

Modelos Determinísticos de Investigação  
Operacional - Trabalho 2  
*MIEI - Universidade do Minho*

Sofia Santos - A89615      Ema Dias - A89518  
Sara Queirós - A89491      Tânia Teixeira - A89613

Ano Letivo 2020/21

# 1 Formulação do problema

Este problema baseia-se na exploração e escavação de uma mina a céu aberto. Através de uma estimativa de minérios localizados em blocos de terreno, devemos ser capazes de determinar quais os blocos a escavar de forma a maximizar o lucro, recorrendo a software de otimização de redes.

De acordo com o maior nº de estudante do nosso grupo (89615), a mina na qual o nosso projeto se baseia apresenta o seguinte quadro de inventário de minério estimado:

nível -1						10	8				
nível -2	—				12	14	15	40			—
nível -3	—	—			16			20		—	—
nível -4	—	—	—	3	18	9		6	—	—	—
nível -5	—	—	—	—	20	1		5	—	—	—

valor total: 197

bloco a ignorar    
  bloco possível de minerar e valor do minério, se aplicável

Figure 1: Quadro representativo da nossa mina.

Para escavar um bloco precisamos de escavar o bloco imediatamente acima e os blocos adjacentes ao bloco de cima. Para ilustrar este cenário, numa mina como a da figura abaixo, para escavar o bloco *F* é necessário escavar também os blocos *B*, *C* e *D*.

A	B	C	D
	E	F	

Figure 2: Exemplo de uma mina e da escavação de um bloco.

Para além disso, a mina está dividida em níveis, como tal o custo de escavar um bloco aumenta à medida que se desce de nível, como consta na seguinte tabela:

nível	-1	-2	-3	-4	-5
custo	1	2	3	4	5

Figure 3: Custos de escavação por nível.

Podemos unir esta tabela ao quadro da figura 1, obtendo o seguinte quadro de lucros. O lucro é calculado subtraindo o custo de extrair um bloco ao valor do seu minério.

lucro = valor minério - custo

nível -1	-1	-1	-1	-1	-1	-1	9	7	-1	-1	-1	-1
nível -2	—	-2	-2	-2	-2	10	12	13	38	-2	-2	—
nível -3	—	—	-3	-3	13	-3	-3	-3	17	-3	—	—
nível -4	—	—	—	-1	14	5	-4	-4	2	—	—	—
nível -5	—	—	—	—	15	-4	-5	0	—	—	—	—

bloco a ignorar
  bloco possível de minerar e lucro de minerar o bloco

Figure 4: Lucro obtido por bloco na nossa mina.

Podemos ver que este problema é um problema de fecho máximo de um grafo, o que significa que podemos representar o nosso problema através de um grafo orientado  $G = (V, A)$ . Neste grafo, cada vértice  $j \in V$  tem um peso,  $c_j$ , e o nosso objetivo é determinar o subconjunto fechado de vértices  $S \subseteq V$  de modo a maximizar  $\sum_{j \in S} c_j$ .

Uma forma possível de resolver este problema seria através de um modelo de programação linear, da mesma forma que o primeiro trabalho prático. Porém, não é isso o pretendido. Deste modo, ao invés de um modelo de programação linear, usaremos uma rede de fluxo máximo, de forma a poder resolver este problema através de software de otimização em rede, neste caso o *Relax4*.

Para facilitar na criação da nossa rede, atribuímos a cada bloco da mina um número. Assim, ao criar o grafo, o vértice com um dado número corresponderá ao bloco com o mesmo número. Os números atribuídos aos blocos da nossa mina são os seguintes:

número do vértice

29	30	31	32	33	34	35	36	37	38	39	40
—	19	20	21	22	23	24	25	26	27	28	—
—	—	11	12	13	14	15	16	17	18	—	—
—	—	—	5	6	7	8	9	10	—	—	—
—	—	—	—	1	2	3	4	—	—	—	—

Figure 5: Índice de cada bloco.

## 2 Modelo

Começamos por desenhar um grafo contendo todos os blocos, representados por vértices. Depois unimos cada vértice aos vértices dos blocos dos quais este depende, ou seja, os blocos que precisamos de extrair para extrair o dado bloco, através de arcos com capacidade infinita. A capacidade é infinita porque não temos nenhuma restrição relativa à quantidade de minério que pode ser extraída. Se, por exemplo, um bloco contivesse um minério de valor infinito seria possível obter lucro infinito com a extração desse bloco. A seguinte figura ilustra o nosso modelo inicial:

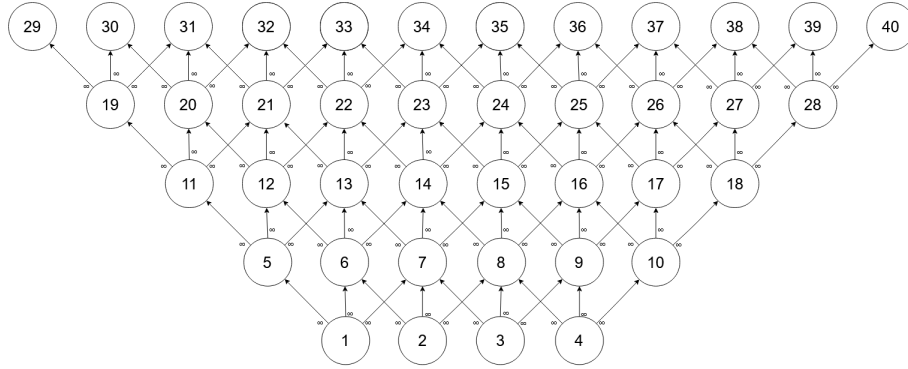


Figure 6: Modelo inicial do nosso projeto.

A partir deste modelo inicial temos de acrescentar um vértice fonte  $s$  (*source*) e uni-lo a todos os nós correspondentes a blocos com lucro positivo através de nós com capacidade igual ao lucro, sempre com origem em  $s$ . Para além de  $s$ , acrescentamos também um vértice terminal  $t$  (*sink*), que terá arcos entre si e os nós correspondentes a blocos com lucro negativo, com capacidade igual ao valor absoluto do lucro e destino em  $t$ .

Usando linguagem matemática, podemos dizer que o nosso grafo de fluxo máximo  $G'$  terá arcos  $(s, v)$ ,  $\forall v \in \{j \in V : c_j > 0\}$  com custo  $c_v$  e arcos  $(v, t)$ ,  $\forall v \in \{j \in V : c_j < 0\}$  com custo  $-c_v$ . Neste último caso, como os pesos são negativos, os custos serão positivos.

Por último, acrescentamos um arco de retorno com origem em  $t$ , destino em  $s$  e valor -1. Este arco extra é necessário pois o *Relax4* assume todos os problemas como problemas de minimização, e neste caso estamos a tentar maximizar o lucro. Deste modo, ao minimizar a função inversa à de maximização, estaremos a maximizar.

Podemos ver na figura 7 o nosso modelo final. É de notar que não ligámos o vértice 4 à fonte nem ao terminal pois o lucro deste é nulo e não há nenhum bloco cuja extração dependa da extração deste, logo, independentemente da nossa solução, não irá valer a pena extrair este bloco. Não haverá perda de lucro, mas haverá perda de tempo e consumo desnecessário de recursos. Até o poderíamos ter removido do modelo que não faria diferença.

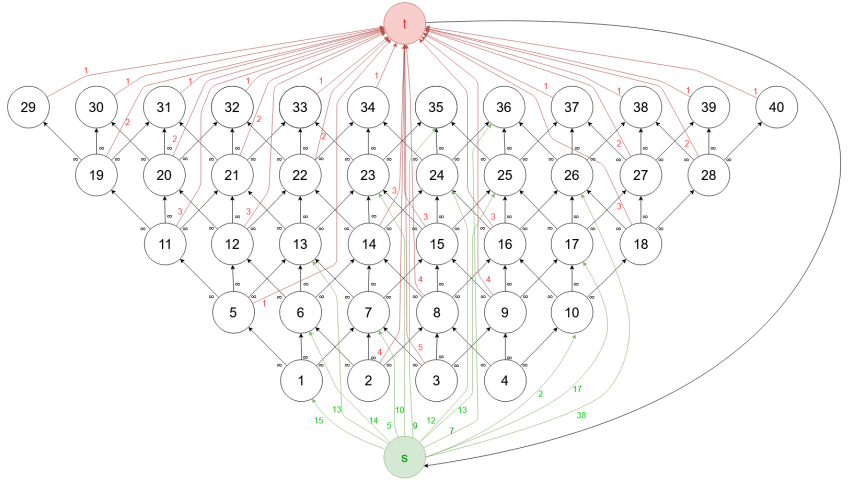


Figure 7: Modelo final do nosso projeto, representa o grafo  $G'$ .

Convertendo este modelo para um formato compatível com o *Relax4*, obtemos o seguinte ficheiro:

# RELAX4 input: X	# RELAX4 input: X	# RELAX4 input: X	# RELAX4 input: X
1 42	49 16 23 0 1000	97 41 26 0 38	144 0
2 125	50 16 26 0 1000	98 41 36 0 9	144 0
3 1 5 0 1000	51 17 23 0 1000	99 41 36 0 7	144 0
4 1 0 0 1000	52 17 26 0 1000	100 2 0 0 4	144 0
5 1 7 0 1000	53 17 27 0 1000	101 3 0 0 5	144 0
6 2 0 0 1000	54 18 26 0 1000	102 5 0 0 6	144 0
7 2 7 0 1000	55 18 27 0 1000	103 8 0 0 4	144 0
8 2 9 0 1000	56 18 28 0 1000	104 9 0 0 4	144 0
9 3 0 0 1000	57 19 29 0 1000	105 11 0 0 3	144 0
10 3 0 0 1000	58 19 30 0 1000	106 12 0 0 2	144 0
11 3 0 0 1000	59 19 31 0 1000	107 14 0 0 3	144 0
12 4 0 0 1000	60 20 26 0 1000	108 15 0 0 3	144 0
13 4 0 0 1000	61 20 27 0 1000	109 15 0 0 3	144 0
14 5 0 0 1000	62 20 28 0 1000	110 16 0 0 2	144 0
15 5 11 0 1000	63 21 31 0 1000	111 16 0 0 2	144 0
16 5 11 0 1000	64 21 32 0 1000	112 20 0 0 2	144 0
17 5 11 0 1000	65 21 33 0 1000	113 21 0 0 2	144 0
18 6 11 0 1000	66 22 33 0 1000	114 22 0 0 2	144 0
19 6 11 0 1000	67 22 34 0 1000	115 22 0 0 2	144 0
20 7 11 0 1000	68 22 35 0 1000	116 28 0 0 2	144 0
21 7 11 0 1000	69 23 35 0 1000	117 28 0 0 1	144 0
22 7 11 0 1000	70 23 36 0 1000	118 30 0 0 1	144 0
23 8 11 0 1000	71 24 35 0 1000	119 31 0 0 1	144 0
24 8 11 0 1000	72 24 36 0 1000	120 32 0 0 1	144 0
25 9 11 0 1000	73 25 35 0 1000	121 35 0 0 1	144 0
26 9 11 0 1000	74 25 36 0 1000	122 36 0 0 1	144 0
27 9 11 0 1000	75 25 37 0 1000	123 39 0 0 1	144 0
28 10 11 0 1000	76 26 36 0 1000	124 40 0 0 1	144 0
29 10 11 0 1000	77 26 37 0 1000	125 42 0 0 1	144 0
30 11 11 0 1000	78 27 37 0 1000	126 42 0 0 1	144 0
31 11 11 0 1000	79 27 38 0 1000	127 42 0 0 1	144 0
32 11 11 0 1000	80 27 39 0 1000	128 0	144 0
33 12 11 0 1000	81 28 38 0 1000	129 0	144 0
34 12 11 0 1000	82 28 39 0 1000	130 0	144 0
35 12 22 0 1000	83 28 40 0 1000	131 0	144 0
36 13 21 0 1000	84 28 40 0 1000	132 0	144 0
37 13 22 0 1000	85 41 3 0 15	133 0	144 0
38 13 22 0 1000	86 41 4 0 4	134 0	144 0
39 13 22 0 1000	87 41 6 0 14	135 0	144 0
40 14 22 0 1000	88 41 7 0 3	136 0	144 0
41 14 22 0 1000	89 41 10 0 2	137 0	144 0
42 14 24 0 1000	90 41 13 0 11	138 0	144 0
43 14 24 0 1000	91 41 17 0 17	139 0	144 0
44 15 24 0 1000	92 41 23 0 10	140 0	144 0
45 15 24 0 1000	93 41 24 0 12	141 0	144 0
46 16 24 0 1000	94 41 25 0 13	142 0	144 0
47 16 24 0 1000	95 41 26 0 30	143 0	144 0

Figure 8: Ficheiro usado como *input* no *Relax4*.

Os valores *1000* no ficheiro representam valores infinitos. Como o *Relax4* não suporta valores infinitos usámos valores muito grandes, neste caso 1000, para representar valores infinitos.

### 3 Solução ótima

Depois de introduzir o ficheiro correspondente ao nosso modelo no *Relax4* (utilizamos uma versão *web* disponível em: <https://neos-server.org/neos/solvers/ln:RELAX4/RELAX4.html>), chegamos a uma aparente solução ótima. Apresentamos aqui a informação relevante dada no output:

```
*****
NUMBER OF NODES = 42, NUMBER OF ARCS = 125
DEFAULT INITIALIZATION USED
*****
Total algorithm solution time = 0.00337100029 sec.
```

OPTIMAL COST = -34.  
 NUMBER OF ITERATIONS = 92  
 NUMBER OF MULTINODE ITERATIONS = 21  
 NUMBER OF MULTINODE ASCENT STEPS = 0  
 NUMBER OF REGULAR AUGMENTATIONS = 21

\*\*\*\*\*

f 1 5 7	f 9 16 0	f 17 27 3	f 26 36 0	f 8 42 0
f 1 6 0	f 9 17 0	f 18 26 0	f 26 37 1	f 9 42 0
f 1 7 1	f 10 16 2	f 18 27 0	f 26 38 1	f 11 42 3
f 2 6 0	f 10 17 0	f 18 28 0	f 27 37 0	f 12 42 3
f 2 7 0	f 10 18 0	f 19 29 1	f 27 38 0	f 14 42 3
f 2 8 0	f 11 19 3	f 19 30 0	f 27 39 1	f 15 42 3
f 3 7 0	f 11 20 0	f 19 31 0	f 28 38 0	f 16 42 2
f 3 8 0	f 11 21 0	f 20 30 1	f 28 39 0	f 18 42 0
f 3 9 0	f 12 20 3	f 20 31 0	f 28 40 0	f 19 42 2
f 4 8 0	f 12 21 0	f 20 32 0	f 41 1 8	f 20 42 2
f 4 9 0	f 12 22 0	f 21 31 1	f 41 4 0	f 21 42 2
f 4 10 0	f 13 21 3	f 21 32 0	f 41 6 6	f 22 42 2
f 5 11 6	f 13 22 3	f 21 33 0	f 41 7 5	f 27 42 2
f 5 12 0	f 13 23 0	f 22 32 1	f 41 10 2	f 28 42 0
f 5 13 0	f 14 22 0	f 22 33 0	f 41 13 6	f 29 42 1
f 6 12 6	f 14 23 0	f 22 34 0	f 41 17 3	f 30 42 1
f 6 13 0	f 14 24 0	f 23 33 1	f 41 23 1	f 31 42 1
f 6 14 0	f 15 23 0	f 23 34 0	f 41 24 1	f 32 42 1
f 7 13 0	f 15 24 0	f 23 35 0	f 41 25 0	f 33 42 1
f 7 14 3	f 15 25 0	f 24 34 1	f 41 26 2	f 34 42 1
f 7 15 3	f 16 24 0	f 24 35 0	f 41 35 0	f 37 42 1
f 8 14 0	f 16 25 0	f 24 36 0	f 41 36 0	f 38 42 1
f 8 15 0	f 16 26 0	f 25 35 0	f 2 42 0	f 39 42 1
f 8 16 0	f 17 25 0	f 25 36 0	f 3 42 0	f 40 42 0
f 9 15 0	f 17 26 0	f 25 37 0	f 5 42 1	f 42 41 34

A partir deste output somos capazes de introduzir os valores dos fluxos no nosso modelo.

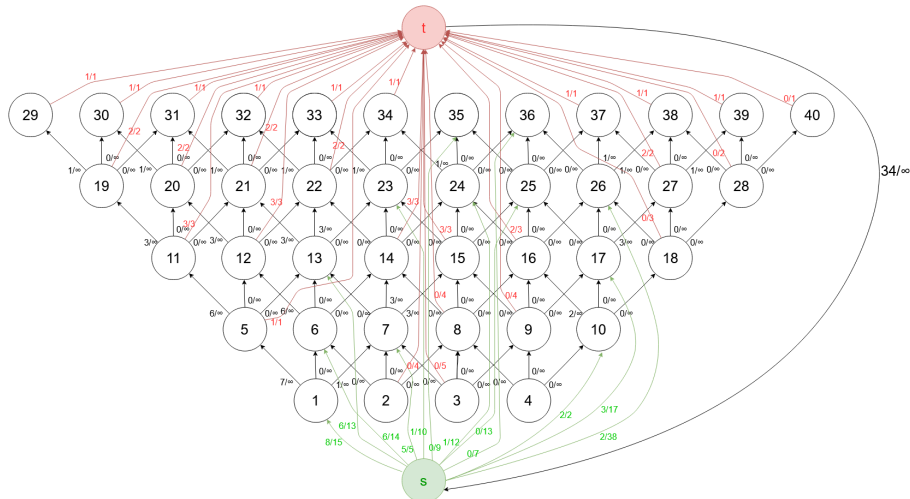


Figure 9: Modelo com os fluxos da solução.

Através da análise do modelo, podemos ver quais os blocos que devem ser escavados ou não. Se um arco com destino no vértice  $t$  tiver fluxo inferior à capacidade do arco, isto é, ao custo de extração, o bloco correspondente ao vértice de origem não irá ser escavado, pois o lucro obtido na extração dos blocos inferiores a estes não é suficiente para cobrir o custo da sua extração. Logo, podemos já excluir os blocos 2, 3, 8, 9, 16, 18, 28 e 40. O bloco 4, como tem lucro nulo e não tem nenhum bloco dependente dele, tal como referimos anteriormente, também pode ser excluído. Também podemos excluir o bloco 10, já que este bloco, apesar de ter lucro positivo, é dependente dos blocos 16 e 18, que não serão extraídos, daí podemos concluir que o custo de alcançar este bloco não compensa a sua extração.

Na figura abaixo podemos ver o corte aplicado ao nosso grafo  $G'$ , representado por linhas vermelhas. Se somarmos o fluxo dos arcos "cortados" pelo corte obtemos o valor do corte dado pelo *Relax4*, ou seja, 34. A secção do corte que inclui  $s$  corresponde à solução do problema.

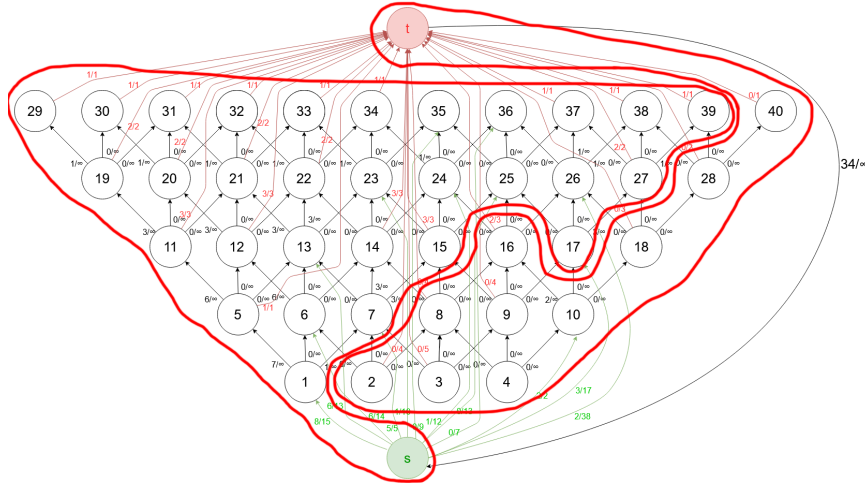


Figure 10: Corte mínimo do nosso grafo.

Poderíamos ter incluído o vértice 7 na secção que inclui  $t$ , visto que o arco que une  $s$  a 7 está saturado. Porém, analisando o grafo, podemos ver que os vértices 13, 14 e 15, dos quais o vértice 7 depende, estão todos no lado do corte que inclui  $s$ , ou seja, serão escavados. Por isso, e como o vértice 7 tem lucro positivo, será proveitoso extrair também este bloco. Vemos assim que é possível realizar múltiplos cortes ao nosso grafo, mas apenas um será o corte mínimo (ou mais se houver um empate).

Temos assim o conjunto dos blocos que devemos extrair, através deste corte mínimo de valor 34. Podemos agora selecionar no nosso quadro da mina quais os blocos a extrair.

número do vértice

29	30	31	32	33	34	35	36	37	38	39	40
—	19	20	21	22	23	24	25	26	27	28	—
—	—	11	12	13	14	15	16	17	18	—	—
—	—	—	5	6	7	8	9	10	—	—	—
—	—	—	—	1	2	3	4	—	—	—	—

bloco a extrair

Figure 11: Quadro com os blocos a extrair assinalados a verde.

Podemos também calcular o lucro total desta extração, que será a soma dos lucros dos blocos extraídos, ou seja, 121 U.M., que pode ser definido pela diferença entre 185 U.M. do valor dos minérios e 64 U.M. do custo da extração.

## 4 Validação do modelo

Para validar o nosso modelo, tentámos resolver o nosso problema através de um modelo de programação linear e verificámos que a solução obtida por este modelo foi a mesma que a solução obtida pela rede de fluxos.

```

Source Matrix Options Result
1 max: 15 x01 - 4 x02 - 5 x03 + 0 x04 - 1 x05 + 14 x06 + 5 x07 - 4 x08 - 4 x09 + 2 x10 - 3 x11
2 - 3 x12 + 13 x13 - 3 x14 - 3 x15 - 3 x16 + 17 x17 - 3 x18 - 2 x19 - 2 x20 - 2 x21 - 2 x22 + 10 x23
3 + 12 x24 + 13 x25 + 38 x26 - 2 x27 - 2 x28 - 1 x29 - 1 x30 - 1 x31 - 1 x32 - 1 x33 - 1 x34 + 9 x35
4 + 7 x36 - 1 x37 - 1 x38 - 1 x39 - 1 x40;
5
6 /* Variable bounds */
7
8 x01 - x05 <= 0;
9 x01 - x06 <= 0;
10 x01 - x07 <= 0;
11 x02 - x06 <= 0;
12 x02 - x07 <= 0;
13 x02 - x08 <= 0;
14 x03 - x07 <= 0;
15 x03 - x08 <= 0;
16 x03 - x09 <= 0;
17 x04 - x08 <= 0;
18 x04 - x09 <= 0;
19 x04 - x10 <= 0;
20 x05 - x11 <= 0;
21 x05 - x12 <= 0;
22 x05 - x13 <= 0;
23 x06 - x12 <= 0;
24 x06 - x13 <= 0;
25 x06 - x14 <= 0;
26 x07 - x13 <= 0;
27 x07 - x14 <= 0;
28 x07 - x15 <= 0;
29 x08 - x14 <= 0;
30 x08 - x15 <= 0;
31 x08 - x16 <= 0;
32 x09 - x15 <= 0;
33 x09 - x16 <= 0;
34 x09 - x17 <= 0;
35 x10 - x16 <= 0;
36 x10 - x17 <= 0;
37 x10 - x18 <= 0;
38 x11 - x19 <= 0;
39 x11 - x20 <= 0;
40 x11 - x21 <= 0;
41 x12 - x20 <= 0;
42 x12 - x21 <= 0;
43 x12 - x22 <= 0;
44 x13 - x21 <= 0;
45 x13 - x22 <= 0;
46 x13 - x23 <= 0;
47 x14 - x22 <= 0;
48 x14 - x23 <= 0;
49 x14 - x24 <= 0;
50 x15 - x23 <= 0;
51 x15 - x24 <= 0;
52 x15 - x25 <= 0;
53 x16 - x24 <= 0;
54 x16 - x25 <= 0;
55 x16 - x26 <= 0;
56 x17 - x25 <= 0;
57 x17 - x26 <= 0;
58 x17 - x27 <= 0;
59 x18 - x26 <= 0;
60 x18 - x27 <= 0;
61 x18 - x28 <= 0;
62 x19 - x29 <= 0;
63 x19 - x30 <= 0;
64 x19 - x31 <= 0;
65 x20 - x30 <= 0;
66 x20 - x31 <= 0;
67 x20 - x32 <= 0;
68 x21 - x31 <= 0;
69 x21 - x32 <= 0;
70 x21 - x33 <= 0;
71 x22 - x32 <= 0;
72 x22 - x33 <= 0;
73 x22 - x34 <= 0;
74 x23 - x33 <= 0;
75 x23 - x34 <= 0;
76 x23 - x35 <= 0;
77 x24 - x34 <= 0;
78 x24 - x35 <= 0;
79 x24 - x36 <= 0;
80 x25 - x35 <= 0;
81 x25 - x36 <= 0;
82 x25 - x37 <= 0;
83 x26 - x36 <= 0;
84 x26 - x38 <= 0;
85 x27 - x37 <= 0;
86 x27 - x38 <= 0;
87 x27 - x39 <= 0;
88 x28 - x38 <= 0;
89 x28 - x39 <= 0;
90 x28 - x40 <= 0;
91
92 bin x01, x02, x03, x04, x05, x06, x07, x08, x09, x10, x11, x12, x13, x14, x15, x16,
93 x17, x18, x19, x20, x21, x22, x23, x24, x25, x26, x27, x28, x29, x30, x31,
94 x32, x33, x34, x35, x36, x37, x38, x39, x40;

```

Figure 12: Modelo de programação linear correspondente ao nosso problema no *LPSolve*.

Aqui, a função objetivo pretende maximizar o lucro com base nos blocos escavados, as restrições dizem respeito a cada arco da nossa rede, e as variáveis são binárias, dizendo-nos se devemos ou não escavar um dado bloco.

Em linguagem matemática, pode ser representado do seguinte modo:

$$\begin{aligned}
 \max: & \sum_{j \in V} c_j x_j \\
 \text{suj. a:} & x_i - x_j \leq 0, \forall (i, j) \in A \\
 & x_j \in \{0, 1\}, \forall j \in V
 \end{aligned}$$



Objective	Constraints	Sensitivity
Variables	MILP Feasible ▲	result
	121	121
x02	0	0
x03	0	0
x04	0	0
x08	0	0
x09	0	0
x10	0	0
x16	0	0
x18	0	0
x28	0	0
x40	0	0
x01	1	1
x05	1	1
x06	1	1
x07	1	1
x11	1	1
x12	1	1

Figure 13: Resultado do modelo de programação linear corrido no *LPSolve*. É possível ver na figura quais os blocos que não devem ser escavados, sendo que os que não aparecem na figura devem todos ser escavados.

Ao correr este modelo, o seu resultado diz-nos que devemos extrair os blocos 1, 5, 6, 7, 11 a 15, 17, 19 a 27 e 29 a 39 de modo a maximizar o lucro, tal como a solução do nosso modelo de rede de fluxo máximo. Daí, podemos argumentar a favor da validade do nosso grafo de fluxos.