Modelos Determinísticos de Investigação Operacional - Trabalho 3

MIEI - Universidade do Minho

Sofia Santos - A89615
Ema Dias - A89518
Sara Queirós - A89491
Tânia Teixeira - A89613

9 de janeiro de 2021

Formulação do problema

Este problema baseia-se no método do caminho crítico (*critical path method*), um método bastante aplicado em projetos que podem ser decompostos num conjunto de atividades com durações determinísticas e relações de precedência. No nosso problema, é-nos dado um projeto deste tipo com uma dada duração, obtida através do cálculo do caminho crítico. As durações e relações de precedência de cada atividade deste projeto podem ser consultadas na tabela abaixo:

Actividade	Duração	Precedências
0	4	–
1	6	0
2	7	1,4
3	2	2,5
4	9	0,7
5	4	4,8
6	5	–
7	6	6
8	4	7,10
9	2	8,11
10	8	6
11	7	10

Figura 1: Atividades e relações de precedência do projeto.

O nosso objetivo é reduzir a duração do projeto em 3 unidades de tempo, reduzindo as durações das atividades usando os valores de custo por U.T. e de redução máxima da seguinte tabela:

Actividade	Custo Normal	c_1	Máx. red. a custo c_1	c_2	Máx. red. a custo c_2
0	400	200	0,5	100	0,5
1	1000	600	1	300	1
2	1400	1000	3	500	1
3	300	200	0,5	100	0,5
4	2000	800	2	400	1
5	1000	1600	0,5	800	0,5
6	800	180	1	90	1
7	900	–	–	–	–
8	600	200	0,5	100	0,5
9	300	–	–	–	–
10	1600	1000	0,5	500	0,5
11	1400	600	1	300	1

Figura 2: Valores de máxima redução e custo de redução por unidade de tempo para cada atividade.

Por exemplo, podemos reduzir a duração da atividade 4 em 2 U.T. no máximo, a um custo de 800 U.M. por cada U.T. reduzida, e se esta redução for máxima podemos ainda reduzir em mais 1 U.T. no máximo, a um custo de 400 U.M./U.T.. Por outras palavras, apenas podemos aplicar a redução a custo c_2 se a redução aplicada a custo c_1 for máxima.

Para além destas reduções lineares, podemos ainda reduzir a duração da atividade 7 para 5 U.T. com um custo de 300 U.M., ou para 4 U.T. com um custo de 1100 U.T., e podemos também reduzir a duração da atividade 9 para 1 U.T. com um custo de 200 U.M., ou para 0 U.T. com um custo de 400. Ao contrário das outras reduções, estas têm um valor fixo, ou seja, realizam-se na sua totalidade ou não se realizam de todo, e apenas podemos realizar uma por atividade.

Parte 0

De acordo com o maior nº de estudante do nosso grupo (89615), devemos remover as atividades 1 e 5 do nosso projeto, alterando as precedências de modo a que os sucessores da atividade 1 passem a suceder aos antecessores de 1 e os sucessores da atividade 5 aos antecessores de 5. Ficamos assim com a seguinte tabela final para o nosso projeto:

Atividade	Duração	Precedências	Custo Normal	c_1	Máx. red. a custo c_1	c_2	Máx. red. a custo c_2
0	4	-	400	200	0.5	100	0.5
2	7	0,4	1400	1000	3	500	1
3	2	2,4,8	300	200	0.5	100	0.5
4	9	0,7	2000	800	2	400	1
6	5	-	800	180	1	90	1
8	4	7,10	600	200	0.5	100	0.5
10	8	6	1600	1000	0.5	500	0.5
11	7	10	1400	600	1	300	1

Atividade	Duração	Precedências	Custo Normal	c	Redução a custo c
7	6	6	900	300/1100	1/2
9	2	8,11	300	200/400	1/2

Figura 3: Projeto após eliminar as atividades devidas.

O grafo associado a este projeto é:

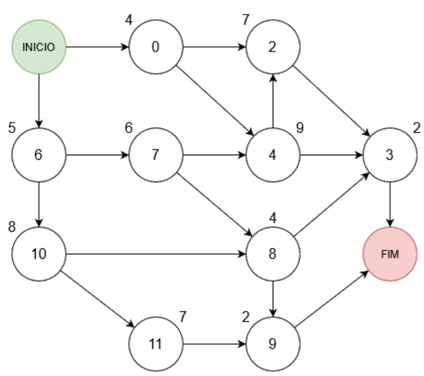


Figura 4: Grafo do projeto, onde as atividades são representadas por nós e as precedências por arcos.

Para calcular a duração do projeto, podemos calcular o caminho crítico. O caminho crítico é ao caminho mais longo entre o vértice inicial do projeto e o vértice final do projecto. Logo, a duração deste caminho corresponderá a duração do projeto. Podemos usar um modelo de programação linear simples para obter este caminho. Para o nosso grafo $G = (V, A)$, com um vértice de entrada s e um vértice de saída t, este modelo pode ser representado um linguagem matemática do seguinte modo:

$$\begin{aligned} \text{max:} \quad & \sum_{(i,j)\in A} c_i x_{ij}, i \neq s \\ \text{suj. a:} \quad & \sum_{(i,j)\in A} x_{ij} = \sum_{(j,k)\in A} x_{jk}, i \neq s \\ & \sum_{i \in V} x_{si} = 1 \\ & x_{ij} \in \{0,1\}, \forall (i,j) \in A \end{aligned}$$

Neste modelo, as variáveis de decisão, x_{ij}, são binárias, e dizem se um arco faz ou não parte do caminho crítico. A função objetivo pretende maximizar a duração do projeto com base nos arcos que são percorridos, os parâmetros são a duração de cada atividade e as restrições garantem que, se houver um arco a entrar num vértice, deve haver um arco a sair desse vértice, por outras palavras, cada vértice deve ter o mesmo número de arcos a entrar e a sair dele, à exceção do vértice s, que deve ter apenas um arco de saída.

Podemos então resolver este modelo usando o *LPSolve*, o que nos dará o caminho crítico.

```
/* Objective function */
max: 4 x02 + 4 x04 + 7 x23 + 2 x3f + 9 x42 + 9 x43 + 5 x67 + 5 x610 + 6 x74 +
6 x78 + 4 x89 + 4 x83 + 2 x9f + 8 x108 + 8 x1011 + 7 x119;

/* Variable bounds */

vertice_i: xi0 + xi6 = 1;
vertice_0: xi0 = x02 + x04;
vertice_2: x02 + x42 = x23;
vertice_3: x23 + x43 + x83 = x3f;
vertice_4: x04 + x74 = x42 + x43;
vertice_6: xi6 = x67 + x610;
vertice_7: x67 = x74 + x78;
vertice_8: x78 + x108 = x83 + x89;
vertice_9: x89 + x119 = x9f;
vertice_10: x610 = x108 + x1011;
vertice_11: x1011 = x119;

bin xi0, xi6, x02, x04, x23, x3f, x42, x43, x67, x610, x74, x78, x89, x83,
x9f, x108, x1011, x119;
```

Figura 5: Ficheiro de *input* no *LPSolve*.

Variables	MILP Feasible	result
	29	29
x23	1	1
x3f	1	1
x42	1	1
x67	1	1
x74	1	1
xi6	1	1
x02	0	0
x04	0	0
x43	0	0
x610	0	0
x78	0	0

Figura 6: *Output* do *LPSolve* para o nosso modelo.

Obtemos assim o nosso caminho crítico, que corresponde às atividades 6, 7, 4, 2 e 3, com uma duração de 29 U.T., a duração total do nosso projeto.

Outra forma de calcular a duração total do projeto seria através de um diagrama de Gantt. Este diagrama diz-nos o tempo de início e de fim de cada atividade. Através de um modelo de programação linear, como o modelo seguinte, cujo objetivo é minimizar o tempo total de execução do projeto, somos capazes de obter estes tempos.

$$\begin{aligned} \text{min:} \quad & t_f \\ \text{suj. a:} \quad & t_j >= t_i + c_i, i \neq s \\ & c_s = 0 \end{aligned}$$

Aqui, t_f é o tempo correspondente ao final da execução da atividade final, ou seja, a duração do projeto, t_i e t_j são os tempos das atividades i e j, tal que $(i, j) \in A$, e c_i é o custo da atividade i.

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

/* Variable bounds */

arco_02: t2>= t0 + 4 ; arco_23: t3>= t2 + 7 ;
arco_s0: t0>= ts + 0 ; arco_04: t4>= t0 + 4 ;
arco_42: t2>= t4 + 9 ; arco_43: t3>= t4 + 9 ;
arco_3t: tt>= t3 + 2 ; arco_s6: t6>= ts + 0 ;
arco_74: t4>= t7 + 6 ; arco_83: t3>= t8 + 4 ;
arco_9t: tt>= t9 + 2 ; arco_67: t7>= t6 + 5 ;
arco_78: t8>= t7 + 6 ; arco_89: t9>= t8 + 4 ;
arco_610: t10>= t6 + 5 ; arco_108: t8>= t10 + 8 ;
arco_119: t9>= t11 + 7 ; arco_1011: t11>= t10 + 8;
```

Figura 7: Ficheiro de *input* no *LPSolve*.

Variables	result
	29
tt	29
t2	20
t0	0
t3	27
ts	0
t4	11
t6	0
t7	5
t8	13
t9	20
t10	5
t11	13

Figura 8: *Output* do *LPSolve* para o nosso modelo.

Através da solução deste modelo vemos que a duração total do projeto é de 29 U.T., o mesmo valor que nos deu o outro modelo, e somos agora capazes de construir um diagrama de Gantt.

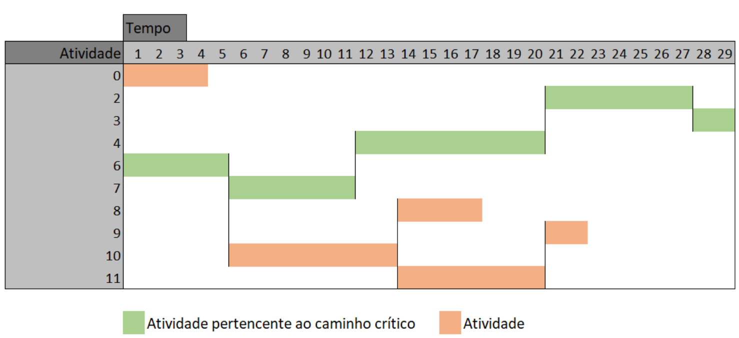


Figura 9: Diagrama de Gantt relativo ao nosso projeto.

Parte I

Para resolver o nosso problema de reduzir a duração total do projeto em 3 U.T., precisamos de criar um modelo com o objetivo de minimizar o custo das reduções que aplicarmos ao projeto. Deste modo, a nossa função objetivo deve ser algo do género:

$$\text{min:} \quad \sum_{i \in V} c_{ri} \times r_i$$

onde c_{ri} é o custo da redução aplicado à atividade i e r_i o valor reduzido.

Porém, como no nosso projeto temos 2 tipos de reduções, fixas nas atividades 7 e 9 e lineares nas restantes, precisamos de uma função objetivo mais complexa. Para as atividades com reduções lineares, esta será a função objetivo para minimizar o custo das reduções, onde r representa a primeira redução e R a segunda, que apenas pode ser aplicada se a primeira for aplicada ao máximo.

$$\text{min:} \quad \sum_{i \in V \setminus \{7,9\}} c_{ri} \times r_i + c_{Ri} \times R_i$$

Para as atividades 7 e 9, a função objetivo é igual, porém aqui os valores de r_i e R_i são binários, ou seja, em vez de representarem o valor reduzido, apenas nos dizem se a redução foi efetuada ou não.

Podemos assim formular a função objetivo do nosso modelo:

$$\text{min:} \quad \sum_{i \in V} c_{ri} \times r_i + c_{Ri} \times R_i$$
$$r_j, R_j \in \{0,1\}, j \in \{7,9\}$$

Devemos incluir no nosso modelo uma restrição do tempo total de execução, para poder responder à nossa questão. Se não restringirmos o tempo máximo do projeto, o resultado do nosso modelo será nulo para 29 U.T., mas não é isso que queremos. Assim, esta restrição:

$$t_t <= 26$$

permite-nos resolver o nosso problema.

Nas atividades com reduções lineares, para realizarmos a segunda redução temos de realizar a primeira redução ao máximo. Essa restrição pode ser representada em programação linear do seguinte modo:

$$\forall i \in V \setminus \{7,9\} : r_i >= m_{ri} - M \times x \wedge R_i <= M(1-x), x \in \{0,1\}$$

Esta restrição, que usa uma variável binária x, diz que, se $x = 1$, então r_i pode tomar qualquer valor dentro dos seus limites. Contudo, R_i terá sempre valor nulo. Por outras palavras, apenas se poderá realizar a primeira restrição. Por outro lado, se $x = 0$, então r_i terá que ter o seu valor máximo, neste caso representado por m_{ri}, e deste modo R_i já não terá que ser nulo, podendo tomar qualquer valor dentro dos seus limites.

Claro que, para esta restrição funcionar, devemos também definir os limites das restrições.

$$\forall i \in V \setminus \{7,9\} : r_i \in [0, m_{ri}] \wedge R_i \in [0, m_{Ri}]$$

onde m_{ri} e m_{Ri} representam os limites máximos das restrições r_i e R_i, respetivamente. Por exemplo, para a atividade 4, $m_{r4} = 2$ e $m_{R4} = 1$.

Para as atividades 7 e 9 as restrições que devemos definir são muito mais simples. Como já definimos que as variáveis associadas a estas devem ser binárias, apenas nos falta definir que apenas poderá ser efetuada no máximo uma delas por atividade. Deste modo, acabamos com esta restrição:

$$\forall i \in \{7,9\} : r_i + R_i <= 1$$

Finalmente, devemos definir as restrições relativas aos arcos, isto é, as relações de precedência. Apenas temos que alterar a restrição que tínhamos no modelo que usámos para calcular os tempos de cada atividade e incluir também as possíveis reduções.

$$t_j >= t_i + c_i - r_i - R_i, \forall (i,j) \in A, i \notin \{s, 7, 9, t\}$$
$$t_j >= t_i + c_i - r_i - 2 \times R_i, \forall (i,j) \in A, i \in \{7, 9\}$$
$$t_j >= t_s, (s,j) \in A$$

Temos assim o nosso modelo de programação linear inteira mista, que podemos executar num *solver* como o *LPSolve*, por exemplo.

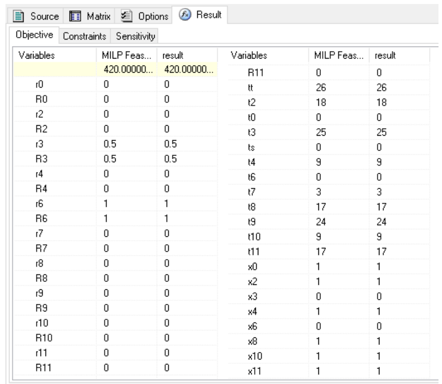
```
Source | Matrix | Options | Result
// custo associado à redução das durações das actividades
min: 200 r0 + 100 R0 + 1000 r2 + 500 R2 + 200 r3 + 100 R3 + 800 r4 + 400 R4 + 180 r6 + 90 R6 + 300 r7 + 1100 R7 + 200 r8 +
100 R8 + 200 r9 + 400 R9 + 1000 r10 + 500 R10 + 600 r11 + 300 R11;

// tempo máximo para concluir o projecto
tt <= 26;

// relações de precedência
arco_02: t2 >= t0 - r0 - R0 + 4 ; arco_23: t3 >= t2 - r2 - R2 + 7 ;
arco_i0: t0 >= ts + 0 ;           arco_04: t4 >= t0 - r0 - R0 + 4 ;
arco_42: t2 >= t4 - r4 - R4 + 9 ; arco_3f: tt >= t3 - r3 - R3 + 2 ;
arco_43: t3 >= t4 - r4 - R4 + 9 ; arco_i6: t6 >= ts + 0 ;
arco_74: t4 >= t7 - r7 - 2 R7 + 6 ; arco_83: t3 >= t8 - r8 - R8 + 4 ;
arco_9f: tt >= t9 - r9 - 2 R9 + 2 ; arco_67: t7 >= t6 - r6 - R6 + 5 ;
arco_78: t8 >= t7 - r7 - 2 R7 + 6 ; arco_89: t9 >= t8 - r8 - R8 + 4 ;
arco_610: t10 >= t6 - r6 - R6 + 5 ; arco_108: t8 >= t10 - r10 - R10 + 8 ;
arco_119: t9 >= t11 - r11 - R11 + 7 ; arco_1011: t11 >= t10 - r10 - R10 + 8;

// reduções máximas permitidas
r0 <= 0.5; R0 <= 0.5; r2 <= 3; R2 <= 1; r3 <= 0.5; R3 <= 0.5; r4 <= 2; R4 <= 1;
r6 <= 1; R6 <= 1; r8 <= 0.5; R8 <= 0.5; r10 <= 0.5; R10 <= 0.5; r11 <= 1; R11 <= 1;

r0 >= 0.5 - 1000 x0;
R0 <= 1000 - 1000 * x0;

r2 >= 3 - 1000 x2;
R2 <= 1000 - 1000 * x2;
```

Figura 10: Ficheiro de input.

```
Source | Matrix | Options | Result

r3 >= 0.5 - 1000 x3;
R3 <= 1000 - 1000 * x3;

r4 >= 2 - 1000 x4;
R4 <= 1000 - 1000 * x4;

r6 >= 1 - 1000 x6;
R6 <= 1000 - 1000 * x6;

r8 >= 0.5 - 1000 x8;
R8 <= 1000 - 1000 * x8;

r10 >= 0.5 - 1000 x10;
R10 <= 1000 - 1000 * x10;

r11 >= 1 - 1000 x11;
R11 <= 1000 - 1000 * x11;

r7 + R7 <= 1;
r9 + R9 <= 1;

bin x0,x2,x3,x4,x6,x8,x10,x11,r7,R7,r9,R9;
```

Figura 11: Ficheiro de input. (cont.)

Executando o programa com este ficheiro de input, obtemos o seguinte output:



Variables	MILP Feas...	result	Variables	MILP Feas...	result
	420.00000...	420.00000...	R11	0	0
r0	0	0	tt	26	26
R0	0	0	t2	18	18
r2	0	0	t0	0	0
R2	0	0	t3	25	25
r3	0.5	0.5	ts	0	0
R3	0.5	0.5	t4	9	9
r4	0	0	t6	0	0
R4	0	0	t7	3	3
r6	1	1	t8	17	17
R6	1	1	t9	24	24
r7	0	0	t10	9	9
R7	0	0	t11	17	17
r8	0	0	x0	1	1
R8	0	0	x2	1	1
r9	0	0	x3	0	0
R9	0	0	x4	1	1
r10	0	0	x6	0	0
R10	0	0	x8	1	1
r11	0	0	x10	1	1
R11	0	0	x11	1	1

Figura 12: Ficheiro de output.

Ficamos assim a saber quais as reduções de custo mínimo que devemos efetuar para reduzir a duração total do projeto em 3 U.T.. Como seria de esperar, as atividades cuja duração reduzimos pertencem ao caminho crítico, nomeadamente as atividades 3 e 6, visto que reduzir a duração de uma atividade não pertencente ao caminho crítico não iria alterar a duração total do projeto. A duração da atividade 3 é reduzida em 1 U.T. e a duração da atividade 6 em 2 U.T., com um custo total de 420 U.M..

Podemos agora colocar este novo plano de execução num diagrama de Gantt.

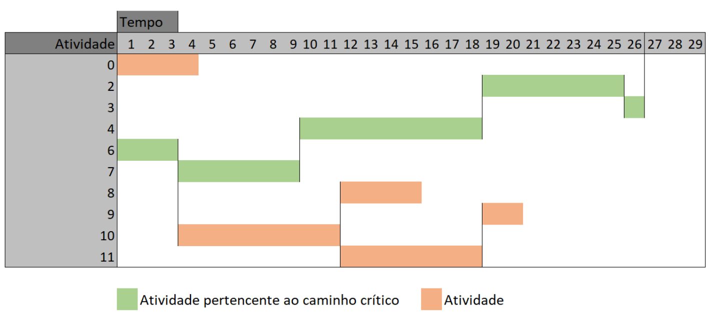


Figura 13: Diagrama de Gantt relativo ao nosso projeto, depois de aplicar as reduções.

Como podemos confirmar, o tempo total de execução do projeto desceu de 29 para 26 U.T., exatamente como pretendido.

Esta redução foi possível graças à redução do tempo das atividades 3 e 6. A atividade 3 foi reduzida em 1 U.T., da qual 0.5 U.T. foram a um custo de 200 U.M./U.T. e as outras 0.5 U.T. a um custo de 100 U.M./U.T.. Isto equivale a um custo total de redução de 150 U.M.. Quanto à atividade 6, a sua redução em 2 U.T. pode ser dividida numa redução em 1 U.T. a um custo de 180 U.M./U.T., seguida de uma redução de 1 U.T. a um custo de 90 U.M./U.T., com um custo total de 270 U.M.. Somando estes dois custos, obtemos o custo total da redução da duração do projeto, ou seja, $270 + 150 = 420$ U.M., o mesmo valor que obtivemos no *LPSolve*.

Através da análise dos custos das reduções, podemos verificar que as atividades cuja duração foi reduzida foram as atividades do caminho crítico com menor custo de redução, algo coerente com aquilo que seria expectável.