Universidade do Minho

Mestrado Integrado em Engenharia Informática

Modelos Determinísticos de Investigação Operacional

Critical Path Method

Etienne Costa (a76089)
João Coutinho (a86272)
Margarida Faria (a71924)
Rui Azevedo (a80789)

8 de Janeiro de 2020

Conteúdo

Lista de Figuras

Lista de Tabelas

1 Introdução

A realização deste trabalho prático ocorre no âmbito da unidade curricular Modelos Determinísticos de Investigação Operacional e visa desenvolver a capacidade de analisar sistemas complexos, de criar modelos para os descrever, de obter soluções para esses modelos utilizando programas computacionais adequados validando e interpretando as soluções obtidas de modo a elaborar recomendações para o sistema em análise. Neste caso em concreto, pretende-se aplicar o método do caminho crítico a um projecto que será decomposto num conjunto de actividades, que se considera terem durações determinísticas, entre as quais existem relações de precedência.

2 Apresentação do projecto

O método do caminho críticio, designado na literatura anglo-saxónica por **critical path method**, 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 actvidades que lhe são precedentes. Num contexto inicial é fornecido um projecto com as seguintes actividades e relações de precedência :

Actividade	Duração	Precedência
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 é o seguinte:

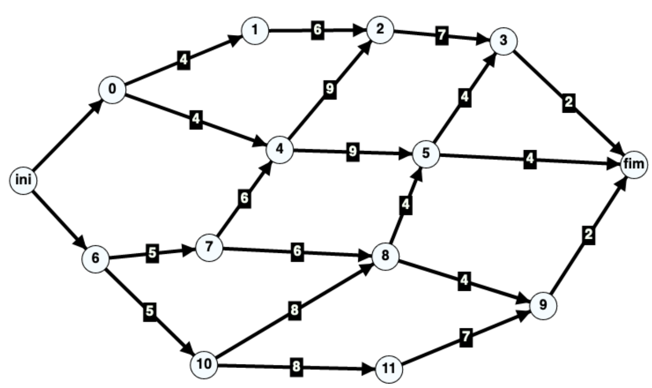

Figura 1: Grafo inicial associado ao projecto

Neste caso , 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.

Seja ABC**DE** o número de inscrição do aluno do grupo com maior número de inscrição, foram removidas da lista inicial as actividades **D e E** e aplicadas as seguintes regras de modo a determinar as precedências :

1. **os sucessores da actividade D passam a ter como novas precedências os antecessores da actividade D.**

2. **os sucessores da actividade E passam a ter como novas precedências os antecessores da actividade E.**

O resultado da aplicação destas regras conduziu-nos a um novo projecto com as seguintes actividades e precedências:

Actividade	Duração	Precedência
ini	0	-
0	4	ini
1	6	0
3	2	1,4,5
4	9	0,6
5	4	4,8
6	5	ini
8	4	6,10
9	2	8,11
10	8	6
11	7	10
fim	-	3,5,9

O grafo associado a este projecto é o seguinte :

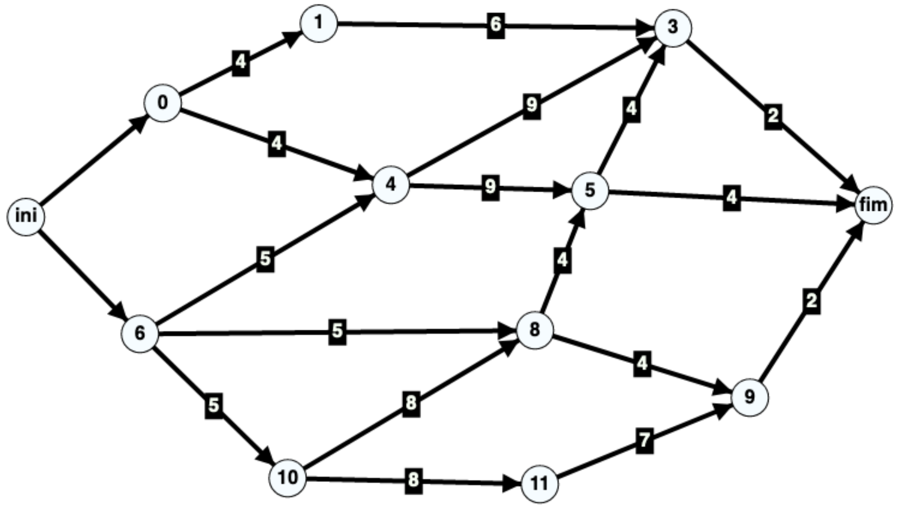

Figura 2: Grafo Final associado ao novo projecto.

2.1 Solução do projecto

Havendo várias formas de determinar o caminho crítico, foi decidida a implementação de dois métodos que convergem para o mesmo valor e simultaneamente para a mesma interpretação do resultado obtido, sendo eles os seguintes :

1. **Caminho mais longo.**
2. **Minimização do tempo de conclusão.**

2.2 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 são críticas, porque qualquer atraso verificado numa delas dá origem inevitavelmente a um atraso no tempo de execução do projecto global. Sendo assim modulou-se o problema de caminho mais longo entre um vértice **ini** e um vértice **fim** de um grafo acíclico como um problema de programação linear utilizando variáveis de decisão $x_{i,j}$ associadas a cada arco (i,j) do grafo, sendo que elas tomam o valor 1, se o arco fizer parte do caminho mais longo, e o valor 0, caso contrário. Relativamente as restrições do problema, aplicadas a cada um dos vértices, traduzem-se na conservação do fluxo, i.e, a quantidade que entra para cada um dos vértices deve ser igual a que sai.

2.2.1 Ficheiro de Input

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

```
/* Funcao Objetivo */

max: 4x0_1 + 4x0_4 + 6x1_3 + 2x3_f + 9x4_3 + 9x4_5 + 4x5_3+ 4x5_f + 5x6_4 +
    5x6_8 + 5x6_10 + 4x8_5 + 4x8_9 + 2x9_f + 8x10_8 + 8x10_11 + 7x11_9 ;

 /* Restri  es  de conserva  o de fluxo */

xi_0 + xi_6 = 1;              xi_0 = x0_1 + x0_4;
x1_3 + x4_3 + x5_3 = x3_f ;   x0_4 + x6_4 = x4_3 + x4_5 ;
x4_5 + x8_5 = x5_3 + x5_f;    xi_6 = x6_4 + x6_8 + x6_10;
x6_8 + x10_8 = x8_5 + x8_9;   x8_9 + x11_9 = x9_f;
x6_10 = x10_8 +x10_11;        x10_11 = x11_9;
x3_f + x5_f + x9_f = 1;       x0_1 = x1_3;

/* Restricoes de nao negatividade */

x0_1 >=0;   x0_4 >=0;    x1_3 >=0;
x3_f >=0;   x4_3 >=0;    x4_5 >=0;
x5_3 >=0;   x5_f >=0;    x6_4 >=0;
x6_8 >=0;   x6_10 >=0;   x8_5 >=0;
x8_9 >=0;   x9_f >=0;    x10_8 >=0;
x10_11 >=0; x11_9 >=0;


/* Restricoes de integralidade */

int x0_1 , x0_4 , x1_3 , x3_f, x4_3, x4_5, x5_3, x5_f, x6_4, x6_8, x6_10,
    x8_5, x8_9, x9_f , x10_8 , x10_11 , x11_9;
```

2.2.2 Ficheiro de Output

De seguida, apresenta-se o conjunto de soluções válidas.

```
Value of objective function : 23.00

Actual values of the variables:
x0_1 = 0  x0_4 = 0  x1_3 = 0  x3_f = 1
x4_3 = 0  x4_5 = 0  x5_3 = 1  x5_f = 0
x6_4 = 0  x6_8 = 0  x6_10 = 1 x8_5 = 1
x8_9 = 0  x9_f = 0  x10_8 = 1 x10_11 = 0
x11_9 = 0 xi_0 = 0 xi_6 = 1
```

Neste caso , o caminho crítico corresponde às actividades 6,10,8,5 e 3, com uma duração de 23 unidades de tempo, que é também o menor tempo necessário para completar a execução de todo o projecto.

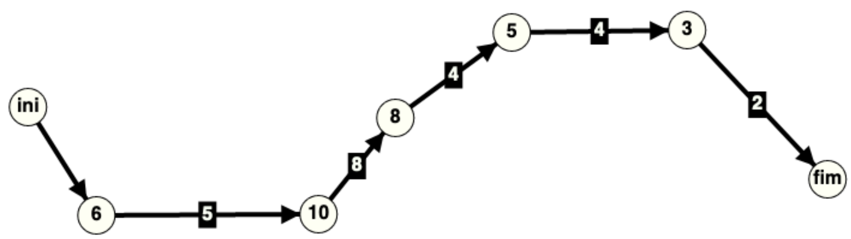

Figura 3: Caminho mais longo.

2.3 Minimização do tempo de conclusão

Existe outro modelo de programação matemática para este problema em que cada variável ti , $\forall(i)$, representa o tempo de início da actividade i, 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, tempo de início da actividade j deve ser posterior ao tempo de conclusão de cada uma das actividades i que precedem j. Sendo ti o tempo de início da actividade i, a função $ti{+}di$ designa o tempo de conclusão da actividade i. O projecto termina no instante de tempo tf, quando todas as actividades predecessoras imediatas da actividade fictícia **fim** estiverem concluídas.

2.3.1 Ficheiro de Input

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

```

/* Funcao Objectivo */

   min: tfinal -tinicial ;

/* Vari veis de decis o  */

   tinicial : Tempo inicial do projecto.

   ti : representa o tempo de in cio da actividade i,com i
   {0,1,3,4,5,6,8,9,10,11}

   tfinal : Tempo final do projecto.


/* Restricoes de precedencia */

 t(i,j) : t(j) >= t(i) + d(i),  sendo d(i) a dura  o de uma determinada
   actividade.

arco(inicial_0)          arco(inicial_6)
t0>= tinicial + 0;       t6>= tinicial + 0;

arco(0,1)                arco(0,4)
t1>= t0 + 4;             t4>= t0 + 4;

```

```
arco(1,3)                 arco(3,final)
t3>= t1 + 6;              tfinal>= t3 + 2;

arco(4,3)                 arco(4,5)
t3>= t4 + 9;              t5>= t4 + 9;

arco(5,3)                 arco(5,final)
t3>= t5 + 4;              tfinal>= t5 + 4;

arco(6,4)                 arco(6,8)
t4>= t6 + 5;              t8>= t6 + 5;

arco(6,10)                arco(8,5)
t10>= t6 + 5;             t5>= t8 + 4;

arco(8,9)                 arco(9,final)
t9>= t8 + 4;              tfinal>= t9 + 2;

arco(10,8)                arco(10,11)
t8>= t10 + 8;             t11>= t10 + 8;

arco(11,9)
t9>= t11 + 7;

/* Restricoes de n o negatividade */
 tinicial>=0;  t0>=0;   t1>=0;
 t3>=0;        t4>=0;   t5>=0;
 t6>=0;        t8>=0;   t9>=0;
 t10>=0;       t11>=0;   tfinal>=0;

/* Restricoes de integralidade */

int tinicial,t0,t1,t3,t4,t5,t6,t8,t9,t10,t11,tfinal;
```

2.3.2 Ficheiro de Output

De seguida, apresenta-se o conjunto de soluções válidas.

```
Value of objective function : 23.00

Actual values of the variables:

tfinal = 23  tinicial = 0  t0 = 0  t6 = 0

t1 = 4  t4 = 5  t3 = 21  t5 = 17

t8 = 13  t10 = 5  t9 = 20 t11 = 13
```

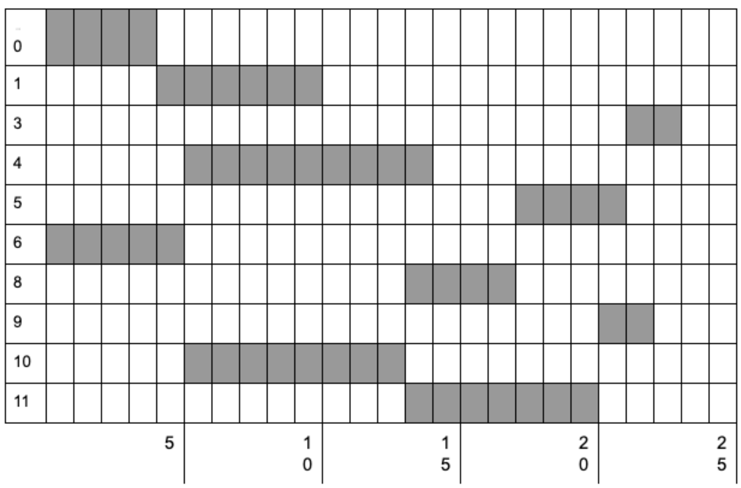

Figura 4: Diagrama de Gantt.

Como era de se esperar os dois valores são exatamente o mesmo, indicando que 23 é o tempo máximo para a execuçao do projecto.

3 Parte I

3.1 Actividades Paralelas

O critical path method assume que qualquer actividade pode ser realizada em paralelo com qualquer outra.
Há no entanto, situações em que os recursos são limitados e a realização das actividades depende dos recursos disponíveis.
Havendo essa limitação de recursos disponíveis , foram escolhidas 3 actividades que decorrem em paralelo e que necessitam do mesmo recurso para a realização da respectiva actividade, sendo assim existe a possibilidade da duração global do projecto ser superior em consequência disso.

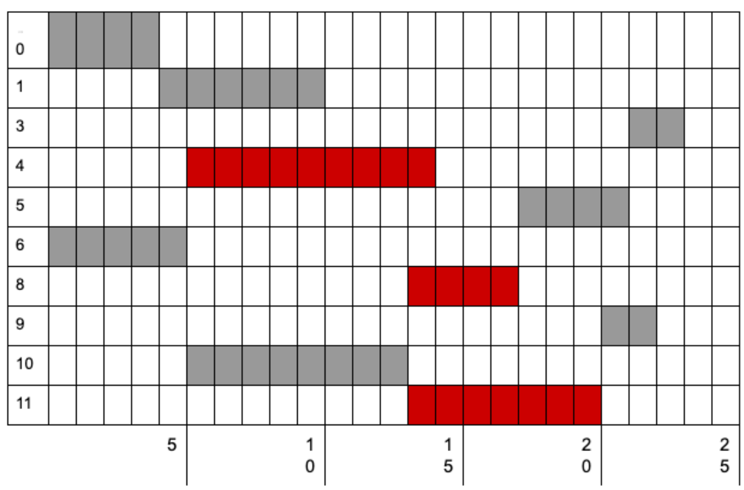

Figura 5: Actividades Paralelas

3.2 Análise do problema

O objectivo do problema é realizar o projecto no menor tempo possível tendo em conta as limitações introduzidas.
Visto que, as actividades não podem ser realizadas ao mesmo tempo devido ao uso do mesmo recurso foi necessário criar novas restrições que contemplam essa situação, que terá como consequência o aumento da duração global do projecto. Como foram escolhidas 3 actividades, existem 6 situações diferentes que podem ocorrer.

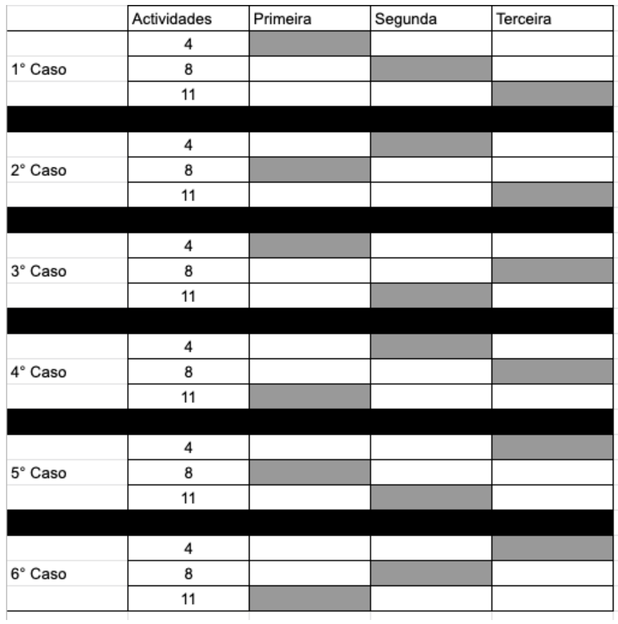

Figura 6: Casos possíveis de precedência

3.3 Solução do problema

A solução do modelo consiste na aplicação de algoritmos existentes ou no desenvolvimento de novos para obtenção da solução óptima do modelo. Visto que este problema consiste em realizar o projecto no menor tempo possível tendo em conta as limitações introduzidas, optou-se por utilizar as restrições de precedência,integralidade , não negatividade e não simultaniedade de modo a obter o conjunto de soluções válidas, tendo sido recorrido a Programação Linear Inteira Mista.

3.3.1 Variáveis de decisão

Para a resolução deste problema foram utilizadas as seguintes variáveis de decisão:

- $tfinal$: Tempo final do projecto.
- $tinicial$: Tempo inicial do projecto.
- $\begin{cases} \text{yi,j= 1} & \text{, se a actividade i precede actividade j} \\ \text{yi,j= 0} & \text{, se a actividade j precede actividade i} \end{cases}$
- di : Duração da actividade i, com i $\in\{0,1,3,4,5,6,8,9,10,11\}$.

- ti : Representa o tempo de início da actividade i, com i $\in\{0,1,3,4,5,6,8,9,10,11\}$.

3.3.2 Restrições

As restrições implementadas ditam as regras para o correto funcionamento do sistema. Com base nisso, dividiu-se as restrições em quatro grupos:

- **Restrições de precedência**: são as restrições que garantem que para toda actividade i que sucede a actividade j, o seu começo será maior ou igual que a actividade precedente mais a sua duração.

$$t_i \geq t_j + d_i \quad \forall_{(j,i)\in A}$$

- **Restrições de não negatividade**: são as restrições que garantem que o conjunto de soluções válidas sejam positivas e, neste caso, maiores ou iguais que 0.

$$t_i \geq 0 \quad \forall i \in A$$

- **Restrições de integralidade**: são as restrições que definem o domínio do conjunto de soluções válidas.

$$t_i \in \mathbb{N}^+$$

$$y_{i,j} \in bin$$

- **Restrições de não simultaniedade**: são as restrições que impedem que as actividades que partilham o mesmo recurso não ocorram em paralelo.

  $$\begin{cases} \mathrm{t}_i + \mathrm{M} - \mathrm{M}y_{j,i} \geq \mathrm{t}_j + d_j \\ \mathrm{t}_j + \mathrm{M}y_{j,i} \geq \mathrm{t}_i + d_i \end{cases}$$

Com o auxílio da variável de decisão binária $y_{j,i}$, foi possível implementar as restrições de não simultaniedade e puder torna-las activas ou redundantes, sendo que o valor de **M** é escolhido de forma apropriada.

A primeira restrição representa a situação em que $y_{j,i}$ tem o valor de **1** , isto faz com que o valor de **M** se anule tornando a restrição redundante,e por consequência disso a segunda restrição será activa.

A segunda restrição representa a situação em que $y_{j,i}$ tem o valor de **0**. Quando isto acontece, a primeira restrição torna-se activa e a segunda redundante.

A combinação de ambas restrições restringe a simultaniedade das respectivas actividades.

Para o valor de **M** utilizou-se o valor de 100 visto que podemos observar que dificilmente uma actividade começará/terminará ao tempo de 100 ,i.e,dificilmente o projecto actual no total demorará mais que 100 [U.T].

3.3.3 Ficheiro de Input LP Solve

```
/* Funcao Objectivo */

   min: tfinal -tinicial ;

 /* Restricoes */

t0>= tinicial + 0;  t6>= tinicial + 0;  t1>= t0 + 4;

t4>= t0 + 4;        t3>= t1 + 6;        tfinal>= t3 + 2;

t3>= t4 + 9;        t5>= t4 + 9;         t3>= t5 + 4;

tfinal>= t5 + 4;    t4>= t6 + 5;         t8>= t6 + 5;

t10>= t6 + 5;       t5>= t8 + 4;         t9>= t8 + 4;

tfinal >= t9 + 2;   t8>= t10 + 8;        t11>= t10 + 8;

t9>= t11 + 7;


t8 + 100 - 100y48 >= t4 + 9;             t4 + 100y48 >= t8 + 4;

t11 + 100 - 100y411 >= t4 + 9;           t4 + 100y411 >= t11 + 7;

t11 + 100 - 100y811 >= t8 + 4;           t8 + 100y811 >= t11 + 7;

/* Restricoes de nao negatividade */

tinicial>=0   ,t0>=0  ,t1>=0   ,t3>=0    ,t4>=0

,t5>=0        ,t6>=0  ,t8>=0   ,t9>=0    ,t10>=0

,t11>=0      ,tfinal>=0 ;

/* Restricoes de integralidade */
int tinicial  ,t0  ,t1 ,t3,
    t4 ,t5  ,t6  ,t8  ,t9  ,t10  ,t11, tfinal;

bin y48,y411,y811;
```

3.3.4 Output do LP Solve

```

Value of objective function: 27.00

Actual values of the variables:

tfinal   = 27      tinicial   = 0      t0  = 0
t6   = 0           t1  = 19            t4  = 5
t3   = 25          t5  = 21            t8  = 14
t10  = 5           t9  = 25            t11 = 18
y48  = 1           y411 = 1            y811 = 1
```

3.3.5 Interpretação dos resultados

A dependência de recursos para a realização de actividades em paralelo fez com que a duração do projecto tivesse um acréscimo de 4 [U.T] , passando para um total de 27 [U.T]. Existem algumas alternativas para combater esse acréscimo de tempo , sendo que muitas vezes o stakeholder opta por contratar um empresa pelo exterior para a realização de uma das actividades ou pelo acréscimo do recurso partilhado , mas está decisão só processegue desde que seja atractiva. Relativamente as actividades selecionadas(4,8,11) podemos reparar no diagrama de gannt que nenhuma delas decorre em simultâneo , o que era espectável visto que elas obedecem as restrições estabelecidas.

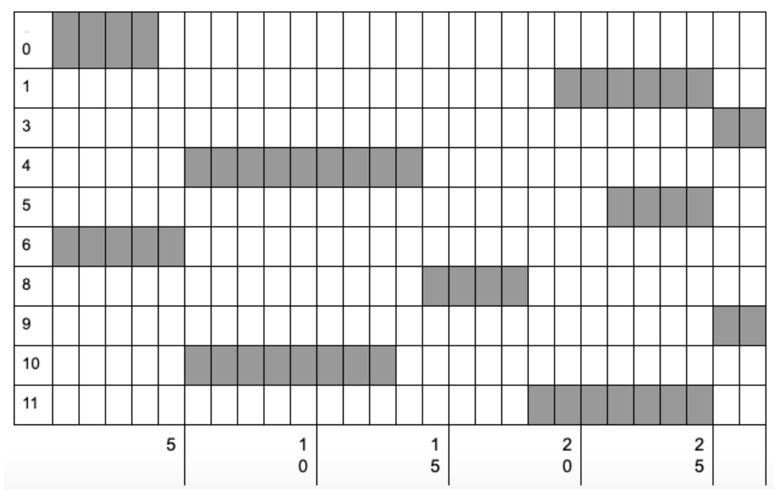

Figura 7: Diagrama de Gantt final

4 Parte II

4.1 Crashing Times

Crashing: é uma técnica de compactação de cronograma usada para reduzir ou encurtar o cronograma do projecto. A gestão de projectos pode tomar várias medidas para atingir esse objectivo . Alguns dos métodos comuns usados são:

- **Incluir recursos adicionais nas tarefas do caminho crítico**: Essa opção possui várias restrições, como a garantia do orçamento para adicionar os recursos e a disponibilidade dos recursos.
- **Reduzir os requisitos ou escopo do projecto**: Isso pode ser feito apenas se o patrocinador e as principais partes interessadas concordarem em reduzir o escopo.

Sendo assim, foi adoptado o primeiro método para reduzir o tempo do projecto em 3 [U.T], passando a ter uma duração máxima de 20 [U.T].

4.2 Análise do problema

Considerando que é possível, aumentar os recursos aplicados e com custos suplementares, reduzir a duração de uma actividade, num caso em que o custo da redução é não-linear. Um modelo com funções não lineares pode ser aproximado por um modelo em que cada uma dessas funções é aproximada por um função contínua linear por partes. No contexto do nosso problema , para aproximar a função não linear, vamos usar uma função linear com 2 partes. Os dados disponíveis para a realização deste problema são os seguintes:

Actividade Máx.red a custo c_1	Custo Normal c_2	c_1 Máx.red. a custo c2
0 0,5	400 100	200 0,5
1 1	1000 300	600 1
3 0,5	300 100	200 0,5
4 2	2000 400	800 1
5 0,5	1000 800	1600 0,5
6 1	800 90	180 1
8 0,5	600 100	200 0,5
9 0	300 -	- 0
10 0,5	1600 500	1000 0,5
11 1	1400 300	600 1

Na variante em análise, cada actividade tem cinco parâmetros adicionais:

1. **Custo Normal**: o custo normal associado a actividade.[U.M]
2. **C1** : o custo suplementar de reduzir a duração da actividade em uma unidade de tempo.[U.M/U.T].
3. **Máxima redução** C_1: é o valor da máxima redução de tempo a um custo c1.
4. **C2**: o custo suplementar de reduzir a duração da actividade em uma unidade de tempo [U.T] após ter aplicado a máxima redução a um custo c_1.[U.M/U.T].
5. **Máxima redução** C_2: é o valor da máxima redução de tempo a um custo c2.

4.3 Solução do problema

4.3.1 Variáveis de decisão

Para a resolução deste problema foram utilizadas as seguintes variáveis de decisão:

- r_is: segunda redução de duração da actividade i, com i $\in\{0,1,3,4,5,6,8,9,10,11\}$.

- $\begin{cases} s_i{=}1 & \text{, se a segunda redução for aplicada com i } \in\{0,1,3,4,5,6,8,9,10,11\}. \\ s_i{=}0 & \text{, se a segunda redução não for aplicada com i } \in\{0,1,3,4,5,6,8,9,10,11\}. \end{cases}$

- rr_i: resto da primeira redução em relação à redução máxima permitida com i $\in\{0,1,3,4,5,6,8,9,10,11\}$.

- $t_{inicial}$: instante de tempo em que o projecto se inicia [U.T].

- t_i: tempo de início de execução da actividade i, com i $\in\{0,1,3,4,5,6,8,9,10,11\}$.

- t_{final}: instante de tempo em que o projecto termina [U.T]

- r_ip: primeira redução de duração da actividade i, com i $\in\{0,1,3,4,5,6,8,9,10,11\}$.

4.3.2 Restrições

As restrições implementadas ditam as regras para o correto funcionamento do sistema. Com base nisso, dividiu-se as restrições em três grupos:

- **Restrições de precedência**: são as restrições que garantem que para toda actividade i que sucede a actividade j, o seu começo será maior ou igual que a actividade precedente mais a sua duração.

$$t_i \geq t_j + d_i \quad \forall_{(j,i)\in A}$$

- **Restrições de não negatividade**: são as restrições que garantem que o conjunto de soluções válidas sejam positivas e, neste caso, maiores ou iguais que 0.

$$t_i \geq 0 \quad \forall i \in V$$

- **Restrições de integralidade**: são as restrições que definem o domínio do conjunto de soluções válidas.

$$t_i \in \mathbb{N}^+$$

$$s_i \in bin$$

Nesta fase é essencial reforçar que o objectivo principal é reduzir o tempo de execução do projecto encontrado na parte 0 em 3 U.T. Sendo assim, esse objectivo faz com que surja uma restrição que obriga que o tempo máximo para conclusão do projecto seja descrito do seguinte modo:

- $t_{final} - t_{inicial} \leq 20$.

Os restos das reduções máximas (rr_i) permitidas no projecto referentes à primeira redução caso a segunda redução seja realizada tem de ser igual a 0.

- $rr_i \leq$ Máxima redução $\mathrm{C}_{1,i}$ $-$ Máxima redução$C_{1,i}s_i$

De seguida separou-se as reduções máximas permitidas no projecto em dois grupos :

- **Primeira redução**: $r_ip + rr_i =$ Máxima redução$C_{1,i}$
- **Segunda redução**: $r_is \leq$ Máxima redução$C_{2,i}S_i$

As relações de precedência implementadas na parte 0 são modificadas , sendo adicionadas nelas os dois grupos de reduções máximas, obtendo o seguinte :

- $t_i + r_jp + r_js \geq t_j + d_j$

4.3.3 Ficheiro de Input LP Solve

```

/* Funcao Objectivo */
min : 200 r0p + 600r1p +200r3p + 800r4p +1600r5p + 180r6p + 200r8p + 0r9p + 1000r10p + 600r11pp + 100r0s + 300r1s + 100r3s + 400r4s + 800r5s + 90r6s + 100r8s + 0r9s + 500r10s + 300r11s ;

/* Restricoes */

/* Tempo m ximo para conclus o do projecto */

tfinal - tinicial <= 20;

/* Reducoes maximas permitidas no projecto */

// Primeira reducao

r0p + rr0 = 0.5;   r1p + rr1 = 1;
r3p + rr3 = 0.5;   r4p + rr4 = 2;
r5p + rr5 = 0.5;   r6p + rr6 = 1;
r8p + rr8 = 0.5;   r9p + rr9 = 0;
r10p + rr10 = 0.5; r11p + rr1 = 1;

// Segunda reducao

r0s <= 0.5s0;    r1s<= 1s1;
r3s<= 0.5s3;     r4s<= 2s4;
r5s<= 0.5s5;     r6s<= 1s6;
r8s<= 0.5s8;     r9s<= 0s9;
r10s<= 0.5s10;   r11s<= 1s11;


/* Resto das reducoes maxima no projecto referentes a primeira reducao caso a segunda reducao seja realizada tem de ser igual a 0.*/

rr0 <= 0.5 -0.5s0;    rr1 <= 1 - 1s1; rr3 <= 0.5 - 0.5s3;   rr4 <= 2 - 2s4;
rr5 <= 0.5 - 0.5s5;   rr6 <= 1 - 1s6;
rr8 <= 0.5 - 0.5s8;   rr9 <= 0 - 0s9;
rr10 <= 0.5 - 0.5s10; rr11 <= 1 - 1s11;

/* Relacoes de precedencia */

t0 >= tinicial + 0;
t6 >= tinicial + 0;
t1 + r0p + r0s >= t0 + 4;
t4  + r0p + r0s >=t0 + 4;
```

```
t3  + r1p + r1s >= t1 + 6;
tfinal  + r3p + r3s >= t3 + 2;
t3  + r4p + r4s >= t4 + 9;
t5  + r4p + r4s >= t4 + 9;
t3  + r5p + r5s >= t5 + 4;
tfinal  + r5p + r5s >= t5 + 4;
t4  + r6p + r6s >= t6 + 5;
t8  + r6p + r6s >= t6 + 5;
t10  + r6p + r6s >= t6 + 5;
t5  + r8p + r8s >= t8 + 4;
t9  + r8p + r8s >= t8 + 4;
tfinal  + r9p + r9s >= t9 + 2;
t8  + r10p + r10s >= t10 + 8;
t9  + r11p + r11s >= t11 + 7;
t11  + r10p + r10s >= t10 +8;

/* Restricoes de integralidade */
bin s0,s1,s3,s4,s5,s6,s8,s9,s10,s11;
```

4.3.4 Output do LP Solve

```

Value of objective function: 420.00

Actual values of the variables:

tinicial = 0      tfinal   = 20          t0  = 0
t1   =  4          t3  = 19              t4  = 6
t5   =  15         t6  = 0               t8  = 11
t9   =  18         t10 = 3               t11 = 11

r0p  =  0          r1p = 0               r3p = 0.5
r4p  =  0          r5p = 0               r6p = 1
r8p  =  0          r9p = 0               r10p = 0
r11p =  0

r0s  =  0          r1s  = 0              r3s  =  0.5
r4s  =  0          r5s  = 0              r6s  =  1
r8s  =  0          r9s  = 0              r10s =  0
r11s =  0

rr0  =  0.5        rr1  = 1              rr3  =  0
rr4  =  2          rr5  = 0.5            rr6  =  0
rr8  =  0.5        rr9  = 0              rr10 =  0.5
rr11 =  1

s0   =  0          s1   = 0              s3   =  1
s4   =  0          s5   = 0              s6   =  1
s8   =  0          s9   = 0              s10  =  0
s11  =  0
```

4.3.5 Interpretação dos resultados

Com base no output produzido podemos constatar que a duração do projecto sofreu a alteração desejada, tendo como tempo máximo 20 U.T. Essa redução do tempo máximo foi derivada do facto de reduzir em uma unidade de tempo as durações da actividade 3 e 6.

Para verificar o custo da solução podemos calcular a função objectivo com base na solução obtida:

$200 * 0 + 600 * 0 + 200 * 0.5 + 800 * 0 + 1600 * 0 + 180 * 1 + 200 * 0 + 0 * 0 + 1000 * 0 + 600 * 0 + 100 * 0 + 300 * 0 + 100 * 0.5 + 400 * 0 + 800 * 0 + 90 * 1 + 100 * 0 + 0 * 0 + 500 * 0 + 300 * 0 = 420$;

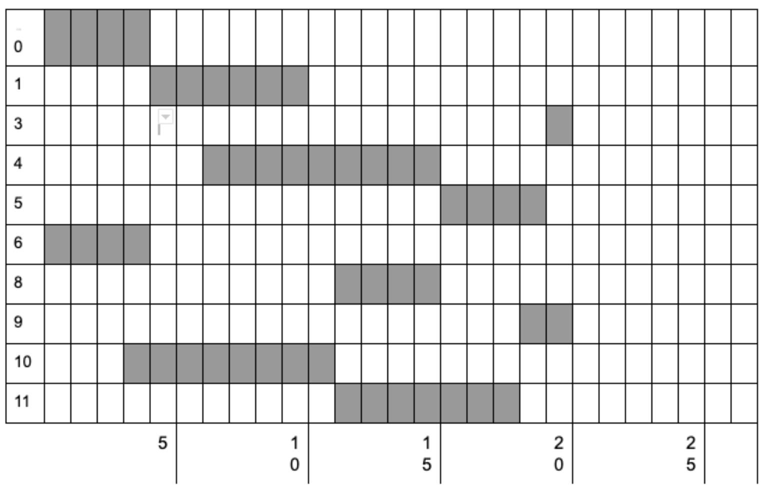

Figura 8: Diagrama de Gantt após reduções