Modelos Determinísticos de Investigação Operacional

Engenharia Informática

Universidade do Minho

Janeiro 2021

Método do caminho crítico

Alexandre Costa - A78890
Luís Pereira – A77667
Ricardo Gomes - A93785
Rúben Rodrigues – A80960
Sara Marques - A89477

Conteúdo

Introdução

Este projeto surge no domínio da unidade curricular Modelos Determinísticos de Investigação Operacional com o objetivo de entender a capacidade de analisar problemas complexos, desenvolver modelos e interpretar as respectivas soluções. Neste trabalho prático será abordado o método do caminho crítico utilizado em projetos que podem ser decompostos em conjuntos de atividades com duração determinística nas quais é possível estabelecer relações de precedência. Neste método, a rede que representa as atividades do projeto foi estabelecida sobre nós.

Desenvolvimento do modelo

Muitos problemas práticos de Investigação Operacional podem ser expressos recorrendo a programação linear. O método do caminho crítico será utilizado de modo a decidir a redução do tempo das atividades para obedecer às novas restrições de duração, tendo em atenção o menor custo suplementar.

Parte 0

Grafo de precedencia

O valor de ABCDE igual a 93785 obriga a retirar as atividades 5 e 8 e a refazer as regras de precedência. Assim sendo, obtemos a seguinte rede:

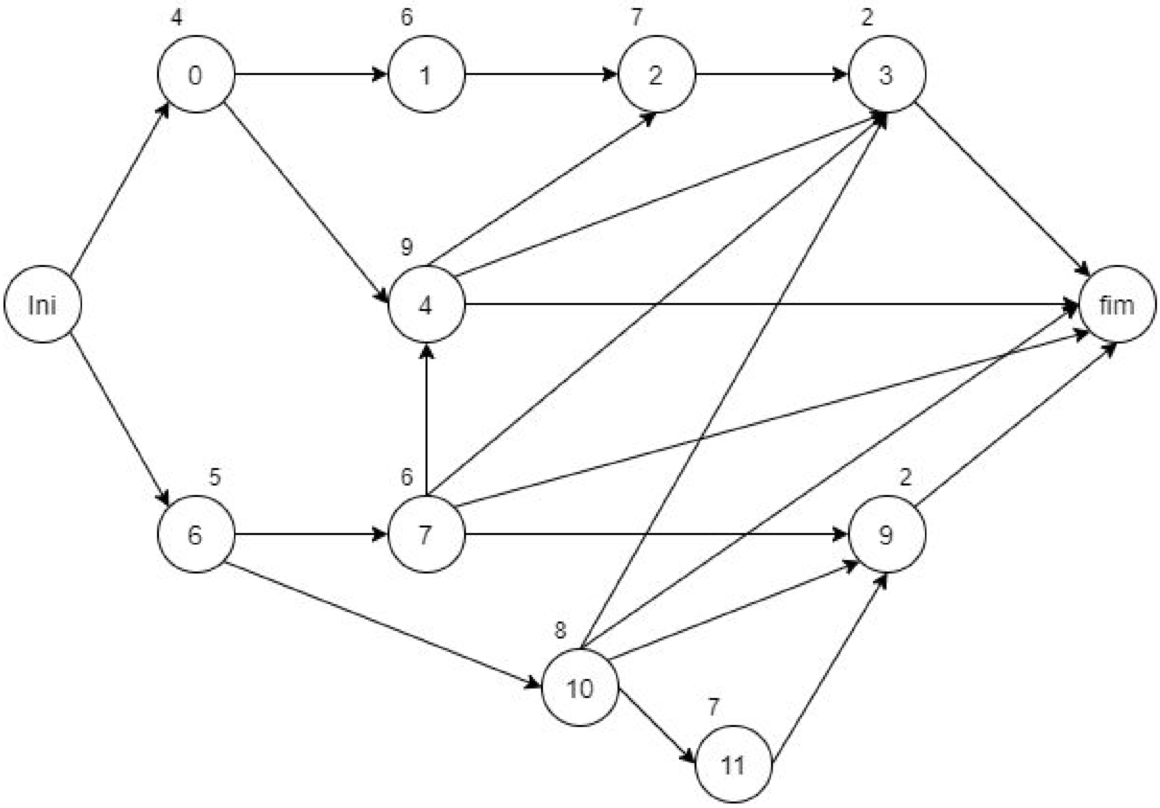


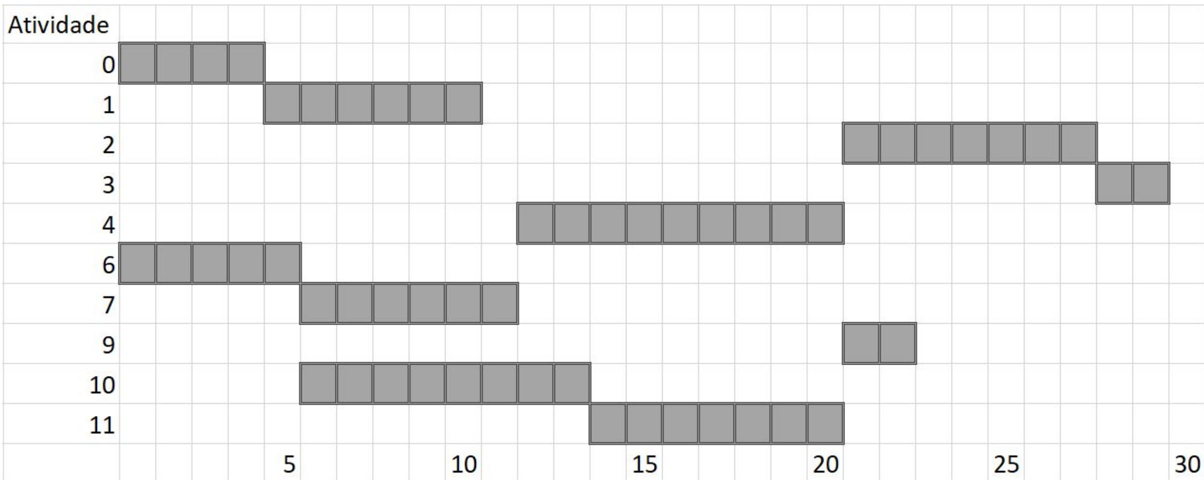
Diagrama de Gantt sem reduções

Código linear para encontrar o tempo mínimo para executar todas as atividades.

```
/* função objectivo */
min: tf ;
/* restrições */
arco_i0: t0 >= ti + 0 ;
arco_01: t1 >= t0 + 4 ;
arco_12: t2 >= t1 + 6 ;
arco_23: t3 >= t2 + 7 ;
arco_3f: tf >= t3 + 2 ;
arco_04: t4 >= t0 + 4 ;
arco_42: t2 >= t4 + 9 ;
arco_43: t3 >= t4 + 9 ;
arco_4f: tf >= t4 + 9 ;
arco_i6: t6 >= ti + 0 ;
arco_67: t7 >= t6 + 5 ;
arco_610: t10 >= t6 + 5 ;
arco_74: t4 >= t7 + 6 ;
arco_73: t3 >= t7 + 6 ;
arco_79: t9 >= t7 + 6 ;
arco_7f: tf >= t7 + 6 ;
arco_9f: tf >= t9 + 2 ;
arco_103: t3 >= t10 + 8;
arco_109: t9 >= t10 + 8;
arco_1011: t11 >= t10 + 8;
arco_10f: tf >= t10 + 8;
arco_119: t9 >= t11 + 7 ;
```

Variables	result
	29
tf	29
t0	0
ti	0
t1	4
t2	20
t3	27
t4	11
t6	0
t7	5
t10	5
t9	20
t11	13

A solução ótima, com uma duração de 29 unidades de tempo, encontra-se no seguinte diagrama de Gantt:



Parte 1

Formulação

O objetivo passa por diminuir a duração total obtida anteriormente, mais concretamente, em 3 U.T. Para além da remoção das atividades 5 e 8, é necessário definir algumas restrições às atividades 7 e 9. A atividade 7 pode ser reduzida em 1 U.T. com um custo adicional de 300 U.M. ou com uma duração de 4 U.T. com um custo adicional de 1100 U.M. A atividade 9 poderá ser realizada com uma redução de 1 U.T. a um custo adicional de 200 U.M. ou com uma duração de 0 U.T. custando 400 U.M. adicionais. Traduzindo o problema para uma tabela em que C1 e C2 é o custo de reduzir 1 U.T e Max.red é a redução máxima de tempo que pode ser efetuada para cada atividade.
Tendo em conta que a redução C2 só pode ser efetuada depois de atingir a redução máxima de C1, concluímos que para a atividade 7 o custo C1 será 300 U.M. com uma redução máxima de 1 U.T. e C2 teria um custo de 1100 - 300 (custo da máxima redução de C1 para atividade 7) = 800 U.M com uma redução máxima também de 1 U.T. Seguindo o mesmo raciocínio para a atividade 9, obtivemos que C1 = 200 U.M. para redução máxima 1 U.T. e C2= 400 - 200 = 200 U.M. para uma redução máxima de 1.

A tabela seguinte traduz estes resultados.

Atividade	Duração	Precedências	Custo normal	C1	Max.red. Custo C1	C2	Max.red. Custo C2
0	4	--	400	200	0.5	100	0.5
1	6	0	1000	600	1	300	1
2	7	1, 4	1400	1000	3	500	1
3	2	2, 4, 7, 10	300	200	0.5	100	0.5
4	9	0, 7	2000	800	2	400	1
6	5	--	800	180	1	90	1
7	6	6	900	300	1	800	1
9	2	7, 10, 11	300	200	1	200	1
10	8	6	1600	1000	0.5	500	0.5
11	7	10	1400	600	1	300	1

Objetivo e coerência do modelo

Em termos reais, o objetivo do nosso modelo passa por diminuir o tempo que uma atividade demora, utilizando mais recursos, sendo estes recursos expressos no nosso problema como unidades monetárias. Sabendo a redução máxima que conseguimos efetuar em cada atividade e o seu custo, pretendemos encontrar o custo mínimo para atingir um certo objetivo, no caso deste projeto, reduzir o tempo de efetuar todas as atividades para 26 U.T. Em termos de programação linear isto será traduzido em implementar uma função objetivo que minimize o custo associado às reduções das atividades e um arco final que seja menor ou igual a 26 U.T. e entre cada arco interior teremos que encontrar o seu tempo de execução tendo como variáveis as reduções de tempo que podem ser efetuadas por C1 e C2. Por fim, implementar as reduções máximas que podem ser efetuadas por C1 e C2 e a restrição em que C2 só pode ser efetuado depois da redução máxima de C1 se esgotar.

Modelo

Variáveis de decisão

- ri e ci - Decisão de quanto reduzir, ri para reduções C1 e ci para C2.
- si - decisão binária de puder ou não fazer reduções C2

Parâmetros

- tf <= 26 - O tempo final de execução tem de ser menor ou igual a 26.
- Tempo normal para efetuar uma atividade.
- Custos associados com cada redução.

Função objetivo

- Problema de minimização do custo de redução.

Restrições

- tj >= ti - ri - ci + di - tempo de conclusão da actividade i após a redução da duração
- ri <= x - Atividade i com redução máxima C1 menor ou igual a x
- x si <= ri - Sendo x o custo máximo de redução da atividade i, si será 1 somente quando ri é máximo, restringindo a opção de redução C2
- ci <= x si - Sendo si binário e x o valor da redução máxima C2, ci terá valor 0 se não puder ser efetuada redução C2 e terá um valor <= à redução máxima de C2 se puder.

Ficheiro de input

De seguida, é apresentado o ficheiro de input definido de acordo com o modelo implementado.

```
// custo associado à redução das durações das actividades
min: 200 r0 + 600 r1 + 1000 r2 + 200 r3 + 800 r4 + 180 r6 + 300 r7 + 200 r9 + 1000 r10+ 600 r11
+ 100 c0 + 300 c1 + 500 c2 + 100 c3 + 400 c4 + 90 c6 + 800 c7 + 200 c9 + 500 c10 + 300 c11;

// tempo máximo para concluir o projecto
tf <= 26;
// relações de precedência
// na restrição tj >= ti - ri - ci + di, a função ti - ri - ci + di designa
// o tempo de conclusão da actividade i após a redução da duração,
// de di para -ri - ci + di
arco_i0: t0 >= ti + 0 ;
arco_01: t1 >= t0 - r0 - c0 + 4 ;
arco_12: t2 >= t1 - r1 - c1 + 6 ;
arco_23: t3 >= t2 - r2 - c2 + 7 ;
arco_3f: tf >= t3 - r3 - c3 + 2 ;
arco_04: t4 >= t0 - r0 - c0 + 4 ;
arco_42: t2 >= t4 - r4 - c4 + 9 ;
arco_43: t3 >= t4 - r4 - c4 + 9 ;
arco_4f: tf >= t4 - r4 - c4 + 9 ;
arco_i6: t6 >= ti + 0 ;
arco_67: t7 >= t6 - r6 - c6 + 5 ;
arco_610: t10 >= t6 - r6 - c6 + 5 ;
arco_74: t4 >= t7 - r7 - c7 + 6 ;
arco_73: t3 >= t7 - r7 - c7 + 6 ;
arco_79: t9 >= t7 - r7 - c7 + 6 ;
arco_7f: tf >= t7 - r7 - c7 + 6 ;
arco_9f: tf >= t9 - r9 - c9 + 2 ;
arco_103: t3 >= t10 - r10 - c10 + 8;
arco_109: t9 >= t10 - r10 - c10 + 8;
arco_1011: t11 >= t10 - r10 - c10 + 8;
arco_10f: tf >= t10 - r10 - c10 + 8;
arco_119: t9 >= t11 - r11 - c11 + 7 ;

// reduções máximas permitidas C1
r0 <= 0.5 ;
r1 <= 1 ;
r2 <= 3 ;
r3 <= 0.5 ;
r4 <= 2 ;
r6 <= 1 ;
r7 <= 1 ;
r9 <= 1 ;
r10 <= 0.5 ;
r11 <= 1 ;

// restricao reducoes C2
0.5 s0 = r0 ;
1 s1 = r1 ;
3 s2 = r2 ;
0.5 s3 = r3 ;
2 s4 = r4 ;
1 s6 = r6 ;
1 s7 = r7 ;
1 s9 = r9 ;
0.5 s10 = r10 ;
1 s11 = r11 ;

// reduções máximas permitidas C2
c0 <= 0.5 s0;
c1 <= 1 s1 ;
c2 <= 1 s2 ;
c3 <= 0.5 s3 ;
c4 <= 1 s4 ;
c6 <= 1 s6 ;
c7 <= 1 s7 ;
c9 <= 1 s9 ;
c10 <= 0.5 s10 ;
c11 <= 1 s11 ;

bin s0,s1,s2,s3,s4,s6,s7,s9,s10,s11;
```

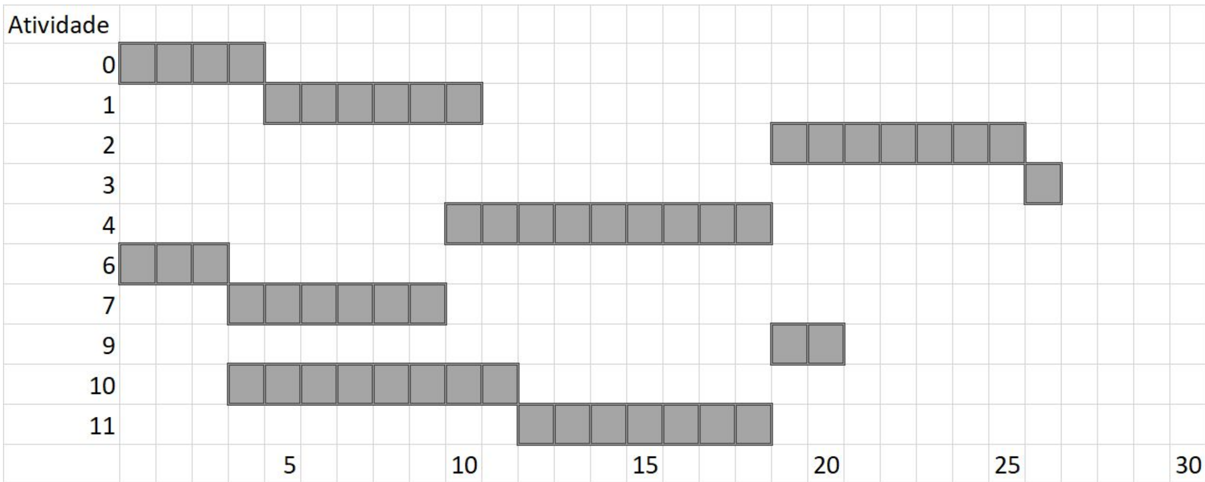
Ficheiro de output

De seguida, é apresentado o ficheiro de output definido de acordo com o modelo implementado.

Variables	MILP ...	res...
	420.0...	420.00...
tf	26	26
t3	25	25
t2	18	18
t9	18	18
t11	11	11
t4	9	9
t1	4	4
t7	3	3
t10	3	3
c6	1	1
r6	1	1
s3	1	1
s6	1	1
r3	0.500...	0.5000...
c3	0.5	0.5
r0	0	0
r1	0	0
r2	0	0
r4	0	0
r7	0	0
r9	0	0
r10	0	0
r11	0	0
c0	0	0
c1	0	0
c2	0	0
c4	0	0
c7	0	0
c9	0	0
c10	0	0
c11	0	0
t0	0	0
ti	0	0
t6	0	0
s0	0	0
s1	0	0
s2	0	0
s4	0	0
s7	0	0
s9	0	0
s10	0	0
s11	0	0

Diagrama de Gantt após redução

O seguinte diagrama de Gantt apresenta a planificação das atividades após redução:



Como se pode verificar, a solução obtida tem uma duração de 26 U.M., tal como foi idealizado.
A melhoria no tempo de duração deve-se à redução da atividade 3 (a um custo de 0.5*200+0.5*100) e da atividade 6 (a um custo 1*180 + 1*90), tendo o custo total acrescido de 420 U.M.

Validação da solução

Como a solução ótima obedeceu a todas as restrições, o seu valor de 420 U.M. para a redução da atividade 3 e 6 é válido, como foi explicado no Diagrama de Gantt após redução e o tempo total para a execução das atividades foi efetivamente reduzido em 3 U.T. de 29 U.T. para 26 U.T.
Podemos de facto considerar que esta é uma solução ótima válida para este modelo.

Conclusão

Num contexto de programação linear, procura-se primeiro encontrar o modelo que melhor descreve o problema, quer seja de maximização ou minimização. Problemas de investigação operacional podem ser expressos como modelos de programação linear, para determinar a solução de, por exemplo, o fluxo de atividades. Neste caso, foi utilizado o método do caminho crítico para resolver o projeto.É uma abordagem que divide o projeto em várias tarefas, exibindo-as num gráfico de fluxo e, de seguida, calcula a sua duração total com base na duração estimada para cada tarefa. O método identifica as tarefas que são essenciais, de acordo com tempo, para a conclusão do projeto.

O modelo apresentado modela qualquer problema do mundo real, desde que contemple as mesmas restrições de precedência.

Por todas estas razões, consideramos que o trabalho modelado é efetivamente correto e obtém o melhor planeamento a seguir.

Referências

Valério de Carvalho J.M., “Programação Linear - modelos - Investigação Operacional”, 2 de outubro de 2020