

Investigação Operacional

Trabalho Prático 1

Licenciatura em Engenharia Informática, Universidade do Minho

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

Braga, 22 de março de 2022

1. Formulação do Problema

Este problema baseia-se no percurso de um drone que deverá percorrer linhas de alta tensão para averiguar se há vegetação a perturbar as mesmas. Para efetuar o seu trajeto, o drone tem que passar por todas as linhas pelo menos uma vez, tendo início em I e fim em F. Assim, o objetivo deste problema é minimizar a distância percorrida pelo drone.

O grafo inicial sofreu alterações conforme as restrições propostas no enunciado. De acordo com o número do aluno mais elevado, 97588, removeu-se as arestas D e E, tal como se pode ver na imagem seguinte:

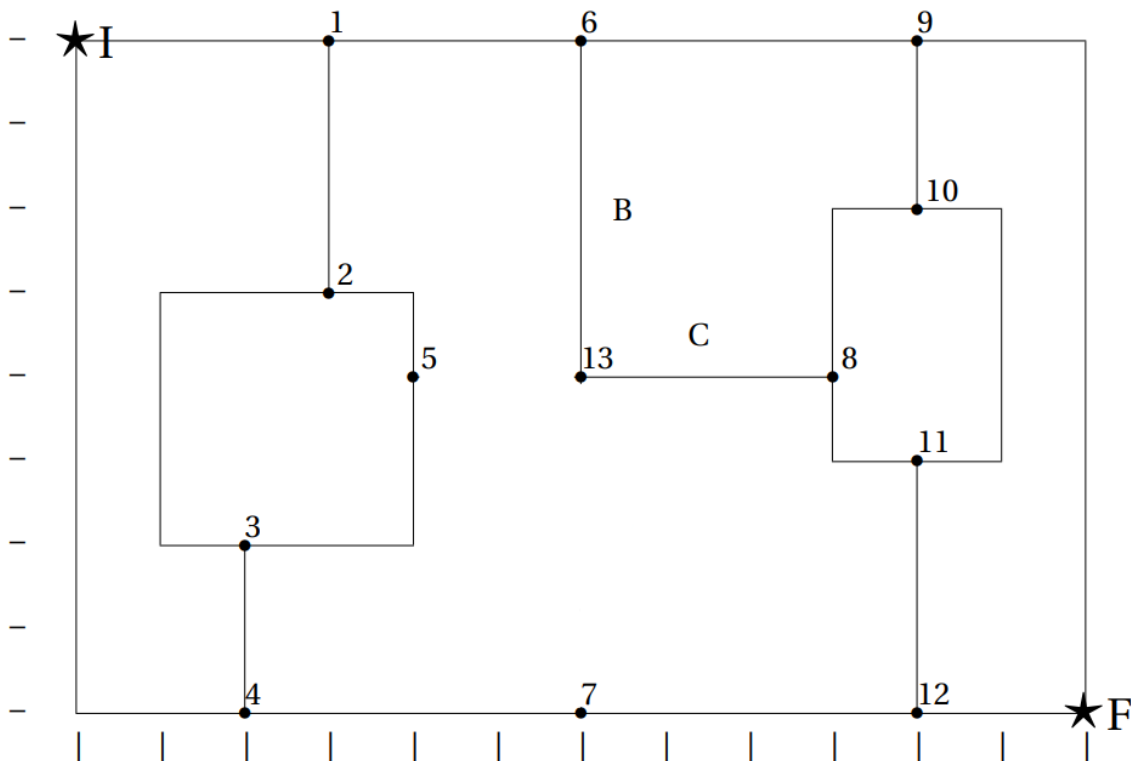


Figura 1: Esquema inicial com alterações.

Desenvolvendo o problema, concluímos que o drone deve seguir um caminho Euleriano. Todas as arestas devem, por isso, ser percorridas exatamente uma vez e os vértices devem ser de grau par, exceto o inicial e o final que devem ter grau ímpar.

Obtemos assim o exercício a ser resolvido neste projeto: que arestas devemos adicionar ao mapa de forma a criar um caminho Euleriano e a minimizar a distância total percorrida pelo drone. É importante referir que esta distância a ser minimizada corresponde apenas ao comprimento das arestas que iremos adicionar ao mapa, uma vez que podemos descartar o comprimento das linhas de alta tensão, pois o drone tem que percorrê-las obrigatoriamente, ou seja, o seu valor será constante.

Desta forma, na figura seguinte podemos verificar os vértices que teremos que incluir no nosso modelo, visto que são os vértices que necessitam de arestas extras.

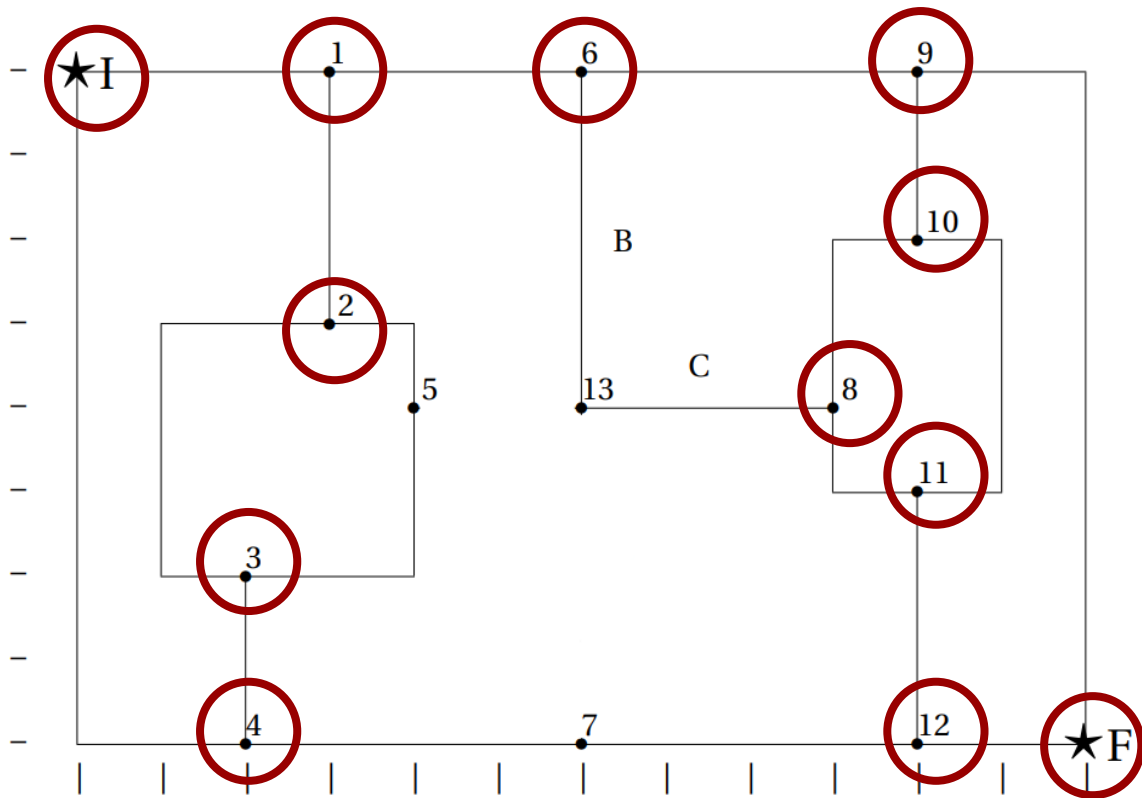


Figura 2: Vértices que necessitam de arestas.

2. Modelo

Neste modelo recorreu-se a um sistema numérico de base 16 para que fosse possível representar todos os vértices do esquema. Assim, temos que ter em conta que o vértice '0' corresponde a I, os de 'a' a 'c' correspondem aos de 10 a 13 e o vértice 'e' corresponde a F.

Desta forma, podemos definir $V=\{0,1,2,3,4,6,8,9,a,b,c,e\}$ como o conjunto das numerações de vértices a serem utilizados no modelo.

→ Variáveis de Decisão

x_{ij} : existe ou não uma aresta a unir o vértice i ao vértice j , $i, j \in V$, $i < j$

$x_{ij} \in \{0, 1\}$;

NOTA: Utilizamos a restrição $i < j$, por razões de simplificação, uma vez que x_{ij} e x_{ji} representam a mesma aresta. Assim, para representar a aresta que une os vértices 1 e 2 utilizamos x_{12} e não x_{21} .

→ Parâmetros

d_{ij} : comprimento da aresta que une os vértices i e j .

Os comprimentos de todas as arestas podem ser calculados a partir da seguinte tabela:

		x															
		0	3	3	2	2	4	6	6	9	10	10	10	10	6	12	
		8	8	5	2	0	4	8	0	4	8	6	3	0	4	0	
x	y	1	1	2	3	4	5	6	7	8	9	10	11	12	13	F	
0	8	1	0,00	3,00	4,24	6,32	8,25	5,66	6,00	10,00	9,85	10,00	10,20	11,18	12,81	7,21	14,42
3	8	1		0,00	3,00	6,08	8,06	4,12	3,00	8,54	7,21	7,00	7,28	8,60	10,63	5,00	12,04
3	5	2			0,00	3,16	5,10	1,41	4,24	5,83	6,08	7,62	7,07	7,28	8,60	3,16	10,30
2	2	3				0,00	2,00	2,83	7,21	4,47	7,28	10,00	8,94	8,06	8,25	4,47	10,20
2	0	4					0,00	4,47	8,94	4,00	8,06	11,31	10,00	8,54	8,00	5,66	10,00
4	4	5						0,00	4,47	4,47	5,00	7,21	6,32	6,08	7,21	2,00	8,94
6	8	6							0,00	8,00	5,00	4,00	4,47	6,40	8,94	4,00	10,00
6	0	7								0,00	5,00	8,94	7,21	5,00	4,00	4,00	6,00
9	4	8									0,00	4,12	2,24	1,41	4,12	3,00	5,00
10	8	9										0,00	2,00	5,00	8,00	5,66	8,25
10	6	10											0,00	3,00	6,00	4,47	6,32
10	3	11												0,00	3,00	4,12	3,61
10	0	12													0,00	5,66	2,00
6	4	13														0,00	7,21
12	0	F															0,00

Figura 3: Distâncias euclidianas entre os vértices.

→ Função Objetivo

Minimizar a distância percorrida, ou seja, o menor comprimento das arestas a adicionar.

$$\min: \sum_{i,j \in V} d_{ij} \times x_{ij}, i < j$$

```
/* Objective function */
min:3.00 x01+ 4.24 x02+ 6.32 x03+ 8.25 x04+ 6.00 x06+ 9.85 x08+ 10.00 x09+ 10.20 x0a+ 11.18 x0b+ 12.81 x0c+ 14.42 x0e+
      3.00 x12+ 6.08 x13+ 8.06 x14+ 3.00 x16+ 7.21 x18+ 7.00 x19+ 7.28 x1a+ 8.60 x1b+ 10.63 x1c+ 12.04 x1e+
      3.16 x23+ 5.10 x24+ 4.24 x26+ 6.08 x28+ 7.62 x29+ 7.07 x2a+ 7.28 x2b+ 8.60 x2c+ 10.30 x2e+
      2.00 x34+ 7.21 x36+ 7.28 x38+ 10.00 x39+ 8.94 x3a+ 8.06 x3b+ 8.25 x3c+ 10.20 x3e+
      8.94 x46+ 8.06 x48+ 11.31 x49+ 10.00 x4a+ 8.54 x4b+ 8.00 x4c+ 10.00 x4e+
      5.00 x68+ 4.00 x69+ 4.47 x6a+ 6.40 x6b+ 8.94 x6c+ 10.00 x6e+
      4.12 x89+ 2.24 x8a+ 1.41 x8b+ 4.12 x8c+ 5.00 x8e+
      2.00 x9a+ 5.00 x9b+ 8.00 x9c+ 8.25 x9e+
      3.00 xab+ 6.00 xac+ 6.32 xae+
      3.00 xbc+ 3.61 xbe+
      2.00 xce;
```

Figura 4: Função objetivo inserida no LPSolve.

→ Restrições

Cada vértice deve pertencer a uma só aresta.

$$\forall i \in V: \sum_{j \in V, i < j} x_{ij} = 1$$

```
/* Variable bounds */
V0: x01 + x02 + x03 + x04 + x06 + x08 + x09 + x0a + x0b + x0c + x0e = 1;
V1: x01 + x12 + x13 + x14 + x16 + x18 + x19 + x1a + x1b + x1c + x1e = 1;
V2: x02 + x12 + x23 + x24 + x26 + x28 + x29 + x2a + x2b + x2c + x2e = 1;
V3: x03 + x13 + x23 + x34 + x36 + x38 + x39 + x3a + x3b + x3c + x3e = 1;
V4: x04 + x14 + x24 + x34 + x46 + x48 + x49 + x4a + x4b + x4c + x4e = 1;
V6: x06 + x16 + x26 + x36 + x46 + x68 + x69 + x6a + x6b + x6c + x6e = 1;
V8: x08 + x18 + x28 + x38 + x48 + x68 + x89 + x8a + x8b + x8c + x8e = 1;
V9: x09 + x19 + x29 + x39 + x49 + x69 + x89 + x9a + x9b + x9c + x9e = 1;
Va: x0a + x1a + x2a + x3a + x4a + x6a + x8a + x9a + xab + xac + xae = 1;
Vb: x0b + x1b + x2b + x3b + x4b + x6b + x8b + x9b + xab + xbc + xbe = 1;
Vc: x0c + x1c + x2c + x3c + x4c + x6c + x8c + x9c + xac + xbc + xce = 1;
Ve: x0e + x1e + x2e + x3e + x4e + x6e + x8e + x9e + xae + xbe + xce = 1;

bin x01, x02, x03, x04, x06, x08, x09, x0a, x0b, x08c, x0e,
    x12, x13, x14, x16, x18, x19, x1a, x1b, x1c, x1e,
    x23, x24, x26, x28, x29, x2a, x2b, x2c, x2e,
    x34, x36, x38, x39, x3a, x3b, x3c, x3e,
    x46, x48, x49, x4a, x4b, x4c, x4e,
    x68, x69, x6a, x6b, x6c, x6e,
    x89, x8a, x8b, x8c, x8e,
    x9a, x9b, x9c, x9e,
    xab, xac, xae,
    xbc, xbe,
    xce;
```

Figura 5: Restrições inseridas no LPSolve.

3. Solução Ótima

Chegamos à solução após inserir no LPSolve o modelo do problema e obtivemos uma solução ótima onde podemos verificar quais as arestas a acrescentar para unir os vértices e totalizar o menor comprimento possível, neste caso, um comprimento de 14,65.

Variables	MILP Feasible	result ▾
	14,65	14,65
x02	1	1
x16	1	1
x34	1	1
x8b	1	1
x9a	1	1
xce	1	1
x01	0	0
x03	0	0
x04	0	0
x06	0	0
x08	0	0
x09	0	0
x0a	0	0
x0b	0	0
x0c	0	0
x0e	0	0
x12	0	0
x13	0	0
x14	0	0
x18	0	0
x19	0	0
x1a	0	0

Figura 6: Output produzido pelo programa LPSolve.

A solução executada pelo LPSolve anteriormente traduz a seguinte figura:

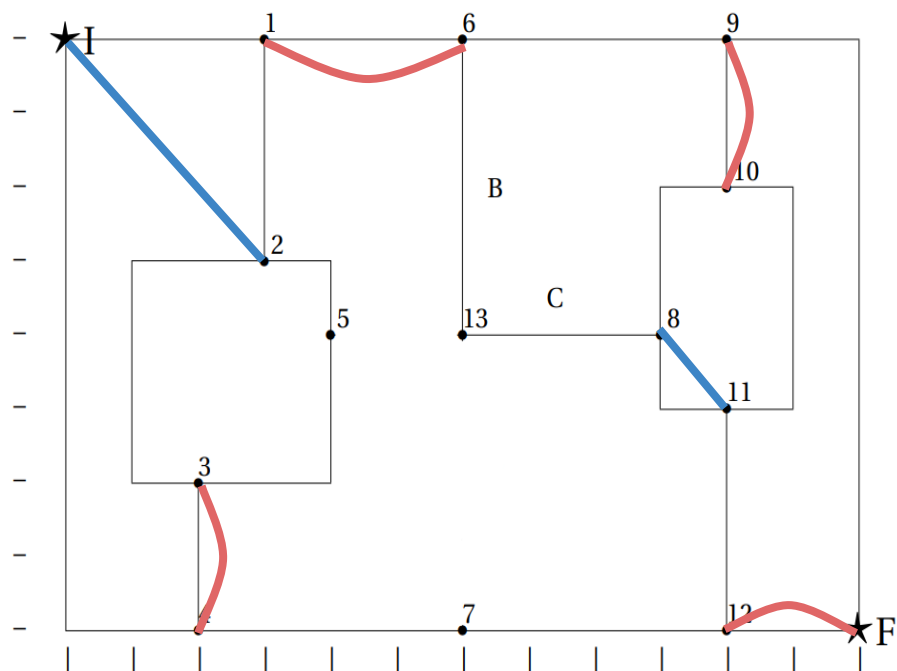


Figura 7: Arestas que devem ser inseridas no esquema.

Entre as arestas que foram inseridas temos que percorrer, por um lado, as arestas a azul que correspondem a ligações que inicialmente não existiam e, por outro, as vermelhas que são arestas que já existiam anteriormente no percurso do drone.

Assim, podemos definir um caminho Euleriano para o drone cumprir, dado que todos os vértices do mapa têm grau par exceto o inicial e o final.

Para além disso, visto que o caminho terá que incluir apenas uma travessia de cada aresta, a distância total percorrida pelo drone será sempre a mesma, sendo que não é necessário minimizá-la. Por conseguinte, resta apresentar um possível trajeto a ser investigado pelo drone:

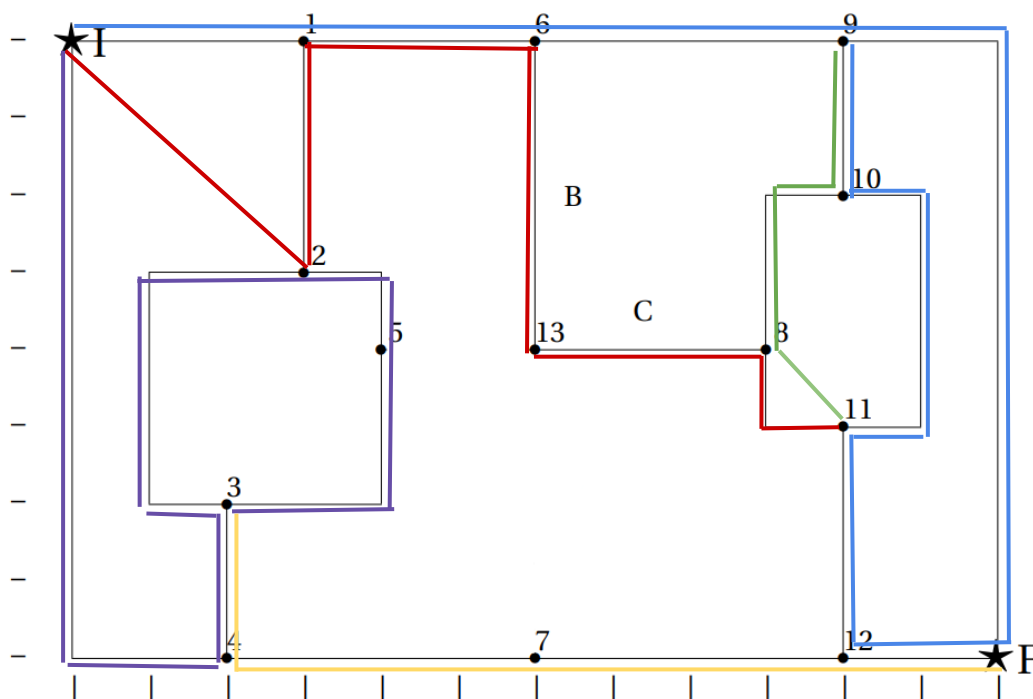


Figura 8: Possível solução do problema.

Nesta solução, o drone começa por percorrer a linha azul, de seguida a linha verde, depois a linha vermelha, a linha roxa e por fim, a linha amarela.

Desta forma, para o percurso definido a partir deste mapa, a distância total percorrida pelo drone será a soma dos comprimentos das linhas de alta tensão com o comprimento das arestas que adicionamos, ou seja, $79 + 14,65 = 93,65$.

4. Validação do Modelo

Para a validação do modelo, temos que verificar se todas as restrições são cumpridas, isto é, verificar se criamos apenas uma aresta para cada vértice em causa.

Desta forma, a seguinte tabela ilustra as restrições do nosso modelo:

j\i		0 (I)	1	2	3	4	6	8	9	a (10)	b (11)	c (12)	e (F)	Total
0 (I)	V0:		0	1	0	0	0	0	0	0	0	0	0	1
1	V1:	0		0	0	0	1	0	0	0	0	0	0	1
2	V2:	1	0		0	0	0	0	0	0	0	0	0	1
3	V3:	0	0	0		1	0	0	0	0	0	0	0	1
4	V4:	0	0	0	1		0	0	0	0	0	0	0	1
6	V6:	0	1	0	0	0		0	0	0	0	0	0	1
8	V8:	0	0	0	0	0	0		0	0	1	0	0	1
9	V9:	0	0	0	0	0	0	0		1	0	0	0	1
a (10)	Va:	0	0	0	0	0	0	0	1		0	0	0	1
b (11)	Vb:	0	0	0	0	0	0	1	0	0		0	0	1
c (12)	Vc:	0	0	0	0	0	0	0	0	0	0		1	1
e (F)	Ve:	0	0	0	0	0	0	0	0	0	0	1		1

Figura 9: As restrições são cumpridas.