #Acrescenta string formatada ao final do ficheiro
    echo $var >> $file
    #Guarda 1ª linha do output de cada execução
    out=$(lp_solve $file | head -n 2)
    #Apresenta resultado no stdin
    echo Redução de $REDUCAO : $out
    REDUCAO=$((REDUCAO + 1))
done
```

4.4 Resultado

Os valores do *output* da execução do *script* figuram a seguir:

```
Redução de 0 (Tfim = 26) : Value of objective function: 9700
Redução de 1 (Tfim = 25) : Value of objective function: 9790
Redução de 2 (Tfim = 24) : Value of objective function: 9880
Redução de 3 (Tfim = 23) : Value of objective function: 9980
Redução de 4 (Tfim = 22) : Value of objective function: 10380
Redução de 5 (Tfim = 21) : Value of objective function: 10780
Redução de 6 (Tfim = 20) : Value of objective function: 11180
Redução de 7 (Tfim = 19) : Value of objective function: 12280
Redução de 8 (Tfim = 18) : This problem is infeasible
```

Como se pode observar, a partir da redução de 8 unidades de tempo, o `lp_solve` não consegue resolver o modelo, e portanto o output do script a partir daí foi omitido. Logo a redução máxima possível é de 7 unidades de tempo, que representa realizar o projeto em 19 unidades de tempo, com o custo total de 12280 unidades monetárias.

A tabela 4.1 associa as reduções possíveis ao tempo total do projeto e respectivo custo total de execução do mesmo.

Redução	Tempo Total	Custos Totais
0	26	9700
1	25	9790
2	24	9880
3	23	9980
4	22	10380
5	21	10780
6	20	11180
7	19	12280
8	18	∞

Tabela 4.1: Reduções em cada execução do modelo no `lp_solve`

O gráfico 4.1 foi elaborado com base na tabela 4.1 e mostra o custo total do projeto para cada valor de redução de tempo possíveis.

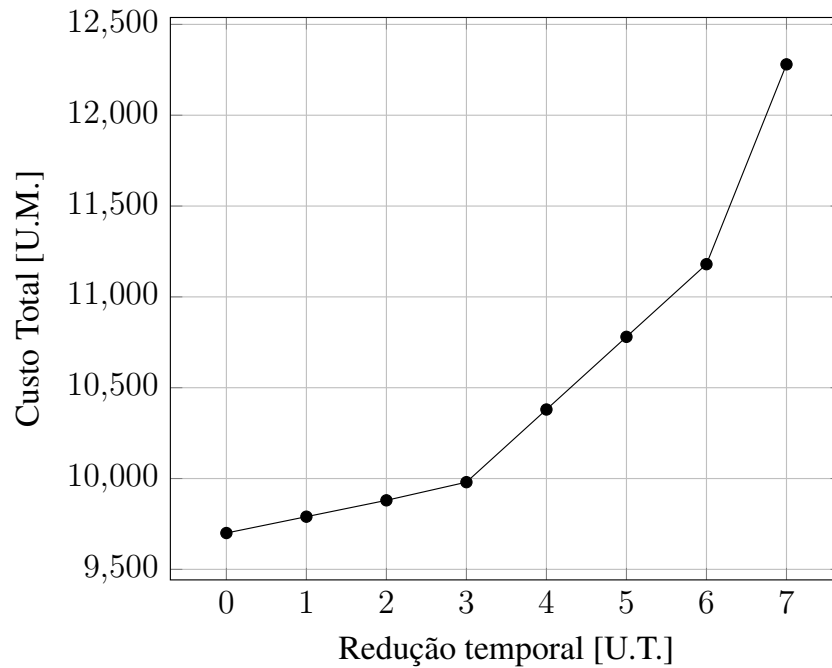


Figura 4.1: Gráfico do custo total do projeto em função das reduções de tempo

O gráfico 4.2 mostra o custo total do projeto para cada valor de duração do projeto.

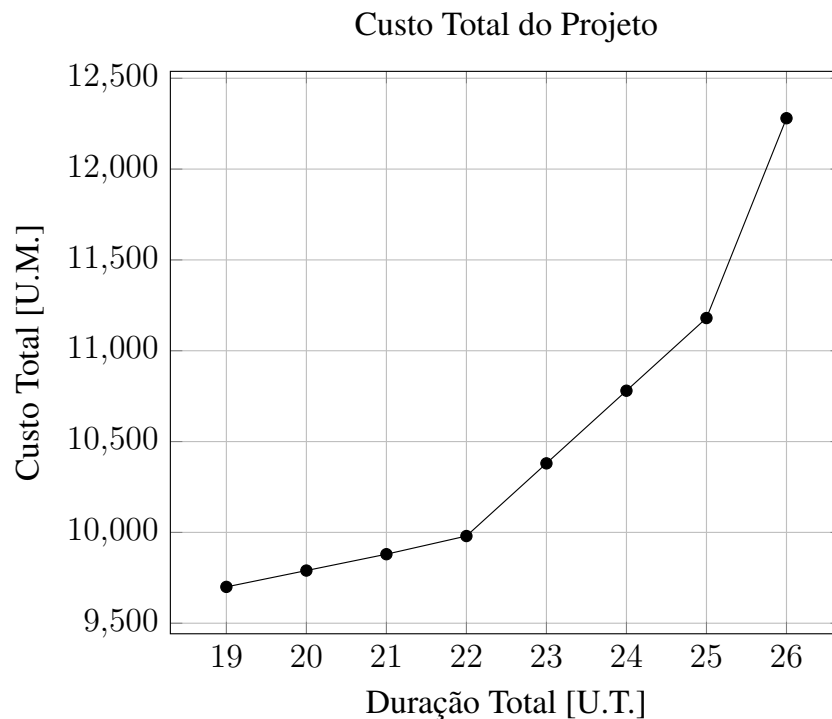


Figura 4.2: Gráfico do custo total do projeto em função da duração do projeto

Foram omitidos do gráfico reduções de tempo para as quais não existe solução do modelo.

Os resultados mostram que a duração projeto pode ser reduzida no máximo em 7 unidades de tempo (duração total de 19 u.t) com um custo total de 12280 unidades monetárias. De igual modo, entre a duração do projeto sem reduções e redução máxima existem mais 6 possíveis reduções, com um custo entre 9790 U.T. e 11180 U.M. (não incluindo a máxima redução).

Através do gráfico 4.2 é possível perceber os vários compromissos entre a duração total do projeto e o custo total. Nomeadamente, o gráfico sugere que para durações do projeto próximas da inicial (26 unidades de tempo), cada redução adicional não acarreta um aumento nos custos significativo. No entanto, tal já não acontece para durações do projeto mais pequenas, onde cada redução adicional tem custos suplementares elevados. Esse fator pode ser visto no gráfico pelos declives das retas que unem os vários pontos, que vai aumentando, até ao ponto em que o problema é impossível e não é possível terminar o projeto num tempo menor.

5. Parte 5

5.1 Análise do problema

Nesta parte do trabalho pretende-se analisar um cenário semelhante à Parte III, em que se pretende reduzir o tempo total do projeto considerando que é possível reduzir a duração de algumas atividades com um custo suplementar. Na Parte III, o custo suplementar de redução era linear. Nesta parte analisa-se a possibilidade de cada atividade poder ter custos não-lineares quando se reduz a sua duração. Para isso, considera-se uma aproximação a uma função não-linear em que se usam duas funções lineares definidas por ramos. Na prática, corresponde a dizer que cada atividade pode estar sujeita a duas reduções de tipos diferentes, em que cada tipo de redução tem um custo associado. Estes dois tipos de redução serão referidos daqui em diante como reduções do tipo 1 e reduções do tipo 2. Como se trata de uma aproximação a uma função não-linear a partir de duas funções, as reduções do tipo 1 deverão ser aplicadas sempre primeiro que as reduções do tipo 2.

5.2 Modelo

5.2.1 Parâmetros

Nesta parte os parâmetros do problema são as precedências das atividades, as suas durações, o custo de ambos os tipos de redução à duração que se pode fazer e os seus respetivos custos suplementares. Em adição, há limites máximos (em unidades de tempo) para cada tipo de redução possível, sendo esses limites também considerados como parâmetros do problema.

Embora os custos normais das atividades sejam um parâmetro do problema, ele não foi considerado na construção do modelo visto o objetivo deste problema ser o de decidir que reduções de tempo se aplicam às atividades, a um custo suplementar mínimo. Os únicos custos relevantes neste contexto são por isso os custos suplementar de redução apenas, não sendo necessário considerar os custos normais de cada atividade na construção do modelo. Os custos normais da atividade foram apenas considerados na análise dos resultados para o cálculo do custo total do projeto (ver secção 5.5).

5.2.2 Variáveis de decisão

À semelhança das partes anteriores, também aqui se usou o tempo de início de cada atividade como variáveis de decisão, com a mesma nomenclatura apresentada anteriormente.

Na Parte III introduziram-se variáveis correspondentes às unidades de tempo a reduzir à duração de cada atividade. Nesta parte, cada atividade tem agora associados dois tipos de reduções possíveis. Cada atividade passa agora a ter associadas duas variáveis de decisão, que traduzem as reduções (em unidades de tempo) do tipo 1 e do tipo 2 a aplicar às durações. A

nomenclatura das variáveis para as reduções da Parte III teve que ser alterada aqui, para traduzir precisamente o facto de cada atividade poder ter agora dois tipos de redução possíveis, em vez de apenas 1. As novas variáveis apresentam o seguinte formato:

$$R_{i_j}$$

Onde:

i Número da atividade

j Tipo de redução a aplicar (1 ou 2)

Assim, R_{5_2} representa a redução (em unidades de tempo) do tipo 2 a aplicar à duração da atividade 5.

5.2.3 Função objetivo

A função objetivo deste modelo é uma expressão que representa o custo total suplementar de aplicar reduções de tipo 1 e tipo 2 às durações de todas as atividades.

Na Parte III viu-se que saber o custo suplementar de redução para uma atividade correspondia multiplicar o custo de redução/unidade de tempo pelas unidades de tempo de redução que a atividade sofreu. Nesta parte, há dois tipos de reduções, pelo que para uma atividade i o seu custo suplementar de redução total, C_i , será dado por:

$$C_i = C_{i_1} \times R_{i_1} + C_{i_2} \times R_{i_2}$$

Onde:

C_{i_1} Custo por unidade de tempo associada a reduções do tipo 1 na duração da atividade i

R_{i_1} Unidades de tempo reduzir à duração da atividade i a custo C_{i_1}

C_{i_2} Custo por unidade de tempo associada a reduções do tipo 2 na duração da atividade i

R_{i_2} Unidades de tempo reduzir à duração da atividade i a custo C_{i_2}

A função objetivo, z , corresponde ao somatório dos custos de redução de todas as atividades, ou seja:

$$\min z = \sum C_i$$

Substituindo os custos das reduções de cada tipo pelo valor dado no enunciado, ficamos com a seguinte função objetivo:

$$\begin{aligned} \min: & 100 R_{0_1} + 200 R_{0_2} + 300 R_{1_1} + 600 R_{1_2} + 100 R_{3_1} + 200 R_{3_2} + \\ & 400 R_{4_1} + 800 R_{4_2} + 800 R_{5_1} + 1600 R_{5_2} + 90 R_{6_1} + 180 R_{6_2} + \\ & 0 R_{7_1} + 0 R_{7_2} + 0 R_{9_1} + 0 R_{9_2} + 500 R_{10_1} + 1000 R_{10_2} + 300 R_{11_1} \\ & + 600 R_{11_2}; \end{aligned}$$

5.2.4 Restrições

Também as restrições serão muito semelhantes às da Parte III. Em primeiro lugar, mantêm-se as restrições referentes às unidades de tempo máximas que se pode reduzir. Assim cada variável de decisão relativa às reduções a aplicar, terá uma restrição a limitar o seu valor máximo. Visto cada atividade poder estar sujeita a 2 tipos de redução, por cada atividade ter-se-á então 2 restrições relativas aos limites máximos de redução. Em termos genéricos, para a atividade i , essas duas restrições podem ser escritas como:

$$R_{i_1} \leq Rmax_{i_1}; R_{i_2} \leq Rmax_{i_2}$$

Onde:

R_{i_1} Variável de decisão — redução (em unidades de tempo) do tipo 1 a aplicar à atividade i

R_{i_2} Variável de decisão — redução (em unidades de tempo) do tipo 2 a aplicar à atividade i

$Rmax_{i_1}$ Redução máxima de tempo do tipo 1 que é possível aplicar à atividade i

$Rmax_{i_2}$ Redução máxima de tempo do tipo 2 que é possível aplicar à atividade i

Por exemplo, para a atividade 5 tem-se as seguintes restrições:

$$R5_1 \leq 0.5;$$

$$R5_2 \leq 0.5;$$

Tal como na Parte III, também as restrições relativas aos tempos de início de cada atividade estão presentes neste modelo. Recorde-se que na Parte III tais restrições tinham o seguinte aspecto:

$$T_i \leq T_j + (D_j - R_j)$$

T_i Tempo em que a atividade i inicia

T_j Tempo em que a atividade j inicia

D_j Duração da atividade j

R_j Redução de tempo a aplicar na atividade j

A inequação apresentada em cima parte tem como pressuposto a atividade j preceder a atividade i . A diferença introduzida no modelo desta parte é que cada atividade pode agora ter dois tipos de redução possíveis, pelo que as restrições têm que contar com esse novo fator. Nas restrições da Parte III, o facto de cada atividade poder ver reduzida a sua duração está patente na parcela $(D_j - R_j)$ no segundo membro da inequação. Seguindo uma lógica semelhante, dizer que uma atividade pode contar com dois tipos de redução, corresponde a dizer a subtrair uma nova redução à duração. Ou seja:

$$T_i \leq T_j + (D_j - R_{j_1} - R_{j_2})$$

R_{j_1} Redução (em unidades de tempo) do tipo 1 a aplicar na atividade j

R_{j_2} Redução (em unidades de tempo) do tipo 2 a aplicar na atividade j

Neste modelo era também necessário forçar um tempo de duração total do projeto. Tal como na Parte III, isso foi feito forçando a variável T_{fim} a tomar um valor concreto. Neste modelo usou-se a mesma ideia. Pedia-se que o tempo total do projeto neste modelo fosse inferior em 4 unidades de tempo ao obtido na Parte I. Na Parte I, o tempo total do projeto foi de 26 unidades de tempo. Assim, a restrição que força o tempo total do projeto é:

$$T_{fim} = 26 - 4;$$

Visto que os tempos e as reduções não podem ser negativos, neste modelo tem-se ainda restrições de não-negatividade:

$$T_i \geq 0; R_{i_1} \geq 0; R_{i_2} \geq 0; \forall i \in \{ini, 0, 1, 3, 4, 5, 6, 7, 9, 10, 11, fim\}$$

5.3 Ficheiro *Input*

```
/*
=== FUNCAO OBJECTIVO ===
Minimizar os custos suplementares de redução das atividades.
*/
min: 100 R0_1 + 200 R0_2 +
300 R1_1 + 600 R1_2 +
100 R3_1 + 200 R3_2 +
400 R4_1 + 800 R4_2 +
800 R5_1 + 1600 R5_2 +
90 R6_1 + 180 R6_2 +
0 R7_1 + 0 R7_2 +
0 R9_1 + 0 R9_2 +
500 R10_1 + 1000 R10_2 +
300 R11_1 + 600 R11_2;

/*
Reducoes maximas
*/
R0_1 <= 0.5;
R0_2 <= 0.5;

R1_1 <= 1;
R1_2 <= 1;

R3_1 <= 0.5;
R3_2 <= 0.5;

R4_1 <= 1;
R4_2 <= 1;

R5_1 <= 0.5;
R5_2 <= 0.5;
```

```

R6_1 <= 1;
R6_2 <= 1;

R7_1 <= 0;
R7_2 <= 0;

R9_1 <= 0;
R9_2 <= 0;

R10_1 <= 0.5;
R10_2 <= 0.5;

R11_1 <= 1;
R11_2 <= 1;

//Nodo Inicial
Tini >= 0 + 0;
//Nodo 0
T0 >= Tini + 0 - Rini;
//Nodo 1
T1 >= T0 + 4 - R0_1 - R0_2;
//Nodo 3
T3 >= T1 + 6 - R1_1 - R1_2;
T3 >= T5 + 4 - R5_1 - R5_2;
T3 >= T4 + 9 - R4_1 - R4_2;
//Nodo 4
T4 >= T0 + 4 - R0_1 - R0_2;
T4 >= T7 + 6 - R7_1 - R7_2;
//Nodo 5
T5 >= T4 + 9 - R4_1 - R4_2;
T5 >= T7 + 6 - R7_1 - R7_2;
T5 >= T10 + 8 - R10_1 - R10_2;
//Nodo 6
T6 >= Tini + 0 - Rini;
//Nodo 7
T7 >= T6 + 5 - R6_1 - R6_2;
//Nodo 9
T9 >= T7 + 6 - R7_1 - R7_2;
T9 >= T11 + 7 - R11_1 - R11_2;
T9 >= T10 + 8 - R10_1 - R10_2;
//Nodo 10
T10 >= T6 + 5 - R6_1 - R6_2;
//Nodo 11
T11 >= T10 + 8 - R10_1 - R10_2;
//Nodo final
Tfim >= T3 + 2 - R3_1 - R3_2;

```

```

Tfim >= T5 + 4 - R5_1 - R5_2;
Tfim >= T9 + 2 - R9_1 - R9_2;

/*Forcar reducao de tempo*/
Tfim = 26-4;

```

5.4 *Output* produzido pelo `lp_solve`

Value of objective function: 820

Actual values of the variables:

R0_1	0
R0_2	0
R1_1	0
R1_2	0
R3_1	0.5
R3_2	0.5
R4_1	1
R4_2	0
R5_1	0
R5_2	0
R6_1	1
R6_2	1
R7_1	0
R7_2	0
R9_1	0
R9_2	0
R10_1	0
R10_2	0
R11_1	0
R11_2	0
Tini	0
T0	0
Rini	0
T1	4
T3	21
T5	17
T4	9
T7	3
T10	3
T6	0
T9	18
T11	11
Tfim	22

5.5 Resultado

Os resultados do lpsolve mostram que se consegue reduzir a duração total do projeto em 4 unidades de tempo, passando o projeto a demorar 22 unidades de tempo, com um custo suplementar de 820 unidades monetárias.

O custo total do projeto é dado pela soma dos custos normais das atividades com os custos suplementares de redução às suas durações, sendo este último valor o resultado obtido na função objetivo. O custo normal das atividades é:

$$400 + 1000 + 300 + 2000 + 1000 + 800 + 900 + 300 + 1600 + 1400 = 9700 \text{ u.m.}$$

O custo total do projeto é dado pela soma dos custos normais das atividades com os custos suplementares de redução às suas durações, sendo este último valor o resultado obtido na função objetivo. Assim, o projeto tem a duração de 22 unidades de tempo, com um custo total de $9700 + 820 = 10520 \text{ u.m.}$

O grafo da figura 5.1 mostra os novos tempos iniciais de cada atividade de acordo com o resultado obtido no lpsolve:

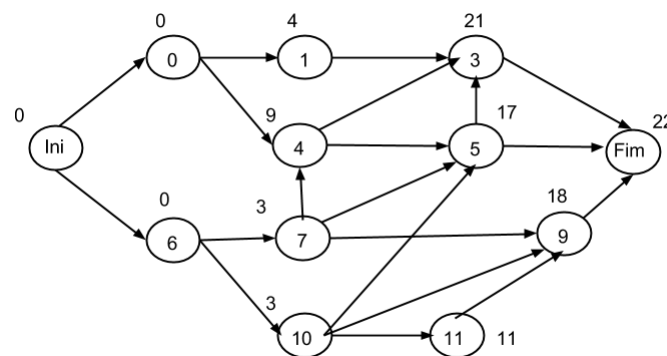


Figura 5.1: Grafo com tempos de início das atividades

A figura 5.2 apresenta o mesmo grafo, com indicação das reduções feitas às atividades e respectivas durações. Os números a roxo indicam as reduções de tipo 1 e os números a verde as reduções de tipo 2.

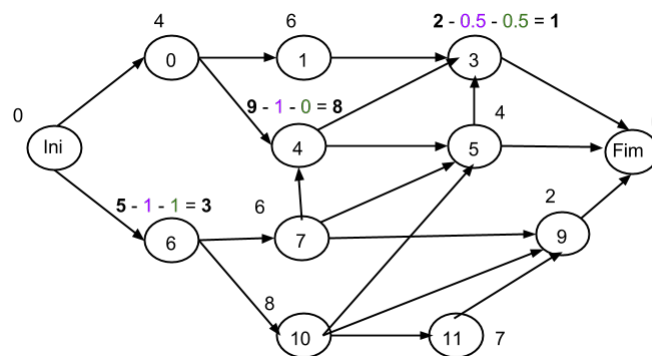


Figura 5.2: Grafo com reduções de tempo às atividades

O Diagrama de Gantt correspondente à execução do projeto encontra-se na figura 5.3:

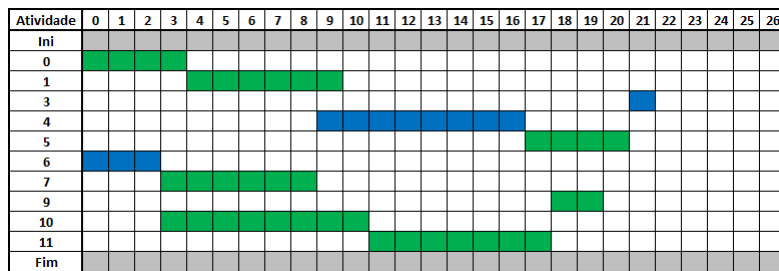


Figura 5.3: Grafo com reduções de tempo às atividades

As atividades que sofreram redução encontram-se marcadas a azul.

É interessante também notar que a duração total do projeto é de 22 unidades de tempo, no entanto, se somarmos as novas durações das atividades que na Parte I faziam parte do caminho crítico tem-se uma duração de $3 + 6 + 8 + 1 = 18$ unidades de tempo. Visto que a duração do projeto não é a mesma do caminho crítico da Parte I, podemos concluir que o caminho crítico se alterou com as reduções aplicadas.

5.6 Validação do Modelo

Para validar os resultados, tanto na função objetivo como nas restrições, substituímos os valores das variáveis de decisão pelo valor que estas tomam na solução que o *lp_solve* indica como ótima. A ideia é verificar que os valores das variáveis de decisão obtidos confirmam o valor da função objetivo obedecendo a todas as restrições.

Para evitar ao máximo o erro humano, a substituição de variáveis foi feita recorrendo a ferramentas que auxiliaram a substituição automática das variáveis pelo seu valor.

5.6.1 Variáveis de Decisão

No resultado obtido, todas as variáveis tomam um valor maior ou igual a 0, tal como seria de esperar.

5.6.2 Função objetivo

Depois da substituição das variáveis pelo seu valor, a função objetivo fica:

$$100 * 0 + 200 * 0 + 300 * 0 + 600 * 0 + 100 * 0.5 + 200 * 0.5 + 400 * 1 + 800 * 0 + 800 * 0 + 1600 * 0 + 90 * 1 + 180 * 1 + 0 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 500 * 0 + 1000 * 0 + 300 * 0 + 600 * 0$$

Inserindo a expressão numa calculadora verifica-se que a expressão é igual a 820, o que confirma o resultado obtido com o *lp_solve*.

5.6.3 Restrições

```
/*
Reducoes maximas
*/
R0_1 <= 0.5;
```

R0_2 <= 0.5;

0 <= 0.5;

0 <= 0.5;

R1_1 <= 1;

R1_2 <= 1;

0 <= 1;

0 <= 1;

R3_1 <= 0.5;

R3_2 <= 0.5;

0.5 <= 0.5;

0.5 <= 0.5;

R4_1 <= 1;

R4_2 <= 1;

1 <= 1;

0 <= 1;

R5_1 <= 0.5;

R5_2 <= 0.5;

0 <= 0.5;

0 <= 0.5;

R6_1 <= 1;

R6_2 <= 1;

1 <= 1;

1 <= 1;

R7_1 <= 0;

R7_2 <= 0;

0 <= 0;

0 <= 0;

R9_1 <= 0;

R9_2 <= 0;

0 <= 0;

0 <= 0;


```
R10_1 <= 0.5;
R10_2 <= 0.5;
```

```
0 <= 0.5;
0 <= 0.5;
```

```
R11_1 <= 1;
R11_2 <= 1;
```

```
0 <= 1;
0 <= 1;
```

```
//Nodo Inicial
Tini >= 0 + 0;
0 >= 0 + 0;
//Nodo 0
T0 >= Tini + 0 - Rini;
0 >= 0 + 0 - 0;
//Nodo 1
T1 >= T0 + 4 - R0_1 - R0_2;
4 >= 0 + 4 - 0 - 0;
//Nodo 3
T3 >= T1 + 6 - R1_1 - R1_2;
T3 >= T5 + 4 - R5_1 - R5_2;
T3 >= T4 + 9 - R4_1 - R4_2;

21 >= 4 + 6 - 0 - 0;
21 >= 17 + 4 - 0 - 0;
21 >= 9 + 9 - 1 - 0;
//Nodo 4
T4 >= T0 + 4 - R0_1 - R0_2;
T4 >= T7 + 6 - R7_1 - R7_2;
9 >= 0 + 4 - 0 - 0;
9 >= 3 + 6 - 0 - 0;
//Nodo 5
T5 >= T4 + 9 - R4_1 - R4_2;
T5 >= T7 + 6 - R7_1 - R7_2;
T5 >= T10 + 8 - R10_1 - R10_2;
17 >= 9 + 9 - 1 - 0;
17 >= 3 + 6 - 0 - 0;
17 >= 3 + 8 - 0 - 0;
//Nodo 6
T6 >= Tini + 0 - Rini;
```

```

0 >= 0 + 0 - 0;
//Nodo 7
T7 >= T6 + 5 - R6_1 - R6_2;
3 >= 0 + 5 - 1 - 1;
//Nodo 9
T9 >= T7 + 6 - R7_1 - R7_2;
T9 >= T11 + 7 - R11_1 - R11_2;
T9 >= T10 + 8 - R10_1 - R10_2;
18 >= 3 + 6 - 0 - 0;
18 >= 11 + 7 - 0 - 0;
18 >= 3 + 8 - 0 - 0;
//Nodo 10
T10 >= T6 + 5 - R6_1 - R6_2;
3 >= 0 + 5 - 1 - 1;
//Nodo 11
T11 >= T10 + 8 - R10_1 - R10_2;
11 >= 3 + 8 - 0 - 0;
//Nodo final
Tfim >= T3 + 2 - R3_1 - R3_2;
Tfim >= T5 + 4 - R5_1 - R5_2;
Tfim >= T9 + 2 - R9_1 - R9_2;
22 >= 21 + 2 - 0.5 - 0.5;
22 >= 17 + 4 - 0 - 0;
22 >= 18 + 2 - 0 - 0;

/*Forcar reducao de tempo*/
Tfim = 26-4;
22 = 26-4;

```

Assim conclui-se que todas as restrições são respeitadas.