Universidade do Minho
Licenciatura em Engenharia Informática

Unidade Curricular de Investigação Operacional

Ano Letivo de 2022/2023

Trabalho Prático 2

Rodrigo Monteiro, Miguel Gramoso, Diogo Abreu e Miguel Marinha

e-mail: {a100706,a100835,a100646,a100651}@alunos.uminho.pt

Abstract. No âmbito da unidade curricular de Investigação Operacional, este trabalho resolve um problema de afetação (*assignment problem*): um número de equipas efetua serviços de clientes distribuídos geograficamente, sendo necessário fazer uma atribuição de equipas a todos os clientes com o objetivo de minimizar o custo total de deslocação, tendo em conta as restrições dos horários de início de serviço, da duração de serviço, e dos tempos de deslocação entre clientes, e entre clientes e a sede. Para a resolução deste problema foram utilizados os dados disponibilizados no enunciado do trabalho, isto é, a informação relativa aos clientes e as matrizes dos tempos e custos de deslocação. Além disso, foi utilizado o software de otimização *Relax4*, e o software *LPSolve* para a validação do modelo.

Keywords: Assignment Problem · Relax4 · LPSolve

Índice

1 Formulação do problema

1.1 Contextualização e dados

Dado um conjunto de clientes V, cada cliente $j \in V$ possui uma hora de início de serviço a_j, uma duração de serviço d_j que equivale a um quarto de hora para todos os clientes — a duração dos serviços é fixa —, e um tempo de deslocação para um destino i, t_{ji}.
Todas as medidas de tempo estão quantizadas em intervalos de um quarto de hora $(1/4\ h)$.
Assim, é possível definir a seguinte restrição:

$$j, i \in V:\ a_j + d_j + t_{ji} \leq a_i$$

Ou seja, uma equipa pode efetuar o serviço do cliente i, apenas se a soma do início de serviço de j com o tempo da duração de serviço e com o tempo de deslocação de j para i for menor do que o início de serviço relativo ao cliente i. Em termos mais simples, cada equipa tem de estar disponível na hora de início de serviço de um dado cliente, estando essa disponibilidade dependente do horário do cliente anterior e de tempos de deslocação.

Cada uma das equipas inicia o seu trabalho às 09:00 na sede da empresa localizada em Keleirós (K).
Na seguinte tabela encontram-se as informações relativas aos clientes:

Número	Nome	$a_j\ (\frac{1}{4}h)$	a_j (hora do serviço)
1	Ana	$a_1 = 1$	9:15
2	Beatriz	7	10:45
3	Carlos	4	10:00
4	Diogo	2	9:30
5	Eduardo	10	11:30
6	Francisca	6	10:30
7	Gonçalo	9	11:15
8	Helena	$a_8 = 9$	11:15
9	Inês	2	9:30
10	José	5	10:15

Seguindo as regras do enunciado, são encontrados os valores de a_1 e de a_8, e obtidas indicações de possíveis remoções de clientes a ter em conta.

Seja ABCDE o número de inscrição do estudante do grupo com maior número de inscrição. $ABCDE\ \rightarrow 100835$

- fazer $a_1 = B + 1 \Rightarrow a_1 = 0 + 1 = 1$;
- fazer $a_8 = C + 1 \Rightarrow a_1 = 8 + 1 = 9$;
- se D é par, remover o cliente D $\Rightarrow\ D = 3$, não ocorre remoção;
- se E é par, remover o cliente E $\Rightarrow\ E = 5$, não ocorre remoção.

Para além dessa informação, o enunciado também fornece dados acerca dos tempos e custos de deslocação através de duas matrizes simétricas.

	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

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

custos de deslocação

É importante, também, destacar que, para além dos custos de deslocação, cada equipa tem um custo fixo de 1 unidade de medida associado.

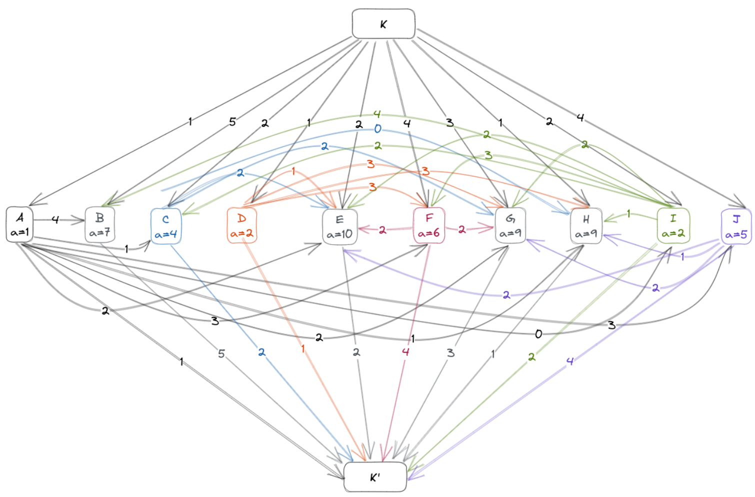
1.2 Objetivo

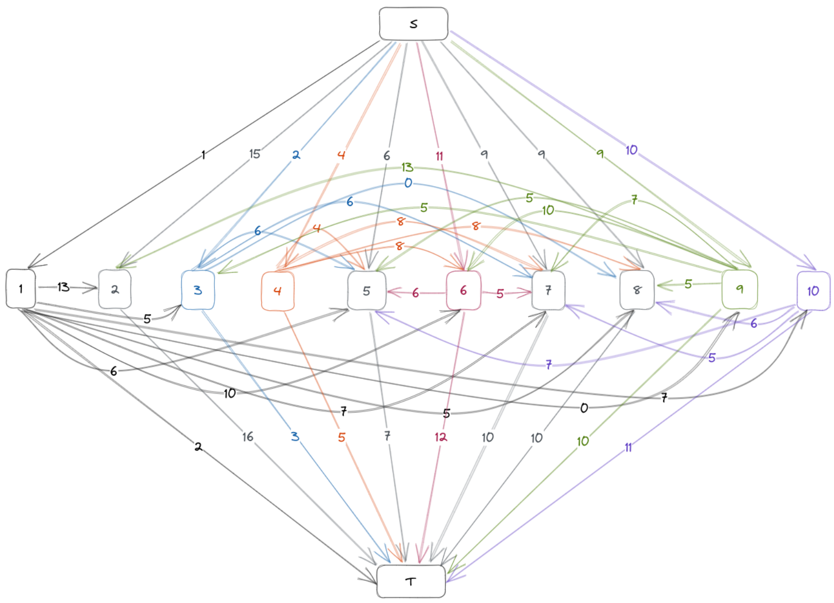
O objetivo principal deste trabalho é determinar o valor mínimo do custo total de deslocação das equipas entre clientes, e entre clientes e a sede. É necessário ter em conta que nem todas as rotas (arestas do grafo) são admissíveis, i.e. úteis ou necessárias, visto que existem restrições quanto ao início de serviço, duração de serviço e tempos de deslocação. Para isso, é necessário criar um *grafo de compatibilidades* e encontrar o fluxo de custo mínimo nessa rede, traduzindo o problema para um input do *Relax4*.

2 Rede do problema de fluxo mínimo

O grafo à direita foi utilizado para encontrar arestas admissíveis, isto é, que obedecem à restrição estabelecida anteriormente.
O valor nas arestas corresponde aos tempos de deslocação e o valor nos vértices corresponde ao início de serviço.

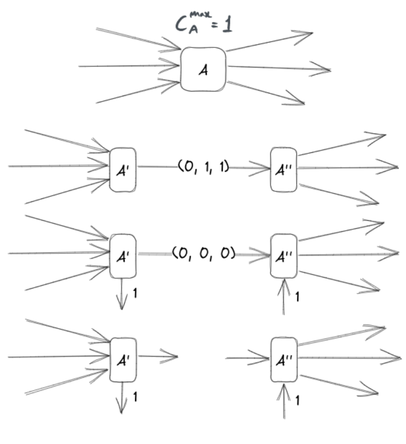
De seguida, foi, então, possível construir um grafo com os custos de deslocação correspondentes.



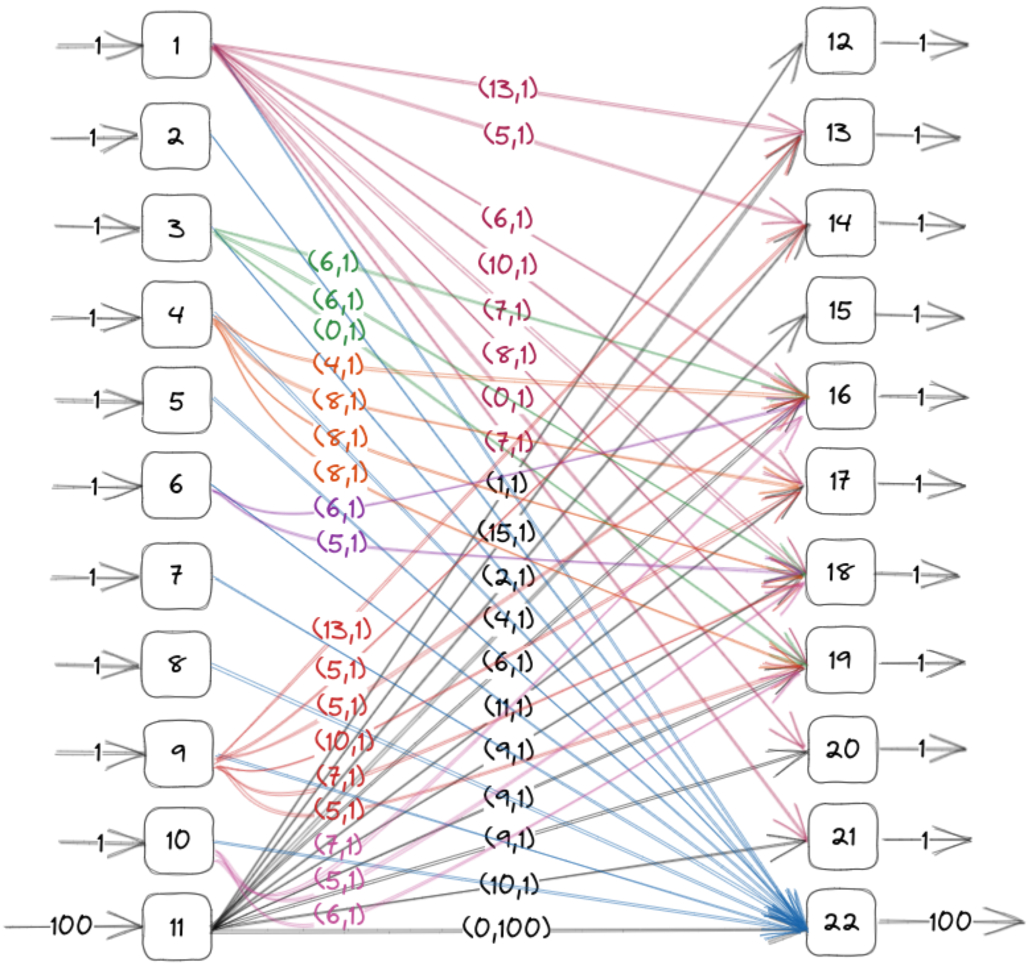


As arestas de retorno dos clientes para a sede (T) possuem a indicação do custo correspondente; este custo tem a adição de uma unidade, uma vez que cada equipa com serviço tem um custo fixo de 1 U.M. associado, o que, de acordo com a lógica do modelo produzido, é equivalente a qualquer equipa que regresse à sede (exceto, no caso de regresso por um arco de S a T, que será visto posteriormente).
Foi tomada esta decisão de modo a que o software que cria o output, portanto a solução ótima do problema, mostre já o valor ótimo tendo em conta este aspeto, não sendo necessário fazê-lo depois manualmente, ou por outros meios. Pode-se argumentar que esta abordagem tem a desvantagem de deixar o modelo "menos próximo da situação real", uma vez que os custos de regresso à sede são modificados. Apesar disso, achamos a decisão justificável, visto que é tomada com o objetivo de aumentar a praticidade do modelo.

No entanto, este grafo não traduz completamente, ou de forma suficiente, o problema de fluxo de custo mínimo. Para isso, é necessário criar um *grafo bipartido*, em que os clientes correspondem a dois vértices e não apenas a um. As arestas possuem um dado custo, um limite inferior igual a 0 e um limite superior igual a 1 — apenas uma equipa tem de efetuar o serviço relativamente a um dado cliente. Porém, para capacitar as ofertas e procuras nos vértices, as instâncias com capacidade nos vértices (clientes) são transformadas em instâncias apenas com capacidades nos arcos. Portanto, os vértices são desdobrados num veŕtice de entrada com procura igual a 1, e num vértice de saída com oferta igual a 1. Porém, como esses arcos ficam nulos, não têm de ser indicados. Deste modo, o modelo pode levar em consideração as restrições de capacidade dos vértices, de forma a minimizar o fluxo total — cada cliente só pode "receber" e "enviar/dispensar" uma equipa (este raciocínio também é usado na criação do input para o LPSolve, para a validação do modelo, como se pode verificar posteriormente neste relatório). Exemplificação da lógica utilizada:



Portanto, foi construído o seguinte grafo bipartido, em que os vértices numerados têm a sua correspondência à direita, que “representa o mesmo cliente”. Por exemplo, 1 e 12 representam o cliente 1, a Ana; 2 e 13 representam o cliente 2, a Beatriz; etc.



	12	13	14	15	16	17	18	19	20	21	22
1		(13,1)	(5,1)		(6,1)	(10,1)	(7,1)	(8,1)	(0,1)	(7,1)	(2,1)
2											(16,1)
3					(6,1)		(6,1)	(0,1)			(3,1)
4					(4,1)	(8,1)	(8,1)	(8,1)			(5,1)
5											(7,1)
6					(6,1)		(5,1)				(12,1)
7											(10,1)
8											(10,1)
9		(13,1)	(5,1)		(5,1)	(10,1)	(7,1)	(5,1)			(10,1)
10					(7,1)		(5,1)	(6,1)			(11,1)
11	(1,1)	(15,1)	(2,1)	(4,1)	(6,1)	(11,1)	(9,1)	(9,1)	(9,1)	(10,1)	(0,100)

É importante destacar que foi adicionado o arco de 11 a 22, da origem até ao destino, (ambos representam a sede) com custo 0 e capacidade igual a 100 — um limite superior suficientemente alto.
Como foi colocada uma oferta igual a 100 no vértice de origem (11), significando que no máximo serão utilizadas 100 equipas, as equipas que não precisam de satisfazer as procuras dos clientes, devido a estes já não precisarem, vão diretamente para o destino com procura de valor 100 por um arco com custo 0 e limite infinito, neste caso, atribuído com um valor de 100. Assim, a conservação de fluxo e de capacidade é mantida.

3. Otimização da rede

3.1 Ficheiro de input para o Relax4

```
22
47
1 13 13 1
1 14 5 1
1 16 6 1
1 17 10 1
1 18 7 1
1 19 8 1
1 20 0 1
1 21 7 1
1 22 2 1
2 22 16 1
3 16 6 1
3 18 6 1
3 19 0 1
3 22 3 1
4 16 4 1
4 17 8 1
4 18 8 1
4 19 8 1
4 22 5 1
5 22 7 1
6 16 6 1
6 18 5 1
6 22 12 1
7 22 10 1
8 22 10 1
9 13 13 1
9 14 5 1
9 16 5 1
9 17 10 1
9 18 7 1
9 19 5 1
9 22 10 1
10 16 7 1
10 18 5 1
10 19 6 1
10 22 11 1
11 12 1 1
11 13 15 1
11 14 2 1
11 15 4 1
11 16 6 1
11 17 11 1
11 18 9 1
11 19 9 1
11 20 9 1
11 21 10 1
11 22 0 100
1
1
1
1
1
1
1
1
1
1
100
-1
-1
-1
-1
-1
-1
-1
-1
-1
-1
-100
```

Nas primeiras duas linhas do ficheiro de input (.txt) é definida a dimensão da rede: na primeira linha é indicado o número de vértices (22), e na segunda linha o número de arcos (47). Nas próximas 47 linhas, número equivalente ao número de arcos, são listadas as características dos arcos da rede, i.e., os vértices de origem, os vértices de destino, os custos e os limites superiores. Por exemplo, “`1 14 5 1`” corresponde ao arco de 1 para 14 com custo 5 e limite superior igual a 1.

Nas restantes linhas (22, que equivale ao número de vértices), são definidos os valores das ofertas e das procuras de cada vértice. Os vértices 1 a 10 têm oferta igual a 1, o vértice 11 (origem/ sede) tem oferta igual a 100, os vértices 12 a 21 têm procura igual a 1, e o vértice 22 (destino/ sede) tem procura igual a 100.

3.2 Output do Relax4

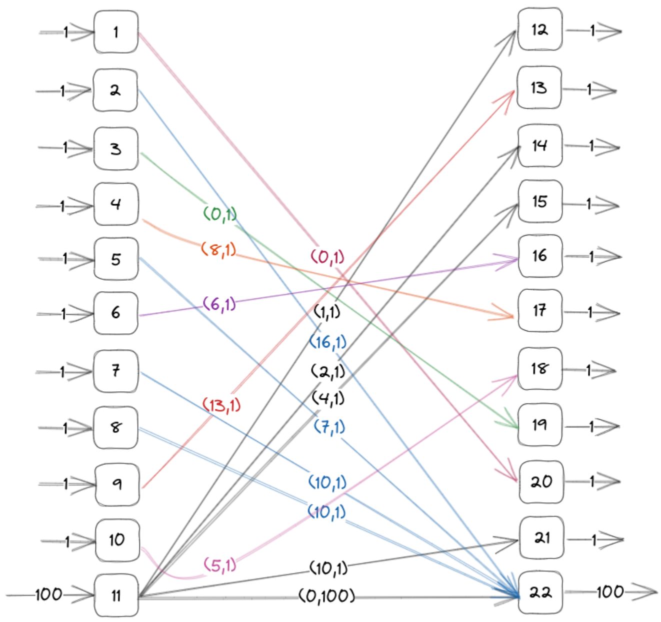
```
NUMBER OF NODES = 22, NUMBER OF ARCS = 47
Total algorithm solution time =   0.003262043 sec.
OPTIMAL COST =   92.
NUMBER OF ITERATIONS =  20
NUMBER OF MULTINODE ITERATIONS =  1
NUMBER OF MULTINODE ASCENT STEPS =  0
NUMBER OF REGULAR AUGMENTATIONS =  4
```

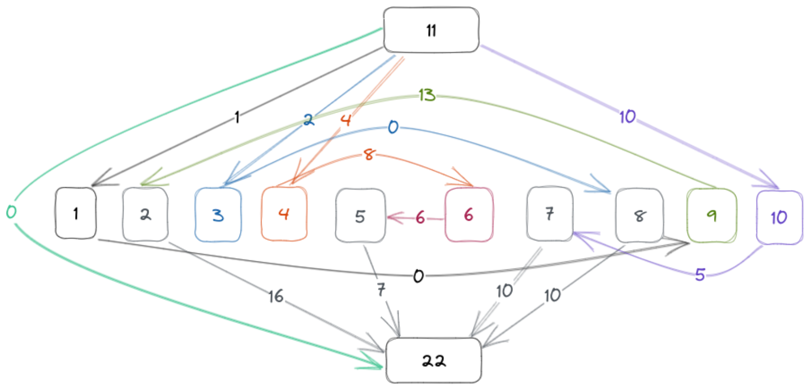
```
s 92.
f 1 13 0
f 1 14 0
f 1 16 0
f 1 17 0
f 1 18 0
f 1 19 0
f 1 20 1
f 1 21 0
f 1 22 0
f 2 22 1
f 3 16 0
f 3 18 0
f 3 19 1
f 3 22 0
f 4 16 0
f 4 17 1
f 4 18 0
f 4 19 0
f 4 22 0
f 5 22 1
f 6 16 1
f 6 18 0
f 6 22 0
f 7 22 1
f 8 22 1
f 9 13 1
f 9 14 0
f 9 16 0
f 9 17 0
f 9 18 0
f 9 19 0
f 9 22 0
f 10 16 0
f 10 18 1
f 10 19 0
f 10 22 0
f 11 12 1
f 11 13 0
f 11 14 1
f 11 15 1
f 11 16 0
f 11 17 0
f 11 18 0
f 11 19 0
f 11 20 0
f 11 21 1
f 11 22 96
```

(https://neos-server.org/neos/jobs/12990000/12990913.html)

3.4 Solução ótima

O output fornecido pelo Relax4 indica um custo ótimo de valor 92 na rede do problema. Para além disso, indica também a quantidade de fluxo que passou por cada arco; por exemplo, “f 1 20 1” indica que passou uma unidade de fluxo entre o vértice 1 e o vértice 20, ou seja, uma equipa deslocou-se do cliente 1 para o cliente 9 (corresponde ao vértice 20). Assim, a solução ótima pode ser indicada através do seguinte grafo bipartido.
É, também, apresentado um segundo grafo simplificado, de mais rápida compreensão.





Para esclarecer o funcionamento do grafo bipartido, será explicada uma das rotas apresentadas no output realizada por uma das equipas.
Primeiramente, passa pelo arco de 11 a 15 uma unidade de fluxo, visto que o limite superior do arco é 1. O vértice 15 tem uma procura de valor 1, portanto essa procura fica satisfeita. No vértice 4 existe uma oferta de valor 1, e este valor passa no arco de 4 a 17, satisfazendo a procura do vértice 17.
É importante realçar que a lógica utilizada na rede possibilita que um cliente possa apenas “receber” e “enviar” uma equipa. Pelo que foi visto até agora, o cliente 4 já “recebeu” uma equipa (11 a 15), e já não receberá mais pois a procura já está satisfeita; para além disso, já “enviou” uma equipa (4 a 17), e já não enviará mais pois a sua oferta de 1 já entrou no fluxo.
Assim sendo, de seguida, a oferta de valor 1 do vértice 6 percorre o arco de 6 a 16 e satisfaz a procura desse vértice. Por fim, a oferta do vértice 5 percorre o arco de 5 a 22, reduzindo a procura nesse vértice de destino em uma unidade.

A seguir encontram-se 4 tabelas com as informações acerca dos serviços atribuídos às equipas (numeradas arbitrariamente, sem ordem específica).

j	cliente	a_j $(\frac{1}{4}h)$	a_j	$a_j + d_j$	tempos de deslocação	custos de deslocação
	Keleirós	0		9:00	[KA]: 1/4 h	1
1	Ana	1	9:15	9:30	[AI]: 0 h	0
9	Inês	2	9:30	9:45	[IB]: 4/4 h	13
2	Beatriz	7	10:45	11:00	[BK]: 5/4 h	15
	Keleirós	13	12:15			1
	custo de operação da equipa **1**					30

j	cliente	a_j $(\frac{1}{4}h)$	a_j	$a_j + d_j$	tempos de deslocação	custos de deslocação
	Keleirós	0		9:00	[KC]: 2/4 h	2
3	Carlos	4	10:00	10:15	[CH]: 0 h	0
8	Helena	9	11:15	11:30	[HK]: 1/4 h	9
	Keleirós	11	11:45			1
	custo de operação da equipa **2**					12

j	cliente	a_j $(\frac{1}{4}h)$	a_j	$a_j + d_j$	tempos de deslocação	custos de deslocação
	Keleirós	0		9:00	[KD]: 1/4 h	4
4	Diogo	2	9:30	9:45	[DF]: 3/4 h	8
6	Francisca	6	10:30	10:45	[FE]: 2/4 h	6
5	Eduardo	10	11:30	11:45	[EK]: 2/4 h	6
	Keleirós	13	12:15			1
	custo de operação da equipa **3**					25

j	cliente	a_j $(\frac{1}{4}h)$	a_j	$a_j + d_j$	tempos de deslocação	custos de deslocação
	Keleirós	0		9:00	[KJ]: 4/4 h	10
10	José	5	10:15	10:30	[JG]: 2/4 h	5
7	Gonçalo	9	11:15	11:30	[GK]: 3/4 h	9
	Keleirós	13	12:15			1
	custo de operação da equipa **4**					25

<u>Custo total</u>: $30 + 12 + 25 + 25 + (96 \times 0) = 92$

4. Validação do modelo

4.1 Input para o LPSolve

```
min:   1 x100to1 + 15 x100to2 + 2 x100to3 + 4 x100to4 + 6 x100to5 + 11 x100to6
       + 9 x100to7 + 9 x100to8 + 9 x100to9 + 10 x100to10 +
       13 x1to2   + 5 x1to3  + 6 x1to5   + 10 x1to6  + 7 x1to7   + 5  x1to8   + 0 x1to9
       + 7 x1to10  +
       6  x3to5   + 6 x3to7  + 0 x3to8   +
       4  x4to5   + 8 x4to6  + 8 x4to7   + 8 x4to8   +
       6  x6to5   + 5 x6to7  +
       13 x9to2   + 5 x9to3  + 5 x9to5   + 10 x9to6  + 7 x9to7   + 5  x9to8   +
       7  x10to5  + 5 x10to7   + 6 x10to8  +
       2  x1to100 + 16 x2to100 + 3 x3to100 + 5 x4to100 + 7 x5to100 + 12 x6to100
       + 10 x7to100 + 10 x8to100 + 10 x9to100 + 11 x10to100;

to1  = x100to1;
to2  = x100to2  + x1to2  + x9to2;
to3  = x100to3  + x1to3  + x9to3;
to4  = x100to4;
to5  = x100to5  + x1to5  + x3to5 + x4to5 + x6to5 + x9to5 + x10to5;
to6  = x100to6  + x1to6  + x4to6  + x9to6;
to7  = x100to7  + x1to7  + x3to7 + x4to7 + x6to7 + x9to7 + x10to7;
to8  = x100to8  + x1to8  + x3to8 + x4to8 + x9to8 + x10to8;
to9  = x100to9  + x1to9;
to10 = x100to10 + x1to10;

to1  = 1; to2  = 1; to3  = 1; to4  = 1; to5  = 1;
to6  = 1; to7  = 1; to8  = 1; to9  = 1; to10 = 1;

x1to100 + x2to100 + x3to100 + x4to100 + x5to100 + x6to100 + x7to100 + x8to100 + x9to100
+ x10to100 = x100to1 + x100to2 + x100to3 + x100to4 + x100to5 + x100to6 + x100to7 +
x100to8 + x100to9 + x100to10;

from1  = x1to2 + x1to3 + x1to5 + x1to6 + x1to7 + x1to8 + x1to9 + x1to10 + x1to100;
from2  = x2to100;
from3  = x3to5 + x3to7 + x3to8 + x3to100;
from4  = x4to5 + x4to6 + x4to7 + x4to8 + x4to100;
from5  = x5to100;
from6  = x6to5 + x6to7 + x6to100;
from7  = x7to100;
from8  = x8to100;
from9  = x9to2 + x9to3 + x9to5 + x9to6 + x9to7 + x9to8 + x9to100;
from10 = x10to5 + x10to7 + x10to8 + x10to100;

from1  = 1; from2  = 1; from3  = 1; from4  = 1; from5  = 1;
from6  = 1; from7  = 1; from8  = 1; from9  = 1; from10 = 1;

bin x100to1, x100to2, x100to3, x100to4, x100to5, x100to6, x100to7, x100to8, x100to9,
x100to10, x1to2, x1to3, x1to5, x1to6, x1to7, x1to8, x1to9, x1to10, x3to5, x3to7, x3to8,
x4to5, x4to6, x4to7, x4to8, x6to5, x6to7, x9to2, x9to3, x9to5, x9to6, x9to7, x9to8,
x10to5, x10to7, x10to8, x1to100, x2to100, x3to100, x4to100, x5to100, x6to100, x7to100,
x8to100, x9to100, x10to100;
```

O objetivo é minimizar a soma das variáveis de decisão multiplicadas pelos seus custos.
As variáveis de decisão representam a quantidade de fluxo que passa entre dois vértices do grafo (por um arco admissível): `x100to1`, `x1to2`, etc. No problema em caso, essa quantidade é sempre 0 ou 1.
Os parâmetros são os custos relativos a esses arcos/rotas, já vistos anteriormente na formulação do problema.
As restrições são feitas utilizando variáveis auxiliares que representam a quantidade de arestas que entram em cada vértice e que saem de cada vértice. As restrições a essas variáveis são feitas de forma a que cada vértice seja apenas um destino para uma aresta e uma origem para outra aresta, visto que cada cliente só pode "receber" e "enviar/dispensar" uma equipa. Por exemplo, existem 3 arestas que têm como destino o vértice 2, portanto, "`to2 = x100to2 + x1to2 + x9to2;`"; e existe uma aresta que tem como origem o vértice 2, "`from2 = x2to100;`". De maneira a satisfazer todos os clientes, todas as variáveis "toX" têm de ser iguais a 1, e como as equipas têm de avançar para outro cliente ou voltar para a sede depois de efetuar um serviço, as variáveis "fromX" têm de ser, também, iguais a 1. Para além disso, também é necessário impor a seguinte restrição: todas as equipas que saem da sede têm de voltar à sede (`x1to100 + ... = x100to1 + ...`).

4.2 Output do LPSolve

```
Value of objective function: 92.00000000

Actual values of the variables:
x100to1                     1
x100to2                     0
x100to3                     1
x100to4                     1
x100to5                     0
x100to6                     0
x100to7                     0
x100to8                     0
x100to9                     0
x100to10                    1
x1to2                       0
x1to3                       0
x1to5                       0
x1to6                       0
x1to7                       0
x1to8                       0
x1to9                       1
x1to10                      0
x3to5                       0
x3to7                       0
x3to8                       1
x4to5                       0
x4to6                       1
x4to7                       0
x4to8                       0
x6to5                       1
x6to7                       0
x9to2                       1
x9to3                       0
x9to5                       0
x9to6                       0
x9to7                       0
x9to8                       0
x10to5                      0
x10to7                      1
x10to8                      0
x1to100                     0
x2to100                     1
x3to100                     0
x4to100                     0
x5to100                     1
x6to100                     0
x7to100                     1
x8to100                     1
x9to100                     0
x10to100                    0
to1                         1
to2                         1
to3                         1
to4                         1
to5                         1
to6                         1
to7                         1
to8                         1
to9                         1
to10                        1
from1                       1
from2                       1
from3                       1
from4                       1
from5                       1
from6                       1
from7                       1
from8                       1
from9                       1
from10                      1
```

Assim, através do LPSolve, foi possível chegar ao mesmo valor de solução ótima.

5 Conclusão

Durante a realização deste trabalho prático, foi possível consolidar diversos conceitos abordados nas aulas teóricas e práticas da unidade curricular Investigação Operacional, nomeadamente, grafos bipartidos, redes com capacidades, e vértices com capacidades, de modo a resolver o problema proposto. Assim, e com o auxílio do software de otimização Relax4, foi possível descobrir uma solução ótima através de uma correta tradução do problema e da aplicação desses conceitos dados.
Portanto, consideramos que este trabalho foi realizado com sucesso, pois, por fim, foi efetuada uma validação do modelo que assegura a solução ótima obtida.