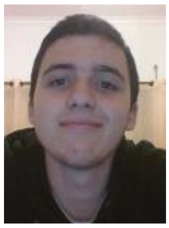
Universidade do Minho
Escola de Engenharia
Departamento de Informática
Mestrado Integrado em Engenharia Informática

Modelos Determinísticos de Investigação Operacional

Trabalho Prático Nº. 2

Braga, 24 de Novembro de 2015

Pedro Vieira Fortes
Nº64309



João Pedro Pereira Fontes
Nº71184

Resumo

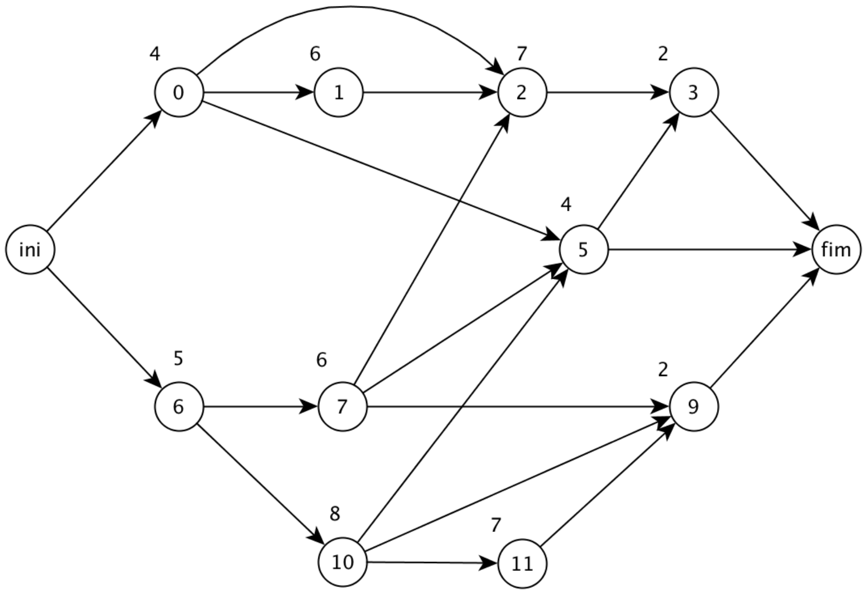
Este relatório tem como intuito apresentar a análise e resolução do trabalho que nos foi proposto na unidade curricular de Modelos Determinísticos de Investigação Operacional pelo professor Valério Carvalho. Neste trabalho deveríamos resolver um caso de gestão de projetos com a ajuda das ferramentas LPSolve e Relax4 aplicando os conhecimentos adquiridos na unidade curricular.

Índice

Parte I

1.1. Rede do Projeto

Através das instruções para a determinação da nossa rede em estudo, obtemos o seguinte resultado:



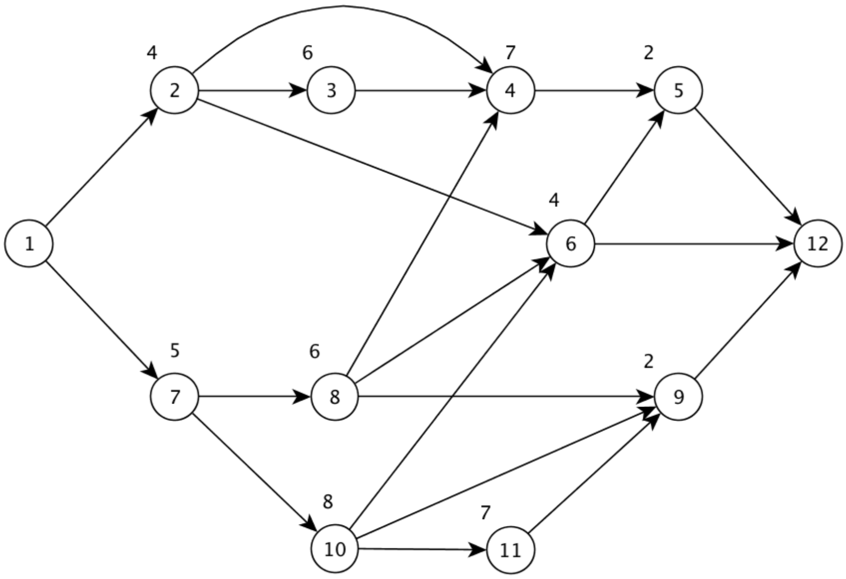
Esta rede é resultante da rede original do enunciado retirando o nodo 4 e 8 que corresponde aos últimos 2 dígitos do aluno com maior número de inscrição (71184).

1.2. Formulação do Problema

Para este trabalho, devido à incompatibilidade do relax em processar nodos com índice zero, nós redefinimos os nodos da seguinte forma:

Nodo Original	Novo Nodo
ini	1
0	2
1	3
2	4
3	5
5	6
6	7
7	8
9	9
10	10
11	11
fim	12

Desta forma obtemos a seguinte rede para o Relax:



- **Objetivo**

O objetivo deste modelo é encontrar o caminho mais longo (caminho crítico) entre os nodos 1 e 12, escolhendo os nodos de 2 a 11, que resultam um maior peso no final. Este modelo foi realizado baseado na rede apresentada em cima.

- **Variáveis de Decisão**

As nossas variáveis de decisão são binárias, ou seja, será atribuído o valor 0 ou 1 aos vértices vi_ j sendo i o nodo origem e j o nodo destino (tal que i, j ϵ {1,2,3,4,5,6,7,8,9,10,11,12}). Assim, para que o modelo seja resolvido é atribuído o valor 0 ou 1 às arestas do caminho, conforme estas constam ou não para um caminho mais crítico.

- **Função Objetivo**

Tem-se então a seguinte função objetivo:

$$\begin{aligned} min\text{: } & 0\,v1\_2 - 0\,v1\_7 - 4\,v2\_3 - 4\,v2\_4 - 4\,v2\_6 - 6\,v3\_4 - 7\,v4\_5 \\ & - 2\,v5\_12 - 4\,v6\_5 - 4\,v6\_12 - 5\,v7\_8 - 5\,v7\_10 \\ & - 6\,v8\_4 - 6\,v8\_6 - 6\,v8\_9 - 2\,v9\_12 - 8\,v10\_6 \\ & - 8\,v10\_9 - 8\,v10\_11 - 7\,v11\_9; \end{aligned}$$

O objetivo é maximizar o custo, encontrando assim o caminho mais longo. No entanto, como o custo implica a saída de unidades monetárias, colocamos os coeficientes negativos e minimizamos a função objetivo. Cada parcela da função objetivo representa o custo em tempo de uma determinada aresta. Por exemplo, 4*v2_3 representa o custo de 4 U.T em escolher o caminho que se inicia no nodo 2 e termina no nodo 3.

- **Restrições**

O problema em questão exigia as seguintes restrições:

$$v1\_2 + v1\_7 = 1$$
$$-v1\_2 + v2\_3 + v2\_4 + v2\_6 = 0$$
$$-v2\_3 + v3\_4 = 0$$
$$-v2\_4 - v3\_4 - v8\_4 + v4\_5 = 0$$
$$-v4\_5 - v6\_5 + v5\_12 = 0$$
$$-v2\_6 - v8\_6 - v10\_6 + v6\_5 + v6\_12 = 0$$
$$-v1\_7 + v7\_8 + v7\_10 = 0$$
$$-v7\_8 + v8\_4 + v8\_6 + v8\_9 = 0$$
$$-v8\_9 - v10\_9 - v11\_9 + v9\_12 = 0$$
$$-v7\_10 + v10\_6 + v10\_9 + v10\_11 = 0$$
$$-v10\_11 + v11\_9 = 0$$
$$v5\_12 + v6\_12 + v9\_12 = 1$$

A primeira e última restrições, obrigam que o fluxo de entrada (nodo 1) e saída (nodo 12) sejam igual a 1, pois é injetada na rede uma unidade de fluxo pelo nodo 1 (oferta) e essa unidade sai pelo nodo 12 (procura). As restantes restrições servem para obrigar a que o fluxo de entrada dos nodos seja igual ao fluxo de saída não tendo qualquer oferta ou procura. É de notar que não foi necessário definir limitações no fluxo dos arcos, sendo estas ilimitadas.

1.3. Input no Relax

O seguinte input foi realizado num editor de texto para execução no Relax4:

```
12
20
1	2	0	1000
1	7	0	1000
2	3	-4	1000
2	4	-4	1000
2	6	-4	1000
3	4	-6	1000
4	5	-7	1000
5	12	-2	1000
6	5	-4	1000
6	12	-4	1000
7	8	-5	1000
7	10	-5	1000
8	4	-6	1000
8	6	-6	1000
8	9	-6	1000
9	12	-2	1000
10	6	-8	1000
10	9	-8	1000
10	11	-8	1000
11	9	-7	1000
1
0
0
0
0
0
0
0
0
0
0
-1
```

1.4. Output do Relax4

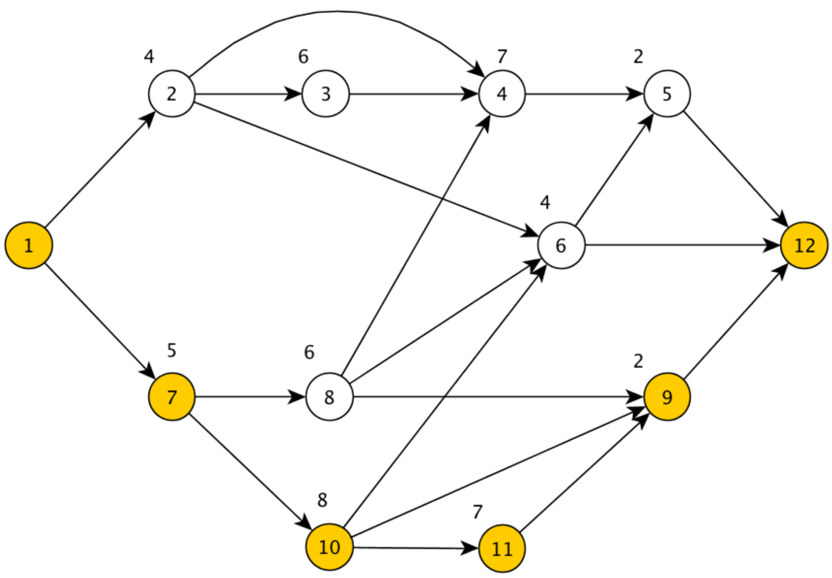
Este foi o output resultante do input inserido no Relax4:

```
END OF READING
 NUMBER OF NODES = 12, NUMBER OF ARCS = 20
 CONSTRUCT LINKED LISTS FOR THE PROBLEM
 CALLING RELAX4 TO SOLVE THE PROBLEM
 ***********************************
 TOTAL SOLUTION TIME =  0. SECS.
 TIME IN INITIALIZATION =  0. SECS.
   1 7  1.
   7 10  1.
   9 12  1.
   10 11  1.
   11 9  1.
 OPTIMAL COST =  -22.
 NUMBER OF AUCTION/SHORTEST PATH ITERATIONS = 37
 NUMBER OF ITERATIONS =  8
 NUMBER OF MULTINODE ITERATIONS =  1
 NUMBER OF MULTINODE ASCENT STEPS =  0
 NUMBER OF REGULAR AUGMENTATIONS =  1
 ***********************************
```

Como podemos ver, foi possível obter um output do programa, ou seja, o problema é possível tendo uma solução ótima de 22 U.T. (o output deu um número negativo exatamente pelo que foi referido na função objetivo).

1.5. Interpretação da Solução e Caminho Crítico

Como se pode verificar pelo output fornecido pelo Relax4, ele interpretou que tínhamos 12 nodos e 20 arcos tal como fornecemos no input e demorou menos de 1 segundo a calcular a solução para o nosso problema (caminho crítico), sendo que a unidade injetada no nodo 1 vai até ao nodo 12 (nodo de procura) passando pelos nodos na seguinte ordem: 7, 10, 11 e 9. O custo ótimo resultante foi de 22 U.T. As restantes considerações do output referem-se ao nº de iterações necessárias para chegar à solução. Então, o caminho crítico resultante na rede é a seguinte:



Parte II

2.1. Formulação do Modelo Primal

O nosso modelo do trabalho 1 tinha como variáveis de decisão ti, no qual representa o tempo acumulado até o nodo i (tal que i ϵ {0,1,2,3,5,6,7,9,10,11,f}). A nossa função objetivo era a seguinte:

$$min: tf;$$

O tf representa o tempo total de execução das atividades pertencentes ao caminho crítico, respeitando as suas devidas precedências. As nossas restrições eram as seguintes:

$$t0 = 0;$$
$$t6 = 0;$$
$$t1 >= t0 + 4;$$
$$t2 >= t0 + 4;$$
$$t2 >= t1 + 6;$$
$$t2 >= t7 + 6;$$
$$t3 >= t2 + 7;$$
$$t3 >= t5 + 4;$$
$$t5 >= t0 + 4;$$
$$t5 >= t7 + 6;$$
$$t5 >= t10 + 8;$$
$$t7 >= t6 + 5;$$
$$t9 >= t7 + 6;$$
$$t9 >= t10 + 8;$$
$$t9 >= t11 + 7;$$
$$t10 >= t6 + 5;$$
$$t11 >= t10 + 8;$$
$$tf >= t3 + 2;$$
$$tf >= t5 + 4;$$
$$tf \geq t9 + 2;$$

2.2. Formulação do Modelo Dual

Para a resolução desta parte, recorremos ao programa LPSolve utilizado no Trabalho 1 para consultar a matriz solução do modelo apresentado na alinha 1 (modelo primal) desta parte:

	tf	t0	t6	t1	t2	t7	t3	t5	t10	t9	t11	RHS
min	1	0	0	0	0	0	0	0	0	0	0	0
R1	0	-1	0	1	0	0	0	0	0	0	0	4
R2	0	-1	0	0	1	0	0	0	0	0	0	4
R3	0	0	0	-1	1	0	0	0	0	0	0	6
R4	0	0	0	0	1	-1	0	0	0	0	0	6
R5	0	0	0	0	-1	0	1	0	0	0	0	7
R6	0	0	0	0	0	0	1	-1	0	0	0	4
R7	0	-1	0	0	0	0	0	1	0	0	0	4
R8	0	0	0	0	0	-1	0	1	0	0	0	6
R9	0	0	0	0	0	0	0	1	-1	0	0	8
R10	0	0	-1	0	0	1	0	0	0	0	0	5
R11	0	0	0	0	0	-1	0	0	0	1	0	6
R12	0	0	0	0	0	0	0	0	-1	1	0	8
R13	0	0	0	0	0	0	0	0	0	1	-1	7
R14	0	0	-1	0	0	0	0	0	1	0	0	5
R15	0	0	0	0	0	0	0	0	-1	0	1	8
R16	1	0	0	0	0	0	-1	0	0	0	0	2
R17	1	0	0	0	0	0	0	-1	0	0	0	4
R18	1	0	0	0	0	0	0	0	0	-1	0	2

Através desta matriz do modelo primal em questão, é fácil construir o modelo dual deste, sendo que:

- ✓ Cada restrição Ri corresponde a uma variável yi (tal que i ϵ [0,17]);
- ✓ A coluna RHS corresponde aos coeficientes da função objetivo;
- ✓ A linha correspondente ao min representa o 2º membro para cada restrição e todas as restantes linhas correspondem ao 1º membro.

Vejamos então o modelo dual conseguido:

- **Função Objetivo:**

$$max: 4\,y1 \;+\; 4\,y2 \;+\; 6\,y3 \;+\; 6\,y4 \;+\; 7\,y5 \;+\; 4\,y6 \;+\; 4\,y7 \;+\; 6\,y8 \;+\; 8\,y9 \;+\; 5\,y10 \;+\; 6\,y11 \;+\; 8\,y12 \;+\; 7\,y13 \;+\; 5\,y14 \;+\; 8\,y15 \;+\; 2\,y16 \;+\; 4\,y17 \;+\; 2\,y18;$$

- **Restrições:**

t0 → $-y1 - y2 - y7 \;<=\; 0$

t1 → $y1 \;-\; y3 \;\leq 0$

t2 → $y2 \;+\; y3 \;+\; y4 \;-\; y5 \;\leq 0$

t3 → $y5 \;+\; y6 \;-\; y16 \;\leq 0$

t5 → $-y6 \;+\; y7 \;+\; y8 \;+\; y9 \;-\; y17 \;\leq 0$

t6 → $-y10 \;-\; y14 \;\leq 0$

t7 → $-y4 \;-\; y8 \;+\; y10 \;-\; y11 \;\leq\; 0$

t9 → $y11 \;+\; y12 \;+\; y13 \;-\; y18 \;\leq 0$

t10 → $-y9 \;-\; y12 \;+\; y14 \;-\; y15 \;\leq 0$

t11 → $-y13 \;+\; y15 \;\leq 0$

tf → $y16 \;+\; y17 \;+\; y18 \;<=\; 1$

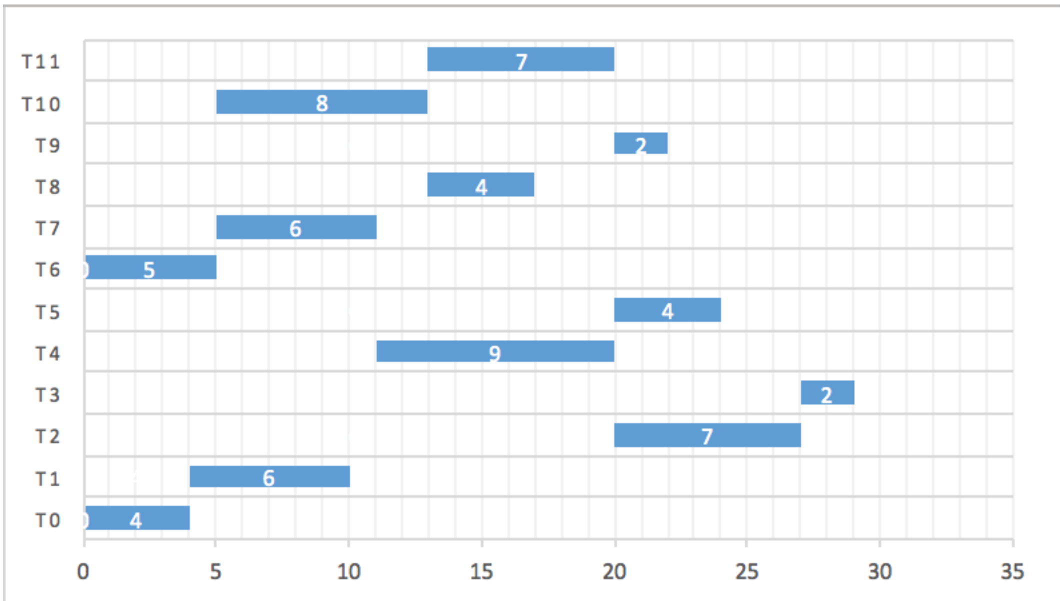
Para conferir se o modelo estava correto, utilizamos o LPSolve e obtemos o seguinte resultado:

Parte 2	
y1	0
y2	0
y3	0
y4	0
y5	0
y6	0
y7	0
y8	0
y9	0
y10	0
y11	0
y12	0
y13	1
y14	1
y15	1
y16	0
y17	0
y18	1
	22

Com isto confirmamos que o resultado do modelo dual é igual ao obtido no modelo primal.

Parte III

3.1. Just-In-Time



A partir do diagrama de gant da rede original, é possível observar que a atividade 1 pode ter início com o tempo de 14 U.T., porque a atividade 2 tem como precedências a própria atividade 1 e a atividade 4, e para que a atividade 2 respeite as precedências da atividade 4, este só pode iniciar no instante de tempo de 20 U.T. (soma dos tempos das precedências de 4 e do próprio 4). Como a atividade 1 tem uma duração de 6 U.T., este deve iniciar pelo menos nas 14 U.T. para que respeite a filosofia do just-in-time aplicada na atividade 2 (20 - 6 = 14).

3.2. Formulação com Just-In-Time

Escolhido o par de atividades 7 e 5 da nossa rede (não adaptada para o relax) e aplicando a filosofia just-in-time, adicionamos a seguinte restrição ao modelo do Trabalho 1 (Parte II):

$$t5 <= t7 + 6;$$

Resolvendo o modelo no LPSolve, o instante mais cedo que a segunda atividade (nodo 5) poderá iniciar é nas 13 U.T. (3 U. T. mais cedo em comparação ao modelo sem a restrição just-in-time) e com isto as atividades 1, 2 e 7 são adiadas 2 U. T. cada uma. Comprove-se o que foi dito com o output resultante do LPSolve:

Parte 3	
t0	0
t1	7
t2	13
t3	20
t5	13
t6	0
t7	7
t9	20
t10	5
t11	13
tf	22
	22

3.3. Restrição no Modelo Dual

Se analisarmos a matriz obtida na parte 2 para o modelo primal, podemos ver que na restrição R8, t7 toma o valor de -1 e t5 toma o valor 1 correspondendo ao nodo de destino e origem, respetivamente.

3.4. Input no Relax4

É de lembrar que aqui usaremos de novo a rede adaptada ao Relax4, ou seja, o nodo 7 será o nº 8 e o nodo 5 será o nº 6. Vejamos então o input com a nova restrição just-in-time adicionada:

```
12
21
1	2	0	1000
1	7	0	1000
2	3	-4	1000
2	4	-4	1000
2	6	-4	1000
3	4	-6	1000
4	5	-7	1000
5	12	-2	1000
6	5	-4	1000
6	12	-4	1000
7	8	-5	1000
7	10	-5	1000
8	4	-6	1000
8	6	-6	1000
8	9	-6	1000
9	12	-2	1000
10	6	-8	1000
10	9	-8	1000
10	11	-8	1000
11	9	-7	1000
6	8	6	1000
1
0
0
0
0
0
0
0
0
0
0
-1
```

3.5. Output no Relax4

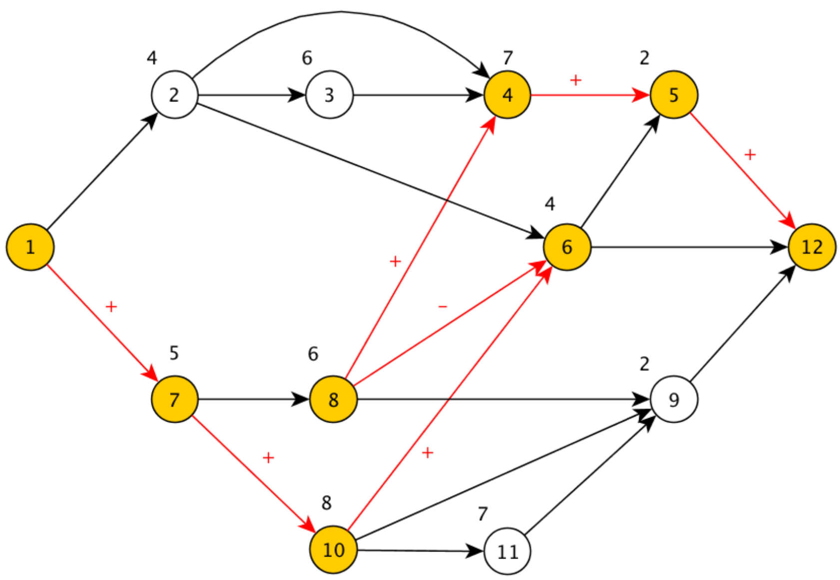
Este foi o output resultante do input inserido no Relax4:

```
END OF READING
 NUMBER OF NODES = 12, NUMBER OF ARCS = 21
 CONSTRUCT LINKED LISTS FOR THE PROBLEM
 CALLING RELAX4 TO SOLVE THE PROBLEM
 ***********************************
 TOTAL SOLUTION TIME =  0. SECS.
 TIME IN INITIALIZATION =  0. SECS.
   1 7  1.
   4 5  1.
   5 12  1.
   7 10  1.
   8 4  1.
   10 6  1.
   6 8  1.
 OPTIMAL COST =  -22.
 NUMBER OF AUCTION/SHORTEST PATH ITERATIONS = 39
 NUMBER OF ITERATIONS =  10
 NUMBER OF MULTINODE ITERATIONS =  1
 NUMBER OF MULTINODE ASCENT STEPS =  2
 NUMBER OF REGULAR AUGMENTATIONS =  1
 ***********************************
```

Como podemos ver, foi possível obter um output do programa, ou seja, o problema é possível tendo uma solução ótima de 22 U.T.

3.6. Interpretação da Solução e Caminho Crítico

Comparando o resultado do nosso modelo inicial com este, podemos observar que foi possível calcular um caminho crítico alternativo, pois o instante mais cedo para realizar a atividade 8 alterou-se de 5 U.T. para 9 U.T. com a nova restrição colocada, o que permitiu ao Relax4 calcular um caminho crítico alternativo ao original (22 U.T.). Assim, o modelo obedece às restrições de conservação de fluxo, como se pode verificar pelo output do relax (entrada e saída de uma unidade). Este foi o caminho crítico resultante na nossa rede:



Parte IV

4.1. Just-In-Time

Escolhido o caminho que passa pelos nodos 0 e 5, podemos dizer que não é possível resolver o problema, pois adicionando as restrições de just-in-time, temos um problema de solução infinita, e convertendo este para o modelo dual, temos um problema impossível de se resolver, segundo o teorema da dualidade fraca, que afirma que cada solução viável do primal é menor ou igual a cada solução viável do dual (caso o primal seja um problema de maximização).

4.2. Formulação do Problema e Resolução no LPSolve

Em relação ao modelo do Trabalho 1 (Parte II), adicionamos as seguintes restrições:

$$t5 \;<=\; t0 \;+\; 4;$$
$$tf \;<=\; t5 \;+\; 4;$$

Inserido este modelo no LPSolve, obtemos um Log de insucesso pois ao adaptar as restrições para a filosofia de just-in-time no caminho escolhido, após 14 iterações, verificou-se que o modelo é impraticável, ou seja, impossível.

4.3. Input no Relax4

O seguinte input foi realizado num editor de texto para execução no Relax4:

```
12
22
1	2	0	1000
1	7	0	1000
2	3	-4	1000
2	4	-4	1000
2	6	-4	1000
3	4	-6	1000
4	5	-7	1000
5	12	-2	1000
6	5	-4	1000
6	12	-4	1000
7	8	-5	1000
7	10	-5	1000
8	4	-6	1000
8	6	-6	1000
8	9	-6	1000
9	12	-2	1000
10	6	-8	1000
10	9	-8	1000
10	11	-8	1000
11	9	-7	1000
12	6	4	1000
6	2	4	1000
1
0
0
0
0
0
0
0
0
0
0
-1
```

4.4. Output do Relax4

Este foi o output resultante do input inserido no Relax4:

```
END OF READING
 NUMBER OF NODES = 12, NUMBER OF ARCS = 22
 CONSTRUCT LINKED LISTS FOR THE PROBLEM
 CALLING RELAX4 TO SOLVE THE PROBLEM
 ***********************************
 TOTAL SOLUTION TIME =  0. SECS.
 TIME IN INITIALIZATION =  0. SECS.
   1 7  1.
   2 3  1000.
   3 4  1000.
   4 5  1000.
   5 12  1000.
   7 10  1.
   9 12  1.
   10 11  1.
   11 9  1.
   12 6  1000.
   6 2  1000.
 OPTIMAL COST =  -11022.
 NUMBER OF AUCTION/SHORTEST PATH ITERATIONS = 40
 NUMBER OF ITERATIONS =  6
 NUMBER OF MULTINODE ITERATIONS =  0
 NUMBER OF MULTINODE ASCENT STEPS =  0
 NUMBER OF REGULAR AUGMENTATIONS =  0
 ***********************************
```

4.5. Output do Relax4 (justificação)

No output do relax, temos um resultado muito maior do que era suposto e quanto maior for a capacidade colocada nos arcos, maior é o resultado, ou seja, isto justifica que o nosso problema tem uma solução infinita.