**Relatório Trabalho Prático 3**

Modelos Determinísticos de Investigação Operacional

Mestrado Integrado em Engenharia Informática

Braga, 5 de janeiro de 2017

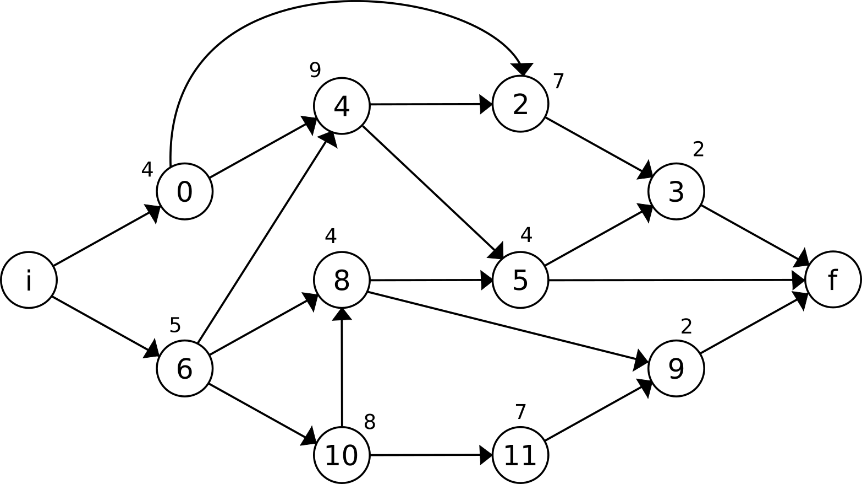
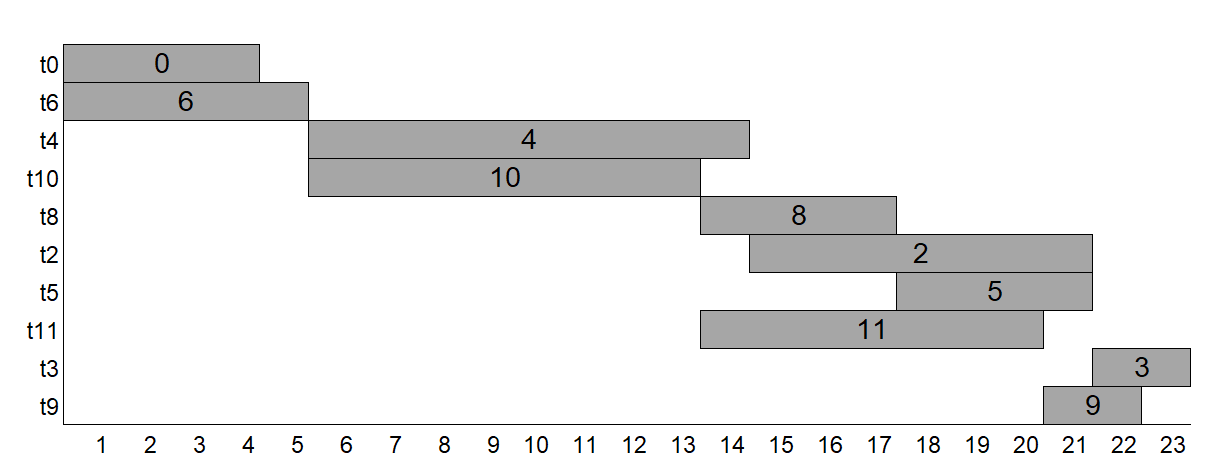
**Grupo 33**

Francisco Oliveira – 78416

Raul Vilas Boas – 79617

Vitor Peixoto – 79175

**PARTE I**

1. Após eliminar as atividades indicadas na secção “Determinação da lista de atividades” com base no número de aluno ABCDE = 79617, obteve-se o seguinte grafo.
2. As atividades escolhidas que decorrem em paralelo foram as atividades 2, 5 e 9.
3. O objetivo do problema é realizar o projeto no menor tempo possível tendo em conta estas novas situações. Por isso, a função objetivo continua a ser minimizar o tempo final do projeto.

Visto que, as tarefas não podem ser realizadas ao mesmo tempo por causa de requererem o uso da mesma máquina foi necessário criar novas restrições. No entanto, conforme dito no enunciado, umas destas atividades pode ser contrada pelo exterior para ser realizada, apesar de que a duração desta é aumentada em 1 U.T.

Para tratar da questão da contratação da empresa criou-se uma variável binária *vi* que toma o valor de 1 se a atividade for realizada pela empresa e 0 caso contrário. Para além disto, visto que apenas uma delas pode ser realizada pela empresa teve-se que criar outra restrição em que a soma deles tem que ser menor ou igual a 1.

*v2 + v5 + v9 <= 1;*

*Bin v2 v5 v9;*

A contratação de uma certa atividade por uma empresa exterior um aumento da duração da atividade numa unidade. Por este motivo, nas restrições de tempo onde aparecem as atividades em causa é adicionado a variável *vi* da respetiva atividade porque assim se essa atividade for realizada pela empresa, isto é, o seu valor é 1, o tempo da atividade vai aumentar numa unidade.

*arco\_23: t3 >= t2 + 7 + v2 ;*

*arco\_53: t3 >= t5 + 4 + v5;*

*arco\_5f: tf >= t5 + 4 + v5;*

*arco\_9f: tf >= t9 + 2 + v9;*

Como foram escolhidas 3 atividades, existem 6 situações diferentes que podem ocorrer.

Ex: A atividade 2 precede a atividade 5 e a atividade 9 é realizada pela empresa (caso 1).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Atividades | Primeira | Segunda | Empresa |
| Caso 1 | 2 | x |  |  |
| 5 |  | x |  |
| 9 |  |  | x |
| Caso 2 | 2 |  | x |  |
| 5 | x |  |  |
| 9 |  |  | x |
| Caso 3 | 2 | x |  |  |
| 5 |  |  | x |
| 9 |  | x |  |
| Caso 4 | 2 |  | x |  |
| 5 |  |  | x |
| 9 | x |  |  |
| Caso 5 | 2 |  |  | x |
| 5 | x |  |  |
| 9 |  | x |  |
| Caso 6 | 2 |  |  | x |
| 5 |  | x |  |
| 9 | x |  |  |

Para estas situações, criou-se uma variável binária yij que representa que atividades precedem quais. Por exemplo, *y25* é destinado ás situações em que a atividade 2 precede a atividade 5 ou caso contrário. Ao esta variável ter o valor de 1 representa as situação em que a atividade 2 precede a 5 e quando tem o valor de 0 é quando a atividade 5 precede a 2.

Logo, criou-se as seguintes restrições para estes acontecimentos.

*t5 + M - M y25 + M v2 + M v5 >= t2 + 7;*

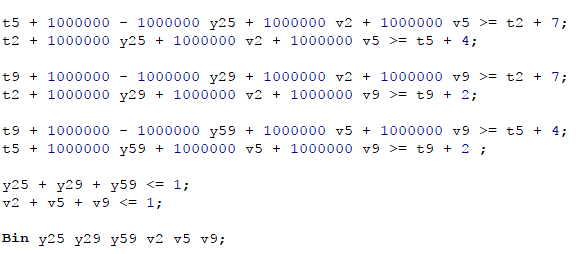
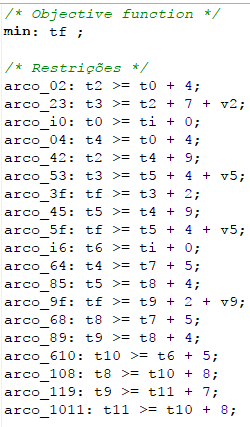
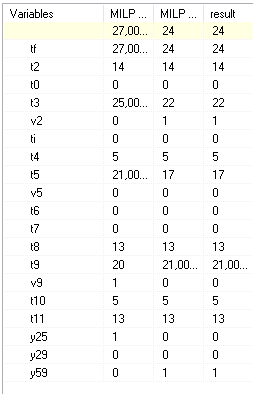
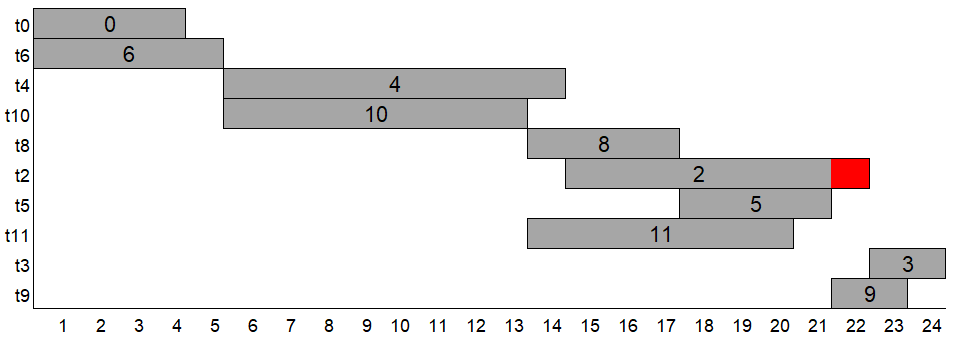
*t2 + M y25 + M v2 + M v5 >= t5 + 4;*

A primeira representa a situação em que y25 tem o valor de um, isto faz com que o valor de M se anule tornando a restrição insignificante. Isto vai implicar que o tempo da atividade 5 seja dependente do tempo da atividade 2, isto é, só pode ser realizada após a atividade 2 ser realizada *(t5 >= t2 + 7)*.

A segunda representa a situação em que y25 tem o valor de zero. Quando isto acontece, a primeira restrição torna-se insignificante pois *M >= t2 + 7* não restringe nada. O que faz com que a segunda restrição possa ter efeito, ficando *t2 >= t5 + 4* que representa que a atividade 2 só pode acontecer quando a atividade 5 acontecer, isto é, 5 precede 2.

Para além disto, em ambas as restrições tem que existir a certeza que nenhuma delas é realizada pela empresa, pois caso isto acontece elas deixam de restringir, por causa dos *M v2 + M v5*.

No LPSolve utilizou-se o valor de 1 milhão para o valor do M.

1. 
2. O resoltado obtido foi 24 porque é o tempo menor possível.
3. Novo diagrama de Grant do projeto.

O diagrama obedece ás restrições porque como a tarefa 5 precede a 9, a atividade 9 só pode acontecer depois da atividade 5 e é isso que acontece no diagrama.

Para além disto como a atividade 2 foi realizada no exterior levou ao aumenta de uma unidade de tempo que por consequência levou ao aumento do tempo final. Isto porque como a atividade 3 dependia da atividade 2 esta atrasou-se numa unidade também.

**PARTE II**

O objetivo do problema é reduzir o tempo de execução encontrado na parte I do trabalho 2 em 4 U.T. com um custo suplementar mínimo. Sendo que o tempo de obtido foi de 24 U.T. a nova solução terá de ter um tempo de execução igual a 19 U.T., reduzindo o tempo nas atividades necessárias.

1. Para isso, criou se uma nova variável de decisão *rij*, que representa o número de reduções que ocorrem na atividade i de custo c1 ou c2.

Ex: *r21*: número de reduções da atividade 2 com o custo de 600 U.M (c1).

Ex: *r22:* número de reduções da atividade 2 com custo de 400 U.M (c2).

Visto que, se trata de um problema de programação inteira e não pode haver variáveis com valores fracionários, nas atividades onde a redução máxima é de 0.5 (atividade 0, 3, 5, 8, 10) juntou-se os custos c1 e c2 fazendo com que a redução máxima passasse a ser 1, para isso então utilizou-se a variável *ri.*

Ex: *r3* -> número de reduções da atividade 3.

Obtendo assim a função objetivo apresentada abaixo.

*Min z: min: 200 r0 + 600 r21 + 400 r22 + 220 r3 + 600 r41 + 400 r42 + 1600 r5 + 120 r61 + 90 r62*

*+ 210 r8 + 900 r10 + 400 r111 + 200 r112;*

O tempo máximo de execução do projeto tem de ser 19 U.T.

*tf = 19;*

As restrições relativas aos arcos também tiveram de ser alteradas, para acrescentar a variável *rij* e *ri* pois, os valores dos tempos podem ser reduzidos devido à variável da respetiva atividade.

|  |  |
| --- | --- |
| *t2 >= t0 - r0 + 4;* | *t4 >= t6 - r61 - r62 + 6;* |
| *t3 >= t2 - r21 - r22 + 7;* | *t5 >= t8 - r8 + 4;* |
| *t0 >= ti + 0;* | *tf >= t9 + 2;* |
| *t4 >= t0 - r01 - r02 + 4;* | *t8 >= t6 - r61 - r62 + 6;* |
| *t2 >= t4 - r41 - r42 + 9;* | *t9 >= t8 - r8 + 4;* |
| *t3 >= t5 - r5 + 4;* | *t10 >= t6 - r61 - r62 + 5;* |
| *tf >= t3 - r3 + 2;* | *t8 >= t10 - r10 + 8;* |
| *t5 >= t4 - r41 - r42 + 9;* | *t9 >= t11 - r111 - r112 + 7;* |
| *tf >= t5 - r5 + 4;* | *t11 >= t10 - r10 + 8;* |
| *t6 >= ti + 0;* | *t4 >= t6 - r61 - r62 + 6;* |

Para além disso, é necessário introduzir outras restrições porque as reduções *ri2* não podem acontecer caso não se atinga a máxima redução com o custo c1. Por esse motivo, acrescentou-se as seguintes variáveis para os casos mais simples.

*r61 >= r62;*

*r111 >= r112;*

Esta restrição faz com que a redução *r62* só possa acontecer caso a redução r61 tenha ocorrido, tal como é descrito no enunciado. O mesmo se verifica com a atividade 11.

|  |  |
| --- | --- |
| r61= 0 -> r62 = 0, | 0 >= 0 |
| r61= 1 -> r62 = 0 ∨ r62 = 1, | 1 >= 0 ∨ 1 >= 1 |

No entanto, estas restrições apenas funcionam para os casos em que a redução máxima de custo c1 e c2 é 1. Para o caso das atividades 2 e 4 é necessário utilizar uma variável binária como auxiliar.

No caso da atividade 2, a redução *r22* só pode acorrer quando o valor da redução *r21* seja 3. Para isso acrescentou-se as seguintes restrições:

*r21 >= 3 y2;*

*r22 <= 1000000 y2;*

A restrição *r21 >= 3 y2* permite que a variável binária *y2* apenas seja 1 quando a redução *r21* seja 3. Como comprovado pelos seguintes exemplos.

|  |  |
| --- | --- |
| r21= 0 -> y2 = 0, | 0 >= 0 |
| r21= 1 -> y2 = 0, | 1 >= 0 |
| r21= 2 -> y2 = 0, | 2 >= 0 |
| r21= 3 -> y2 = 0 ∨ y2 = 1, | 3 >= 0 ∨ 3 >= 3 |

Visto que, a variável binária *y2* representa se a redução máxima de custo c1 ocorre então temos de permitir que a variável *r22* tenha a possibilidade de ser reduzida. Para isso utiliza-se um valor bastante elevado, normalmente denominado por *M*, em que neste caso foi usado 1 milhão.

y2 = 0 -> r22 <= 0

y2 = 1 -> r22 <= 1000000

Estas restrição fazem com que, se *y2* for zero r22 vai ser zero também, enquanto que se *y2* for 1, *r22* vai poder tomar qualquer valor até M.

O mesmo acontece com a atividade 4, em que se utilizou as seguintes restrições:

*r41 >= y4;*

*r42 <= 1000000 y4;*

Por último, adicionou-se as restrições relativas ás máximas reduções permitidas de cada redução.

*r0 <= 1;*

*r21 <= 3;*

*r22 <= 1;*

*r3 <= 1;*

*r41 <= 1;*

*r42 <= 2;*

*r5 <= 1;*

*r61 <= 1;*

*r62 <= 1;*

*r8 <= 1;*

*r10 <= 1;*

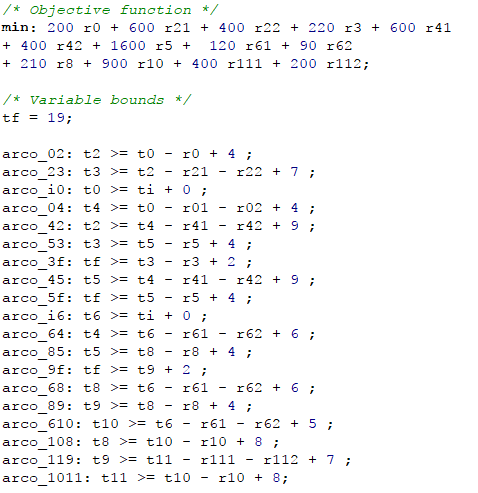
*r111 <= 1;*

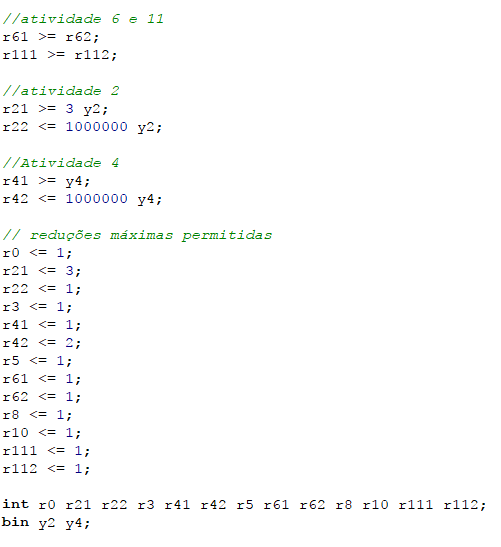
*r112 <= 1;*

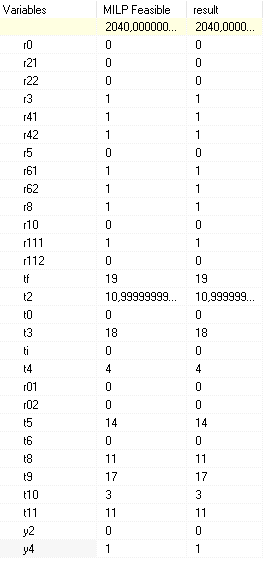
Como é um problema de programação inteira é necessário que as variáveis não tenham valores inteiros, por isso definiu-se as como inteiras e as variáveis y2 e y4 como binárias.

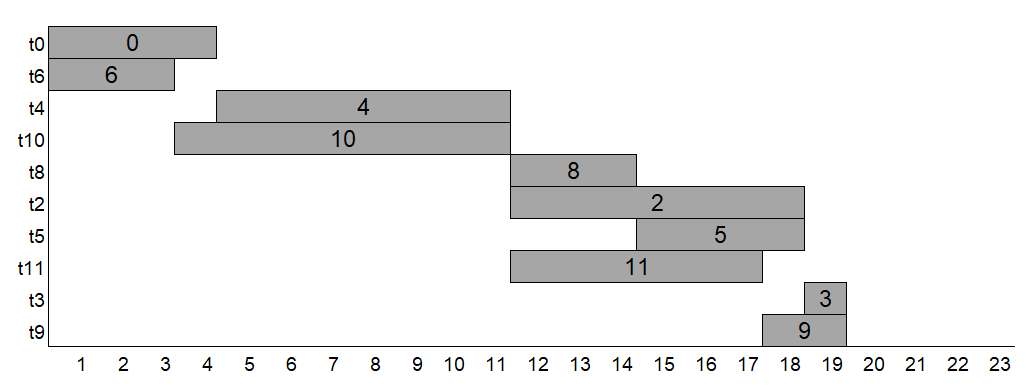
*int r0 r21 r22 r3 r41 r42 r5 r61 r62 r8 r10 r111 r112;*

*bin y2 y4;*

1. 



1. 
2. Diagrama de Gantt após serem efetuadas as reduções.



1. Após uma análise ao grafo, conclui-se que iria ser necessário ocorrer 7 reduções, duas na atividade 4, outras duas na atividade 6 e uma nas atividades 3, 8 e 11.

Sendo assim, somou-se os custos destas reduções tento em conta os c1 e c2 que influenciavam algumas destas reduções.

Custo = (120 + 100) r3 + 600 r41+ 400 r42 + 120 r61 + 90 r62 + (120 + 90) r8 + 400 r11

Visto que, para todas estas variáveis o número de redução foi para todas uma, então os valores de r3, r41, r42, r61, r62, r8 e r11 são todos um.

Obtendo assim um custo de 2040 que é o mesmo obtido pelo LPSolve, sendo que se verifica que o custo da solução está correto.