Investigação Operacional

Trabalho Prático 2

Licenciatura em Engenharia Informática, Universidade do Minho

Dinis Estrada (A97503)
Inês Ferreira (A97372)
Joana Branco (A96584)
Joana Pereira (A97588)
Marta Sá (A97158)

Braga, 7 de maio de 2022

1. Formulação do Problema

Este problema baseia-se na atribuição de serviços a efetuar a clientes distribuídos geograficamente a equipas. O objetivo é minimizar o custo total deste procedimento tendo em conta os gastos de deslocação e o tempo.

De acordo com o enunciado, cada cliente *j*, *j E V*, sendo V conjunto de clientes, tem associada uma hora de início do serviço *aj*. Uma equipa pode efectuar o serviço do cliente *j* se, após terminar o serviço de um cliente *i*, puder chegar ao cliente *j* num instante igual ou inferior a *aj* , i.e., ***ai + tij ≤ aj*** , em que *tij* é o tempo de deslocação entre os clientes *i* e *j,* ∀ *i, j* ∈ *V*. Caso a equipa chegue antes, é necessário esperar pela hora de início do serviço. Para além disso, para a resolução deste problema temos que ter em conta algumas considerações:

- A duração do serviço no cliente é desprezável.
- Cada equipa inicia o trabalho às 09:00, na sede da empresa (*K*).
- Cada equipa pode visitar um qualquer número de clientes, uma vez que a carga não constitui uma limitação.

As horas dos serviços dos clientes, em quartos de hora desde o início do período de trabalho, efetuados pelas diversas equipas encontram-se na tabela abaixo já com as alterações necessárias indicadas no enunciado. Neste caso, o maior número de estudante é 97588 (ABCDE), ou seja,

a1 = B + 1 = 7 + 1 = 8;
a8 = C + 1 = 5 + 1 = 6.

Para além disto, como D e E são pares (D=E=8) iremos remover os clientes D e E.

j	Cliente	aj (¼ hora)	aj (hora do serviço)
1	Ana	8	11:00
2	Beatriz	7	10:45
3	Carlos	4	10:00
4	Diogo	2	09:30
5	Eduardo	10	11:30
6	Francisca	6	10:30
7	Gonçalo	9	11:15
8	Helena	6	10:30
9	Inês	2	09:30
10	José	5	10:15

Tabela 1: Detalhes dos serviços dos clientes.

Os tempos de deslocação *tij* , ∀ *i,j* ∈ *V* ∪ *{K}*, em quartos de hora, e os custos de deslocação *cij* , ∀ *i,j* ∈ *V* ∪ *{K}*, são os indicados nas seguintes figuras:

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

Figura 1: Tempos de deslocação.

	B	C	D	E	F	G	H	I	J	K
A	13	5	6	5	10	7	5	0	7	1
B		11	14	10	8	6	11	13	4	15
C			8	6	10	6	0	5	6	2
D				4	8	8	8	6	11	4
E					6	4	6	5	7	6
F						5	10	10	8	11
G							10	7	5	9
H								5	6	9
I									7	9
J										10

Figura 2: Custos de deslocação.

De seguida foi construída uma tabela com os caminhos possíveis tendo em conta a hora de cada um dos serviços e o tempo de deslocação entre os diversos clientes.

	A	B	C	F	G	H	I	J	K	
A										11:00
B										10:45
C										10:00
F										10:30
G										11:15
H										10:45
I										09:30
J										10:15
K										09:00

Legenda:

→ As células a rosa são as células em que o trajeto é impossível uma vez que s hora de início do primeiro serviço é superior à hora de início do segundo.

→ As células a azul são as arestas válidas, que cumpres com as restrições.

→ As células a amarelo são as células que se exclui devido ao tempo de deslocação entre células afetar que a equipa chegue a tempo ao cliente.

→ As células cinzas são as células em que o trajeto é para o mesmo nodo.

Desenvolvendo o problema, podemos concluir que o mesmo pode ser resolvido encontrando o **fluxo de custo mínimo** numa rede que inclui um vértice (K) para representar a sede da empresa e um **grafo de compatibilidades orientado G=(V,A), A ⊆ V×V** relativo aos custos de deslocação. Neste grafo, os vértices representam todos os clientes e os arcos correspondem às deslocações possíveis das equipas entre os mesmos. Para além disso, os valores, *cij*, representados pelos arcos indicam o custo de deslocação entre clientes ou entre cliente e sede. O exercício em questão tem como objetivo minimizar este mesmo valor e, para resolvê-lo, iremos utilizar um software de otimização em rede, o *Relax4*.

Assim, na figura seguinte pode-se visualizar a representação da nossa rede:

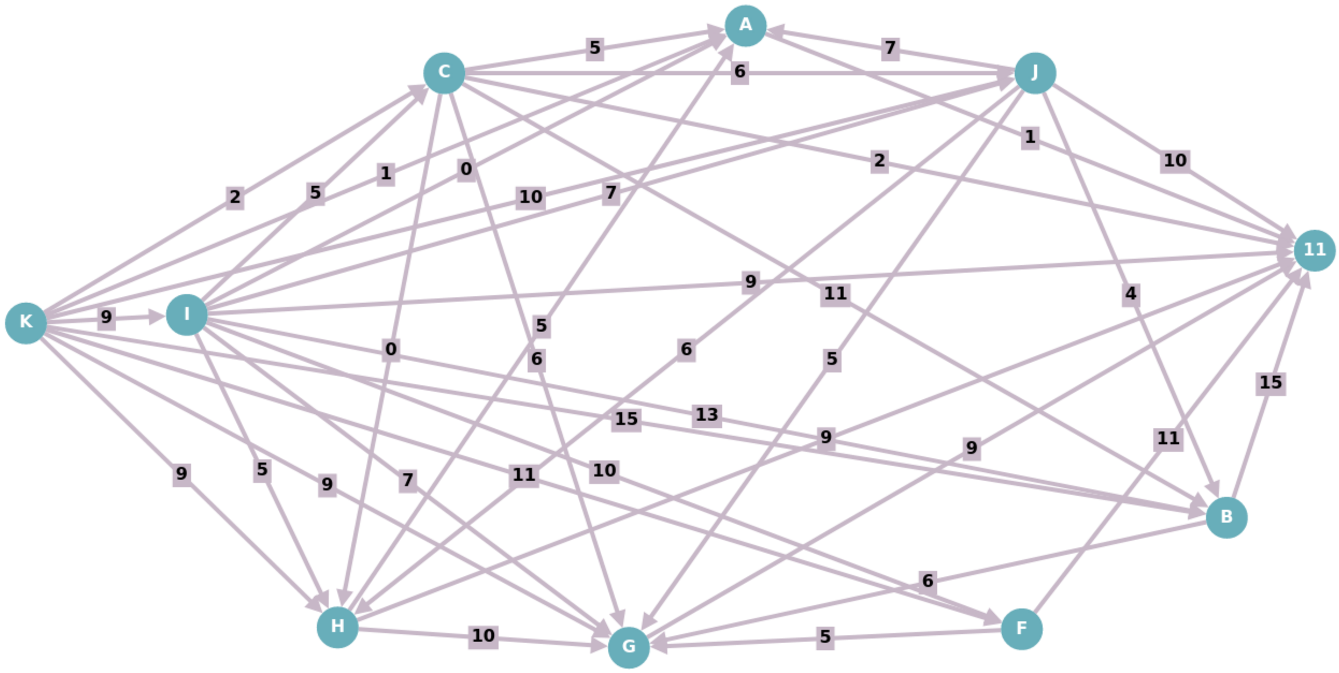


Figura 3: Representação inicial da rede.

De forma a tornar mais fácil a representação do modelo no *Relax4*, criamos as seguintes associações entre 2 números e 1 letra.

Cliente	Recebe	Envia
A	1	11
B	2	12
C	3	13
F	4	14
G	5	15
H	6	16
I	7	17
J	8	18
k	9	10

Tabela 2: Novos valores dos nodos.

A coluna “Recebe” representa a chegada de fluxo até um dado cliente (representado por uma letra) e, por sua vez, a coluna “Envia” representa o envio de fluxo a partir de um cliente. É de notar que as arestas que partem dos nodos quando representados pelos números da coluna da esquerda têm que ir apenas para o correspondente número da coluna da direita com um custo 0 associado.

Num sistema real, é necessário implementar decisões que permitam identificar os melhores caminhos entre nodos. Por conseguinte, para determinar a nossa solução, basta associar os valores dos fluxos aos valores representados pelos custos de deslocação, uma vez que já restringimos o fluxo no tempo. Assim, conseguimos minimizar as despesas no modelo.

2. Modelo

Neste modelo recorreu-se à numeração dos vértices como números naturais. Assim, os vértices de 1 a 3 correspondem aos clientes de ‘A’ a ‘C’, os de 6 a 10 aos clientes de ‘F’ a ‘J’ e o vértice 11 representa a sede da empresa (‘K’). Desta forma, podemos definir V={1,2,3,6,7,8,9,10,11} como o conjunto das numerações de vértices a serem utilizados no modelo. Para além disso, podemos definir A={(1,11), (2,7), (2,11), (3,1), (3,2), (3,7), (3,8), (3,10), (3,11), (6,7), (6,11), (7,11), (8,1), (8,7), (8,11), (9,1), (9,2), (9,3), (9,6), (9,7), (9,8), (9,10), (9,11), (10,1), (10,2), (10,7), (10,8), (10,11), (11,1), (11,2), (11,3), (11,6), (11,7), (11,8), (11,9), (11,10)} como o conjunto de arcos existentes no modelo.

Utilizando linguagem matemática, podemos afirmar que o nosso grafo orientado terá arcos (i, j), $\forall (i, j) \in A$ com custo unitário cij>=0, capacidade uij>=0 e vértices com valores de oferta ou consumo bj , $\forall j \in V$.

Desta forma, o nosso modelo no formato *Relax4*, assume a seguinte estrutura:

```
18
44

11 9 2 1000
1 11 0 1000

12 9 16 1000
12 5 6 1000
2 12 0 1000

13 9 3 1000
13 1 5 1000
13 2 11 1000
13 5 6 1000
13 6 0 1000
13 8 6 1000
3 13 0 1000

14 9 12 1000
14 5 5 1000
4 14 0 1000

15 9 10 1000
5 15 0 1000

16 9 10 1000
16 1 5 1000
16 5 10 1000
6 16 0 1000

17 9 10 1000
17 1 0 1000
17 2 13 1000
17 3 5 1000
17 4 10 1000
17 5 7 1000
17 6 5 1000
17 8 7 1000
7 17 0 1000

18 9 11 1000
18 1 7 1000
18 2 4 1000
18 5 5 1000
18 6 6 1000
8 18 0 1000

10 1 1 1000
10 2 15 1000
10 3 2 1000
10 4 11 1000
10 5 9 1000
10 6 9 1000
10 7 9 1000
10 8 10 1000

-1
-1
-1
-1
-1
-1
-1
-1
-3
3
1
1
1
1
1
1
1
1

```

Figura 4: Dados de *input* no *Relax4.*

Por um lado, as duas primeiras linhas representam, respetivamente, o número de nodos e o número de arestas do modelo. Por outro lado, as linhas com 4 valores simbolizam o fluxo de dados nas arestas onde o primeiro valor corresponde ao nodo de saída, o segundo ao nodo de entrada, o terceiro ao custo de deslocação entre os nodos e o quarto à capacidade, definida com o valor 1000 para que não interfira com o modelo. É importante realçar que somamos uma unidade ao terceiro valor da primeira linha de cada bloco devido ao custo necessário para enviar uma equipa.

Por último, o ficheiro de *input* termina com 18 linhas sendo que as 9 primeiras caracterizam o número de equipas que saem (valores de oferta) e as 9 últimas o número de equipas que entram no respetivo nodo (valores de consumo).

3. Solução Ótima

A nossa primeira tentativa foi enviar apenas duas equipas, mas não funcionou como verificamos na seguinte figura:

```
************************************************************
 NUMBER OF NODES = 18, NUMBER OF ARCS = 44
 DEFAULT INITIALIZATION USED
 **********************************
  PROBLEM IS FOUND TO BE INFEASIBLE.
 PROGRAM ENDED; PRESS <CR> TO EXIT
```

Figura 5: Dados de *output* com duas equipas.

De seguida tentamos obter uma solução com três equipas, obtendo o seguinte resultado:

```
************************************************************
 NUMBER OF NODES = 18, NUMBER OF ARCS = 44
 DEFAULT INITIALIZATION USED
 **********************************
 Total algorithm solution time =   0.00336194038 sec.
 OPTIMAL COST =   68.
 NUMBER OF ITERATIONS =  22
 NUMBER OF MULTINODE ITERATIONS =  3
 NUMBER OF MULTINODE ASCENT STEPS =  4
 NUMBER OF REGULAR AUGMENTATIONS =  3
 **********************************
```

Figura 6: Dados de *output* com três equipas.

Por fim, experimentamos com um número de quatro equipas obtendo assim a solução apresentada na imagem seguinte:

```
************************************************************
 NUMBER OF NODES = 18, NUMBER OF ARCS = 44
 DEFAULT INITIALIZATION USED
 **********************************
 Total algorithm solution time =   0.00317406654 sec.
 OPTIMAL COST =   71.
 NUMBER OF ITERATIONS =  23
 NUMBER OF MULTINODE ITERATIONS =  4
 NUMBER OF MULTINODE ASCENT STEPS =  5
 NUMBER OF REGULAR AUGMENTATIONS =  4
 **********************************
```

Figura 7: Dados de *output* com quatro equipas.

Através da análise das figuras verifica-se que o número de nodos e arcos foi mantido. Estes campos correspondem ao número de vértices e arcos existentes no nosso modelo.

O *Optimal Cost* é o que garante qual o modelo com o menor custo para a resolução do problema em questão e, comparando as três figuras, a solução com três equipas é a melhor opção para poder servir todos os clientes sendo o *Optimal Cost* igual a 68.

```
s 68.
f 11 9 1
f 1 11 0
f 12 9 0
f 12 5 1
f 2 12 0
f 13 9 0
f 13 1 0
f 13 2 0
f 13 5 0
f 13 6 1
f 13 8 0
f 3 13 0
f 14 9 1
f 14 5 0
f 4 14 0
f 15 9 1
f 5 15 0
f 16 9 0
f 16 1 1
f 16 5 0
f 6 16 0
f 17 9 0
f 17 1 0
f 17 2 0
f 17 3 0
f 17 4 0
f 17 5 0
f 17 6 0
f 17 8 1
f 7 17 0
f 18 9 0
f 18 1 0
f 18 2 1
f 18 5 0
f 18 6 0
f 8 18 0
f 10 1 0
f 10 2 0
f 10 3 1
f 10 4 1
f 10 5 0
f 10 6 0
f 10 7 1
f 10 8 0
```

Figura 8: Solução ótima dada pelo *Relax4.*

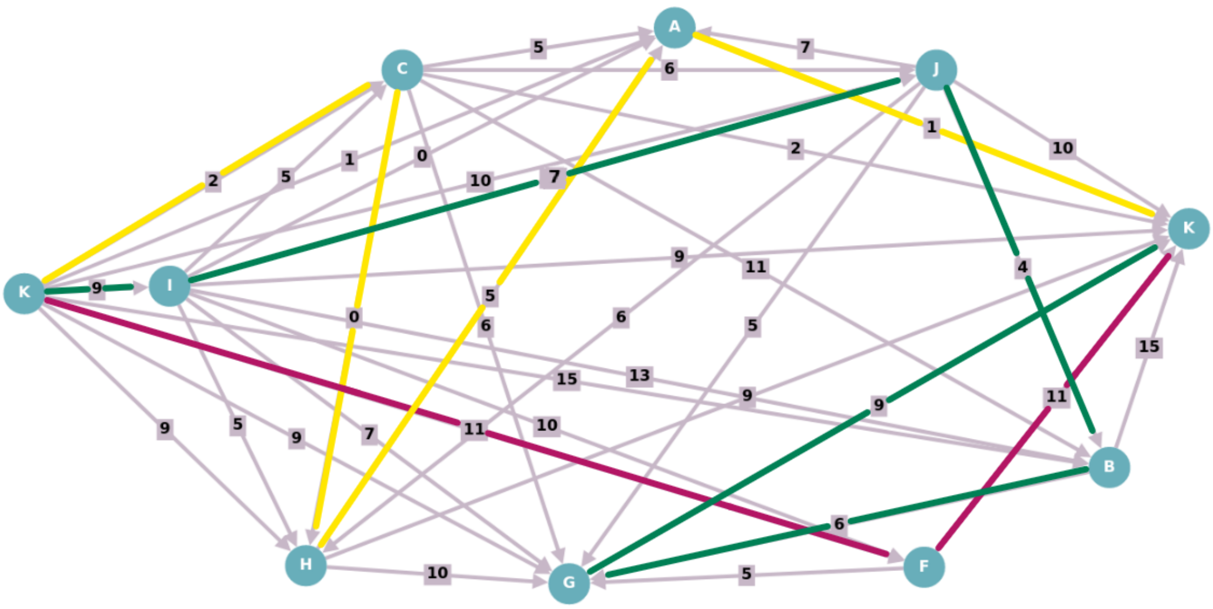


Figura 9: Percurso realizado por cada uma das equipas.

Para diferenciar o percurso realizado por cada uma das equipas utilizamos cores diferentes para esse efeito. Assim sendo, o menor custo, dado pela soma dos arcos percorridos, é 68 e os caminho possíveis são:

K→C→H→A→K
K→I→J→B→G→K
K→F→K

4. Validação do Modelo

Para validar o nosso modelo, tentamos resolver o problema através de um modelo de programação linear e verificámos que a solução obtida por este modelo é igual à solução obtida pela rede de fluxos.

➔ **Função Objetivo**

Minimizar o custo total.

$$min: \sum_{i,j \epsilon V} c_{ij} * x_{ij}, \quad x_{ij} \epsilon A$$

```
/* Objective function */
min: 2 xkc + 1 xka + 10 xkj + 9 xki + 15 xkb + 11 xkf + 9 xkg + 9 xkh +
     5 xic + 0 xia + 7 xij + 10 xik + 13 xib + 10 xif + 7 xig + 5 xih +
     0 xch + 6 xcg + 11 xcb + 3 xck + 6 xcj + 5 xca +
     2 xak +
     7 xja + 6 xjh + 5 xjg + 4 xjb + 11 xjk +
     5 xha + 10 xhk + 10 xhg +
     10 xgk +
     5 xfg + 12 xfk +
     6 xbg + 16 xbk;
```

Figura 10: Função objetivo inserida no LPSolve.

➔ **Restrições**

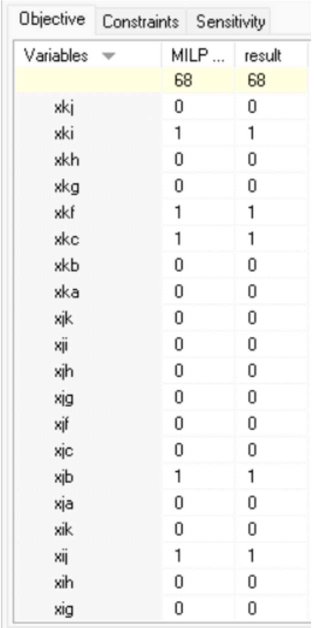
A cada vértice só deve chegar uma equipa exceto no vértice K que deve receber e enviar 3 equipas.

```
/* Variable bounds */
VA: xka + xca + xia + xja + xha = 1;
VA': xab + xac + xaf + xag + xah + xai + xaj + xak = 1;
VB: xkb + xib + xcb + xjb = 1;
VB': xba + xbc + xbf + xbg + xbh + xbi + xbj + xbk = 1;
VC: xkc + xic = 1;
VC': xca + xcb + xcf + xcg + xch + xci + xcj + xck = 1;
VF: xkf + xif = 1;
VF': xfa + xfb + xfc + xfg + xfh + xfi + xfj + xfk = 1;
VG: xkg + xig + xcg + xjg + xhg + xfg + xbg = 1;
VG': xga + xgb + xgc + xgf + xgh + xgi + xgj + xgk = 1;
VH: xkh + xih + xch + xjh = 1;
VH': xha + xhb + xhc + xhf + xhg + xhi + xhj + xhk = 1;
VI: xki = 1;
VI': xia + xib + xic + xif + xig + xih + xij + xik = 1;
VJ: xkj + xij + xcj = 1;
VJ': xja + xjb + xjc + xjf + xjg + xjh + xji + xjk = 1;
VK: xka + xkb + xkc + xkf + xkg + xkh + xki + xkj = 3;
VK': xak + xbk + xck + xfk + xgk + xhk + xik + xjk = 3;

bin xkc, xka, xkj, xki, xkb, xkf, xkg, xkh,
    xic, xia, xij, xik, xib, xif, xig, xih,
    xch, xcg, xcb, xck, xcj, xca,
    xak,
    xja, xjh, xjg, xjb, xjk,
    xha, xjh, xjg, xjb, xjk,
    xha, xhk, xhg,
    xgk,
    xgh, xfk,
    xbg, xbk;
```

Figura 11: Restrições inseridas no LPSolve.

➔ **Solução Ótima**



Objective | Constraints | Sensitivity

Variables	MILP ...	result
	68	68
xkj	0	0
xki	1	1
xkh	0	0
xkg	0	0
xkf	1	1
xkc	1	1
xkb	0	0
xka	0	0
xjk	0	0
xji	0	0
xjh	0	0
xjg	0	0
xjf	0	0
xjc	0	0
xjb	1	1
xja	0	0
xik	0	0
xij	1	1
xih	0	0
xig	0	0

Figura 12: Output produzido pelo programa LPSolve.

dest / orig	A	B	C	F	G	H	I	J	K	
A									X	11:00
B					X					10:45
C						X				10:00
F									X	10:30
G									X	11:15
H	X									10:45
I								X		09:30
J		X								10:15
K			X	X			X			09:00

Tabela 3: Output obtido no LPSolve traduzido no problema

A tabela introduzida anteriormente fica então marcada com representações dos caminhos efetuados pelas diferentes equipas. Os "X" assinalados podem ser interpretados como o arco percorrido a partir de um vértice origem para um vértice destino.

Assim, podemos confirmar que o resultado obtido foi o correto através da validação anterior. Foram garantidas as condições deste problema de que cada cliente só recebe uma das equipas e, também, que as mesmas chegam a horas ao cliente seguinte, tendo em conta o tempo deste percurso.