Universidade do Minho

Licenciatura em Engenharia Informática

Investigação Operacional

Trabalho Prático 2

Ana Murta (A93284)
Ana Henriques (A93268)
Leonardo Freitas (A93281)
Luís Faria (A93209)
Tiago Carneiro (A93027)

8 de maio de 2022

Conteúdo

Lista de Figuras

Capítulo 1

Introdução

O segundo trabalho prático da Unidade Curricular de Investigação Operacional tem por base o problema de escalonamento de equipas, em que se pretende atribuir a equipas serviços efetuados por clientes distribuídos geograficamente. O principal objetivo é, então, minimizar o custo total, desde custos de deslocação a custos fixos de utilização de veículos. Para a resolução deste problema, recorreu-se ao *software* de otimização de redes `Relax4` e utilizou-se, ainda, o ficheiro Excel disponibilizado pelos docentes, que contém todos os dados necessários.

Capítulo 2

Formulação do problema

Tal como foi mencionado na introdução – Capítulo 1, o objetivo deste trabalho prático passa por minimizar o custo total de operação, desde custos de deslocação a custos fixos de utilização de veículos, de modo a que as equipas sirvam todos os clientes uma vez, no máximo.

A pedido no enunciado, foi considerado ABCDE, o maior número de inscrição do grupo, para a remoção de clientes e de tempos de deslocação do quadro proposto inicialmente.

Como tal, sendo 93284 o maior número de estudante do nosso grupo, os valores de a_1 e a_8 são, respetivamente, 4 (=3+1) e 3 (=2+1). Uma vez que tomam valores pares, foram, também, removidos os clientes D e E que, segundo o maior número de inscrição do grupo, correspondem aos dígitos 8 e 4, respetivamente. O quadro resultante é o ilustrado na figura 2.1.

j	cliente	a_j (1/4 hora)	a_j (hora do serviço)
1	Ana	$a_1 = 4$	10:00
2	Beatriz	7	10:45
3	Carlos	4	10:00
4	-	-	-
5	Eduardo	10	11:30
6	Francisca	6	10:30
7	Gonçalo	9	11:15
8	-	-	-
9	Inês	2	09:30
10	José	5	10:15

Figura 2.1: Nosso quadro das horas de serviço dos clientes

Tendo sido removidos os clientes cujos números são 8 e 4, os quadros iniciais onde estavam indicados os tempos e os custos de deslocação também sofreram alterações, estando os resultados ilustrados nas figuras 2.2 e 2.3, respetivamente.

	B	C	E	F	G	I	J	K
A	4	1	2	3	2	0	3	1
B		3	3	3	2	4	2	5
C			2	3	2	1	1	2
E				2	1	2	2	2
F					2	3	3	4
G						2	2	3
I							3	2
J								4

Figura 2.2: Tempos de deslocação – $t_i j$

	B	C	E	F	G	I	J	K
A	13	5	5	10	7	0	7	1
B		11	10	8	6	13	4	15
C			6	10	6	5	6	2
E				6	4	5	7	6
F					5	10	8	11
G						7	5	9
I							7	9
J								10

Figura 2.3: Custos de deslocação – $c_i j$

Capítulo 3

Modelo

Após a formulação do problema, começámos por desenhar o grafo de compatibilidades, onde cada vértice representa um cliente. Em seguida, aplicámos a expressão $a_i + t_{ij} \leq t_j$ para conectar os vértices. Como não existe um limite relativamente ao número de equipas que podem ser enviadas, recorreu-se a arcos com capacidade infinita na ligação entre os vértices. Na Figura 3.1, podemos observar, então, o grafo obtido numa fase inicial.

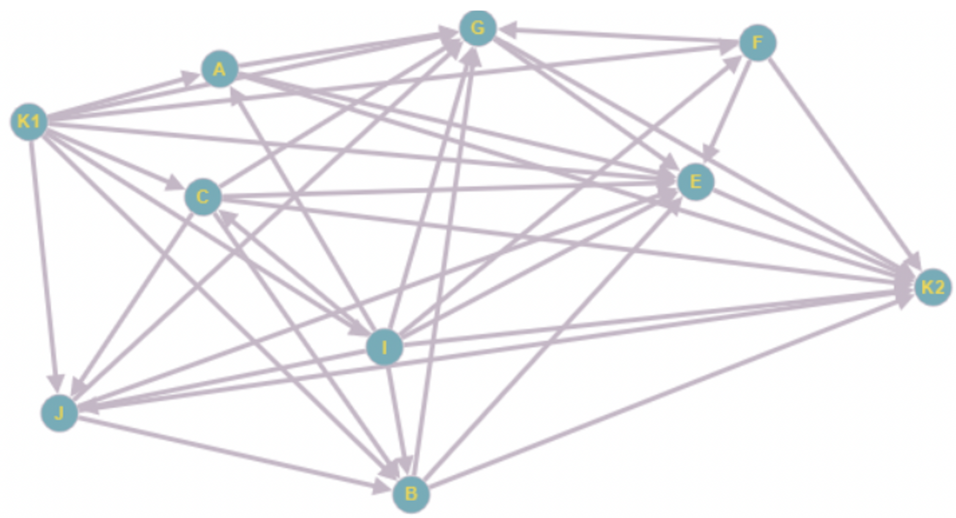

Figura 3.1: Grafo de compatibilidades

Seguidamente, foi necessário ligar unidirecionalmente aos outros vértices dois vértices que não se encontram simbolizados no modelo inicial: um vértice que representa a sede da empresa, em Keleirós, identificado pela letra K, e um vértice destino que também corresponde à sede. Isto porque, é referido no enunciado que, assim que uma equipa termine de servir o/os cliente/s que lhe foi atribuído/s, a mesma deve retomar à sede da empresa. Adicionalmente, importa referir que se acrescentou o custo de uma 1 unidade às ligações entre os clientes e a sede da empresa. Posto isto, e considerando todos os vértices do modelo, é pedido para minimizar o número de equipas utilizadas para percorrer os clientes, de modo a diminuir o custo total da operação, i.e. o custo da deslocação e os custos fixos de utilização de veículos.

A Figura 3.2 representa o ficheiro de *input* criado e submetido ao *software* de otimização em rede utilizado, conhecido por *Relax4*. Este ficheiro foi, então, conseguido através da conversão do grafo de compatibilidades, ilustrado na Figura 3.1. Visto que estamos perante um problema que trabalha arcos com capacidade infinita na ligação entre os vértices, foi imperativo a utilização de um grande valor – neste caso, o valor 1000 – para representar os valores infinitos e, assim, permitir que o *Relax4* consiga suportar todos os valores que constituem o nosso grafo. Para além disso, a solução ótima tem que conter todos os nodos uma só vez, ou seja, nos arcos da solução, um nodo cliente só pode ser uma vez a origem e uma vez o destino, logo, para forçar o *Relax4* a dar a solução que pretendemos, desdobramos todos os nodos em dois, o nodo de entrada e o nodo de saída, e demos aos vértices correspondentes um de procura e um de oferta, respetivamente. Ao nodo correspondente à sede demos oito de procura ao nodo de entrada (nodo sede a que as equipas retornam) e oito de oferta ao nodo de saída (nodo sede de onde as equipas partem), pois oito equipas seriam o máximo de equipas necessárias, visto que existem oito nodos cliente, e todos eles estão conectados à sede. Para não forçar todas as oito equipas a passar por arcos com custo desnecessariamente, adicionamos um arco de custo zero que conecta o nodo da sede de saída ao de entrada. Os custos dos arcos são iguais ao custo da tabela dos custos de deslocação(figura 2.3), exceto os arcos que terminam no nodo sede de entrada. Esses arcos custam mais uma unidade do que está apresentado na tabela e essa unidade representa o custo fixo da mobilização da equipa que vai usar aquele arco no seu percurso.

```
22
38
1 15 5 1000
1 17 7 1000
1 22 2 1000
2 15 10 1000
2 17 6 1000
2 22 16 1000
3 12 11 1000
3 15 6 1000
3 17 6 1000
3 20 6 1000
3 22 3 1000
5 22 7 1000
6 15 6 1000
6 17 5 1000
6 22 12 1000
7 15 4 1000
7 22 10 1000
9 22 10 1000
9 11 0 1000
9 12 13 1000
9 13 5 1000
9 15 5 1000
```

```
9 15 5 1000
9 16 10 1000
9 17 7 1000
9 20 7 1000
10 22 11 1000
10 12 4 1000
10 15 7 1000
10 17 5 1000
21 11 1 1000
21 12 15 1000
21 13 2 1000
21 15 6 1000
21 16 11 1000
21 17 9 1000
21 19 9 1000
21 20 10 1000
21 22 0 1000
1
1
1
0
1
1
1
0
```

```
1
0
1
1
-1
-1
-1
0
-1
-1
-1
0
-1
-1
8
```

Figura 3.2: Ficheiro de *input* submetido ao Relax4

Capítulo 4

Solução ótima

Depois de introduzir o ficheiro, apresentado na Figura 3.2, que corresponde ao nosso modelo no *Relax4* (`https://neos-server.org/neos/solvers/lno:RELAX4/RELAX4.html`), chegámos a uma aparente solução ótima. A Figura 4.1 representa a dita solução ótima, conseguida com o ficheiro anteriormente mencionado.

```
   NEOS Server Version 6.0
   Job#     : 11915967
   Password : NUyATFVL
   User     :
   Solver   : lno:RELAX4:RELAX4
   Start    : 2022-04-30 11:50:26
   End      : 2022-04-30 11:50:33
   Host     : prod-sub-1.neos-server.org

   Disclaimer:

   This information is provided without any express or
   implied warranty. In particular, there is no warranty
   of any kind concerning the fitness of this
   information for any particular purpose.

   Announcements:
*************************************************************
 NUMBER OF NODES = 22, NUMBER OF ARCS = 38
 DEFAULT INITIALIZATION USED
 ***********************************
 Total algorithm solution time =   0.00331401825 sec.
 OPTIMAL COST =   63.
 NUMBER OF ITERATIONS =  8
 NUMBER OF MULTINODE ITERATIONS =  1
 NUMBER OF MULTINODE ASCENT STEPS =  0
 NUMBER OF REGULAR AUGMENTATIONS =  1
 ***********************************

-------------------- begin dimacs-format results --------------------
```

Figura 4.1: Ficheiro de *output* do *Relax4*

```
c
s 63.
f 1 15 0
f 1 17 0
f 1 22 1
f 2 15 0
f 2 17 1
f 2 22 0
f 3 12 0
f 3 15 0
f 3 17 0
f 3 20 1
f 3 22 0
f 5 22 1
f 6 15 0
f 6 17 0
f 6 22 1
f 7 15 1
f 7 22 0
f 9 22 0
f 9 11 0
f 9 12 0
f 9 13 0
f 9 15 0
f 9 16 1
f 9 17 0
f 9 20 0
f 10 22 0
f 10 12 1
f 10 15 0
f 10 17 0
f 21 11 1
f 21 12 0
f 21 13 1
f 21 15 0
f 21 16 0
f 21 17 0
f 21 19 1
f 21 20 0
f 21 22 5
```

Figura 4.2: *Script* resultante do *Relax4*

No ficheiro *output* obtido do *Relax4*, temos informação suficiente que possibilita a tradução de um caminho e de um grafo, que resultará, deste modo, na solução ideal que procuramos. A partir desta análise, podemos concluir quais os caminhos que nos permitem minimizar a quantidade de equipas necessária para percorrer os clientes, i.e. conseguimos perceber que clientes podem ser percorridos sequencialmente. Assim sendo, obtivemos a seguinte solução:

K $\rightarrow I \rightarrow F \rightarrow K$

A equipa que faz o percurso acima parte da sede às 09:00, chega a I às 09:30, a hora do serviço. Parte logo de seguida, chega a F às 10:15 e tem que esperar até às 10:30 para acabar o serviço. Não sendo responsável por mais serviços, a equipa volta para a sede, chegando às 11:30. O custo de operação desta equipa é 31.

K $\rightarrow C \rightarrow J \rightarrow B \rightarrow G \rightarrow E \rightarrow K$

No percurso imediatamente acima, uma equipa parte da sede às 09:00, chega a C às 09:30, tendo que esperar 30 minutos para efetuar o serviço. Parte às 10:00 e chega a J às 10:15, a hora certa do serviço. Parte imediatamente para B e chega às horas do serviço, 10:45. Dirige-se logo pra G e chega pontualmente ao serviço às 11:15. Departe de seguida para E e chega às 11:30, hora certa do serviço. Retorna para a sede e chega às 12:00. O custo de operação desta equipa é 29.

K $\rightarrow A \rightarrow K$

No ultimo percurso, a equipa responsável parte às 09:00 para A e chega às 09:15, tendo que esperar 45 minutos para efetuar o serviço. Sendo o único serviço que tem que realizar, às 10:00 volta para a sede, a que chega às 10:15. O custo de operação desta equipa é 3.

O custo total da solução ótima é 63.

Na figura 4.3, podemos observar o conjunto de três caminhos que minimizam os custos da operação, desde custos de deslocação a custos fixos de utilização de veículos. Por outras palavras, temos representada a quantidade de equipas necessárias para servir todos clientes sem que haja desperdício de recursos.

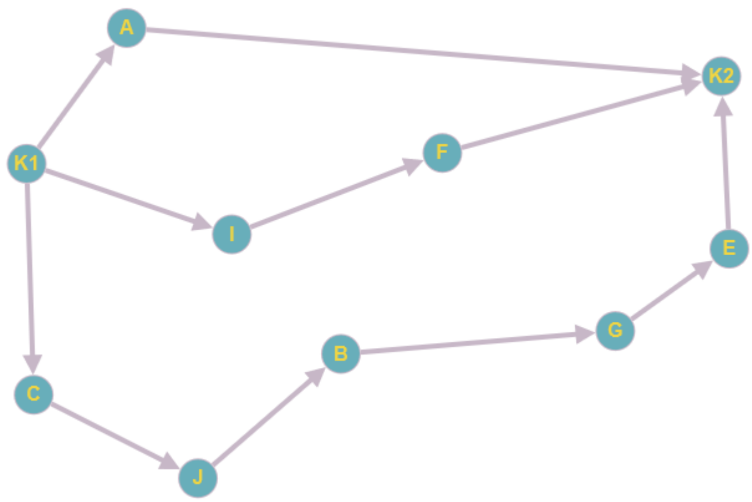

Figura 4.3: Solução ótima do modelo de uma rede de fluxos representada num grafo

Capítulo 5

Validação do modelo

Por fim, e visando como o objetivo a validação do nosso modelo, resolvemos o problema em mãos utilizando um modelo de programação linear, através do programa *LPsolve*. Com este modelo, a solução ótima obtida foi 63, mostrando esta ser a mesma solução conseguida pelo o modelo de uma rede de fluxos. Contudo, uma diferença encontrada entre estes dois tipos de modelos é a definição dos caminhos a percorrer, que revelaram resultados ligeiramente diferentes.

Na Figura 5.1, temos o ficheiro *input* passado ao *LPSolve*, onde estão presentes a função objetivo, as restrições e as variáveis binárias necessárias para definir o modelo de programação linear. A função objetivo foi desenvolvida de modo a minimizar a quantidade de equipas necessárias para servir todos os clientes. Adicionalmente, importa referir que as restrições se referem a cada arco da rede e, também, que as variáveis binárias refletem sobre a questão do cliente dever ser ou não colocado no caminho.

```
min: 1*xKA + 15*xKB + 2*xKC +  6*xKE + 11*xKF + 9*xKG + 9*xKI +  10*xKJ +
                               5*xAE +          7*xAG +                    2*xAR +
                              10*xBE +          6*xBG +                   16*xBR +
             11*xCB +          6*xCE +          6*xCG +          6*xCJ +  3*xCR +
                                                                           7*xER +
                               6*xFE +          5*xFG +                   12*xFR +
                               4*xGE +                                    10*xGR +
      0*xIA + 13*xIB + 5*xIC +         5*xIE + 10*xIF + 7*xIG +          7*xIJ + 10*xIR +
              4*xJB +          7*xJE +          5*xJG +                   11*xJR;

/* Restrições */

xKA + xIA = 1;
xAE + xAG + xAR = 1;
xCB + XIB + xJB + xKB = 1;
xBE + xBG + xBR = 1;
xIC + xKC = 1;
xCB + xCE + xCG + xCJ + xCR = 1;
xAE + xBE + xCE + xFE + xGE + xIE + xJE + xKE = 1;
xER = 1;
xIF + xKF = 1;
xFE + xFG + xFR = 1;
xAG + xBG + xCG + xFG + xIG + xJG + xKG = 1;
xGE + xGR = 1;
xKI = 1;
xIA + xIB + xIC + xIE + xIF + xIG + xIJ + xIR = 1;
xCJ + xIJ + xKJ = 1;
xJB + xJE + xJG + xJR = 1;
xKA + xKB + xKC + xKE + xKF + xKG + xKI + xKJ >= 1;
xAR + xBR + xCR + xER + xFR + xGR + xIR + xJR >= 1;

bin xKA, xKB, xKC, xKE, xKF, xKG, xKJ,
    xAE, xAG, xAR,
    xBE, xBG, xBR,
    xCB, xCE, xCG, xCJ, xCR,
    xFE, xFG, xFR,
    xGE, xGR,
    xIA, xIB, xIC, xIE, xIF, xIG, xIJ, xIR,
```

Figura 5.1: Ficheiro *input* no *LPsolve*

O resultado obtido pelo *LPSolve* está ilustrado na Figura 5.2 e, a partir do mesmo, chegamos à solução de que, no grafo final, serão criadas as arestas KI, KC, KA, JB, IF, GR, FE, ER, CJ, BG, AR, com um comprimento total de 63.

Variables	M...	result
	63	63
xKI	1	1
xKC	1	1
xKA	1	1
xJB	1	1
xIF	1	1
xGR	1	1
xFE	1	1
xER	1	1
xCJ	1	1
xBG	1	1
xAR	1	1
xKJ	0	0
xKG	0	0
xKF	0	0
xKE	0	0
xKB	0	0
xJR	0	0
xJG	0	0
xJE	0	0
xIR	0	0
xIJ	0	0
xIG	0	0
xIE	0	0
xIC	0	0
xIB	0	0
XIB	0	0
xIA	0	0
xGE	0	0
xFR	0	0
xFG	0	0
xCR	0	0
xCG	0	0
xCE	0	0
xCB	0	0
xBR	0	0

Figura 5.2: *Output* produzido pelo *LPSolve*

Comparando os dois modelos previamente trabalhados, averigua-se que, em ambos, o tamanho da solução obtida é 63, o que permite validar o grafo de compabilidade construído e previamente apresentado na Figura 3.1. Todavia, importa denotar que, embora o tamanho de ambas as soluções seja igual (=63), os caminhos obtidos em cada modelo são ligeiramente diferentes, tal como se pode verificar com o grafo que representa o caminho da solução ótima do modelo de programação linear, ilustrado na Figura 5.3.

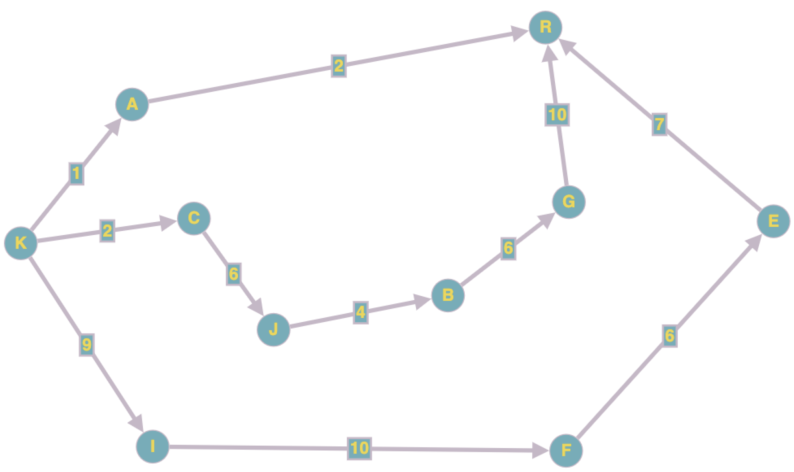

Figura 5.3: Solução ótima do modelo de programação linear representada num grafo

Capítulo 6

Conclusão

No decorrer do desenvolvimento deste projeto, o grupo consolidou os métodos e conceitos abordados na aulas teóricas e práticas da Unidade Curricular de Investigação Operacional, permitindo aperfeiçoar a capacidade de análise e criação de modelos que descrevem sistemas complexos, bem como interpretar as soluções para os mesmos. Assim sendo, para este trabalho prático, o grupo utilizou o *software* de otimização em rede, o *Relax4*, para determinar o número de equipas necessário para servir todos os clientes a fim de minimizar o custo total da operação.

Apesar de ter enfrentado algumas dificuldades ao concretizar este trabalho prático, nomeadamente na interpretação do problema e na construção do ficheiro *input* no *Relax4*, o grupo encontra-se satisfeito com o resultado obtido uma vez que, graças à validação do modelo, estamos seguros de que chegámos à solução ótima.