Programação linear: aplicações

J.M. Valério de Carvalho

19 de Novembro de 2019

Conteúdo

1 Método do caminho crítico

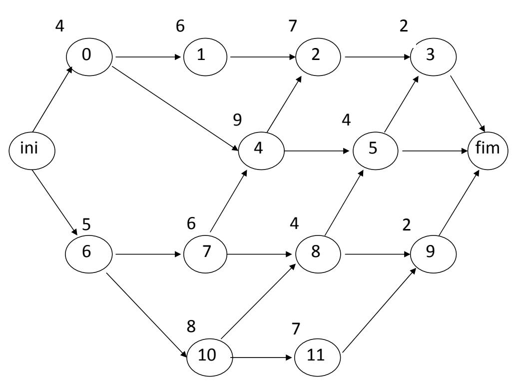
O método do caminho crítico, designado na literatura anglo–saxónica por *critical path method (CPM)*, constitui uma ferramenta muito importante em gestão de projectos. O método do caminho crítico é aplicado a projectos que podem ser decompostos num conjunto de actividades, entre as quais existem relações de precedência. Todas as actividades têm de ser realizadas, e considera-se que têm durações determinísticas. As restrições de precedência traduzem o facto de o instante em que se pode dar início a uma dada actividade ter de ser posterior aos instantes em que terminam as actividades que lhe são precedentes.

No método do caminho crítico, a rede que representa o projecto pode ser representada de duas formas alternativas: uma, em que as actividades do projecto são representadas por arcos do grafo, e a outra, em que são representadas por nós. Vamos considerar a segunda representação.

Considere um projecto com as actividades e as relações de precedência a seguir indicadas:

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

O grafo associado a este projecto, que inclui as actividades fictícias `ini` e `fim`, que podem ser pensadas como actividades, sem duração, de início e de conclusão do projecto, servindo para definir um vértice origem e um vértice destino do grafo, é o seguinte:



No problema em análise, o caminho crítico corresponde às actividades 6, 7, 4, 2 e 3, com uma duração de 29 unidades de tempo, que é também o menor tempo necessário para completar a execução de todo o projecto.

Caminho mais longo

O caminho crítico corresponde ao caminho mais longo entre o vértice que define o início do projecto e o vértice que define o fim do projecto. As actividades pertencentes ao caminho mais longo (que pode não ser único) são críticas, porque qualquer atraso verificado numa delas dá origem inevitavelmente a um atraso no tempo de execução do projecto global. Assim, estas actividades são as que devem ser seguidas e controladas mais de perto.

O problema de caminho mais longo entre um vértice s e um vértice t de um grafo acíclico (sem ciclos) pode ser formulado como um problema de programação linear utilizando variáveis de decisão x_{ij} associadas a cada arco (i, j) do grafo. Estas variáveis tomam o valor 1, se o arco fizer parte do caminho mais longo, e o valor 0, caso contrário.

O problema pode ser entendido como um modelo em que se injecta na rede, a partir do vértice s, uma unidade de fluxo que segue pelo caminho mais longo até atingir o vértice t. As restrições do problema, relativas a cada um dos vértices do grafo, traduzem a conservação de fluxo em cada vértice, ou seja, o número de unidades de fluxo que entram no vértice j através dos arcos (i, j) deve ser igual ao número de unidades que dele saem através dos arcos (j, k). Isto é equivalente a afirmar que, se há um arco do caminho a entrar num vértice, deverá haver um arco a sair; também que, se não houver arco a entrar no vértice, o caminho não passará pelo vértice. Esta relação é válida para todos os vértices, excepção feita ao vértice s, onde é introduzida uma unidade de fluxo na rede, e ao vértice t, onde a unidade é removida. De facto, a restrição relativa ao vértice t pode ser retirada do modelo, por ser linearmente dependente das outras restrições.

```
//   caminho mais longo num grafo acíclico
max: 4 x01 + 4 x04 + 6 x12 + 7 x23 + 2 x3f + 9 x42 + 9 x45 + 4 x53 +
     4 x5f + 5 x67 + 5 x610 + 6 x74 + 6 x78 + 4 x85 + 4 x89 + 2 x9f +
     8 x108 + 8 x1011 + 7 x119;

// fluxo que entra no vértice = fluxo que sai
vertice_i: xi0 + xi6 = 1;
vertice_0: xi0 = x01 + x04;
vertice_1: x01 = x12;
vertice_2: x12 + x42 = x23;
vertice_3: x23 + x53 = x3f;
vertice_4: x04 + x74 = x42 + x45;
vertice_5: x45 + x85 = x53 + x5f;
vertice_6: xi6 = x67 + x610;
vertice_7: x67 = x74 + x78;
vertice_8: x78 + x108 = x85 + x89;
vertice_9: x89 + x119 = x9f;
vertice_10: x610 = x108 + x1011;
vertice_11: x1011 = x119;
```

As variáveis de decisão que tomam o valor 1 na solução óptima definem o caminho mais longo entre os vértices i e f.

Minimizaçao do tempo de conclusão

Existe outro modelo de programação matemática para este problema em que cada variável de decisão $t_i, \forall i$, representa o tempo de início da actividade i, e em que o objectivo é minimizar o tempo de execução total do projecto obedecendo a todas as precedências.

As restrições do problema, relativas a cada um dos arcos do grafo, traduzem as relações de precedência entre as actividades. Para uma dada actividade j, o tempo de início da actividade j deve ser posterior ao tempo de conclusão de cada uma das actividades i que precedem j. Dado que t_i designa o tempo de início da actividade i, a função $t_i + d_i$ designa o tempo de conclusão da actividade i. O projecto termina no instante de tempo t_f, quando todas as actividades predecessoras imediatas da actividade fictícia `fim` estiverem concluídas.

```
/* função objectivo */
min: tf ;

/* restrições */
arco_01: t1 >= t0 + 4 ;
arco_12: t2 >= t1 + 6 ;
arco_23: t3 >= t2 + 7 ;
arco_i0: t0 >= ti + 0 ;
arco_04: t4 >= t0 + 4 ;
arco_42: t2 >= t4 + 9 ;
arco_53: t3 >= t5 + 4 ;
arco_3f: tf >= t3 + 2 ;
arco_45: t5 >= t4 + 9 ;
arco_5f: tf >= t5 + 4 ;
arco_i6: t6 >= ti + 0 ;
arco_74: t4 >= t7 + 6 ;
arco_85: t5 >= t8 + 4 ;
arco_9f: tf >= 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;
```

O valor da variável de decisão t_i na solução óptima define o tempo de início de execução da actividade i, permitindo construir um plano de execução do projecto, designado por diagrama de Gantt. O projecto é executado num tempo com a duração do valor da solução óptima. O Diagrama de Gantt do projecto em análise é apresentado na Figura 1.

Crashing times

Embora haja uma duração definida para cada actividade, é muitas vezes possível, aumentando os recursos nela aplicados, reduzir a sua duração. Isto é feito com custos suplementares, mas pode trazer o benefício de reduzir a duração de execução do projecto global.

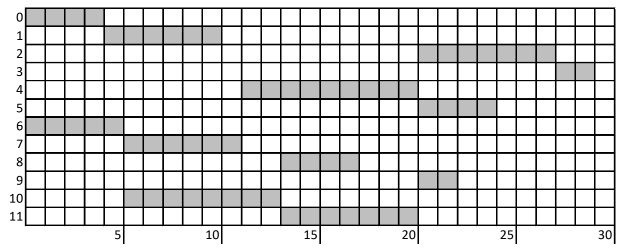


Figura 1: Diagrama de Gantt do projecto

A forma de modelar o aumento de custo em função da redução da duração da actividade pode ser feita de várias formas diferentes. Vamos assumir que as reduções de custo são lineares com o tempo, e associar a cada actividade três parâmetros adicionais: o primeiro é o valor do custo normal, expresso em unidades monetárias [U.M.], o segundo é o valor do custo suplementar de reduzir a duração da actividade de uma unidade de tempo [U.T.], expresso em [U.M./U.T.], e o terceiro é o valor da máxima redução de tempo que é permitida para a actividade, expresso em [U.T.]. Esses valores estão apresentados na seguinte Tabela:

Actividade	Custo Normal	Custo suplementar de redução	Máxima redução
0	400	100	1
1	1000	300	2
2	1400	500	4
3	300	100	1
4	2000	400	3
5	1000	800	1
6	800	90	2
7	900	–	0
8	600	100	1
9	300	—	0
10	1600	500	1
11	1400	300	2

A título ilustrativo, para a Actividade 1, cuja duração normal é de 6 U.T. e custo normal é 1000 U.M., reduzir a duração da actividade de 1 U.T., passando a 5 U.T., tem um custo suplementar de 300 U.M., ou seja, a Actividade 1 passa a ter um custo total de 1300 U.M.. A redução máxima que se pode obter é de 2 U.T., ou seja, é possível, com um custo suplementar de 600 U.M., realizar esta actividade no tempo mínimo de 4 U.T., com um custo total de 1600 U.M..

Suponha que se pretende que o tempo de execução do projecto seja reduzido em 3 U.T.. O objectivo do problema é decidir como devem ser reduzidas as durações das actividades, de modo a realizar o projecto na nova duração desejada, com um custo suplementar mínimo.

```
// custo associado à redução das durações das actividades
min: 100 r0 + 300 r1 + 500 r2 + 100 r3 + 400 r4 + 800 r5 +  90 r6 +   0 r7 +
     100 r8 +   0 r9 + 500 r10+ 300 r11;

// tempo máximo para concluir o projecto
```

```
tf <= 26;

// relações de precedência
// na restrição tj >= ti - ri + di, a função ti - ri + di designa
// o tempo de conclusão da actividade i após a redução da duração,
// de di para -ri + di

arco_01: t1 >= t0 - r0 + 4 ;
arco_12: t2 >= t1 - r1 + 6 ;
arco_23: t3 >= t2 - r2 + 7 ;
arco_i0: t0 >= ti + 0 ;
arco_04: t4 >= t0 - r0 + 4 ;
arco_42: t2 >= t4 - r4 + 9 ;
arco_53: t3 >= t5 - r5 + 4 ;
arco_3f: tf >= t3 - r3 + 2 ;
arco_45: t5 >= t4 - r4 + 9 ;
arco_5f: tf >= t5 - r5 + 4 ;
arco_i6: t6 >= ti + 0 ;
arco_74: t4 >= t7 - r7 + 6 ;
arco_85: t5 >= t8 - r8 + 4 ;
arco_9f: tf >= t9 - r9 + 2 ;
arco_67: t7 >= t6 - r6 + 5 ;
arco_78: t8 >= t7 - r7 + 6 ;
arco_89: t9 >= t8 - r8 + 4 ;
arco_610: t10 >= t6 - r6 + 5 ;
arco_108: t8 >= t10 - r10 + 8 ;
arco_119: t9 >= t11 - r11 + 7 ;
arco_1011: t11 >= t10 - r10 + 8;

// reduções máximas permitidas
r0 <= 1 ;
r1 <= 2 ;
r2 <= 4 ;
r3 <= 1 ;
r4 <= 3 ;
r5 <= 1 ;
r6 <= 2 ;
r7 <= 0 ;
r8 <= 1 ;
r9 <= 0 ;
r10 <= 1 ;
r11 <= 2 ;
```

Balanceamento entre a duração do projecto e seu custo

A duração global do projecto e o seu custo são, por vezes, balanceados para se atingir um compromisso. Tomando a duração global do projecto t_f como parâmetro, e resolvendo o modelo para cada valor do parâmetro, podemos obter o respectivo valor de custo suplementar, como se mostra na seguinte tabela:

tempo de execução	custo suplementar
29	0
28	90
27	180
26	280
25	680
24	1080
23	1480
22	1980
21	2480
20	2980
19	4580

É de salientar que não é possível executar o projecto numa duração de 18 U.T., por as reduções de duração máximas para as actividades não o permitirem.

Os resultados desta análise são muitas vezes apresentados num gráfico em que, no eixo das abscissas, se representam as diferentes durações possíveis para o projecto e, no eixo das ordenadas, se representam os custos totais (ou suplementares) de execução do projecto.