

Universidade do Minho
Investigação Operacional
Trabalho Prático I

Ano letivo 2021/2022

a97363, Gabriel Alexandre Monteiro da Silva

a96106, Miguel Silva Pinto

a97755, Orlando José da Cunha Palmeira

a97613, Pedro Miguel Castilho Martins

1. Formulação do problema

Neste problema, um drone tem de percorrer todas as linhas de alta tensão, partindo de um ponto inicial **I** e acabando num ponto final **F**, para verificar se existe alguma vegetação a interferir com as linhas. Pretendemos que o drone atravessasse todas as linhas percorrendo a menor distância possível. O drone pode viajar pelas linhas em qualquer sentido e também pode viajar pelo ar.

Tendo em conta que o maior nº de aluno do grupo é 97755, a rede de linhas não sofre qualquer alteração, mantendo-se a original.

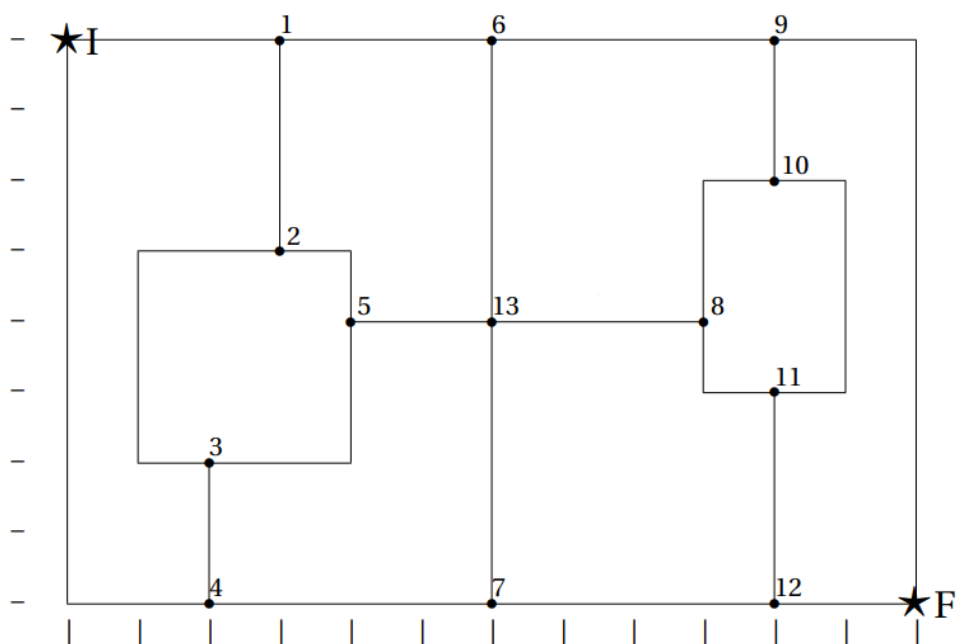


Fig.1 : Rede de linhas de alta tensão.

Após a análise do problema, chegamos à conclusão que precisávamos de criar um caminho Euleriano de modo a que o drone passe em todas as arestas, começando num determinado ponto inicial e acabando num determinado ponto final. Para que um grafo tenha um caminho Euleriano, os vértices iniciais (**I**) e finais (**F**) têm de ter grau ímpar, enquanto que todos os restantes vértices têm de ter grau par. Como o nosso grafo de linhas de alta tensão não apresenta estas condições, teremos que lhe adicionar arestas para que seja possível a existência de um caminho Euleriano.

Assim o nosso problema de otimização é: *quais arestas temos de adicionar de modo a que o drone percorra a menor distância possível e seja um caminho Euleriano?*

Como referimos anteriormente, para que o nosso grafo tenha um caminho Euleriano, é preciso que os vértices iniciais e finais tenham grau ímpar e os restantes vértices tenham grau par. No nosso grafo, os vértices iniciais e finais têm grau par e os

restantes vértices têm grau ímpar, exceto o vértice 13, pelo que precisamos de ligar estes vértices com arestas de maneira a corrigir a paridade dos seus graus. Na figura seguinte (Fig. 2) estão assinalados a cor vermelha, os vértices com grau ímpar que precisam de ter grau par, com a cor amarela estão os vértices com grau par e que deveriam ter grau ímpar, e a verde, os vértices com a paridade de grau do vértice correta para que o grafo obedeça às condições de existência de um caminho Euleriano.

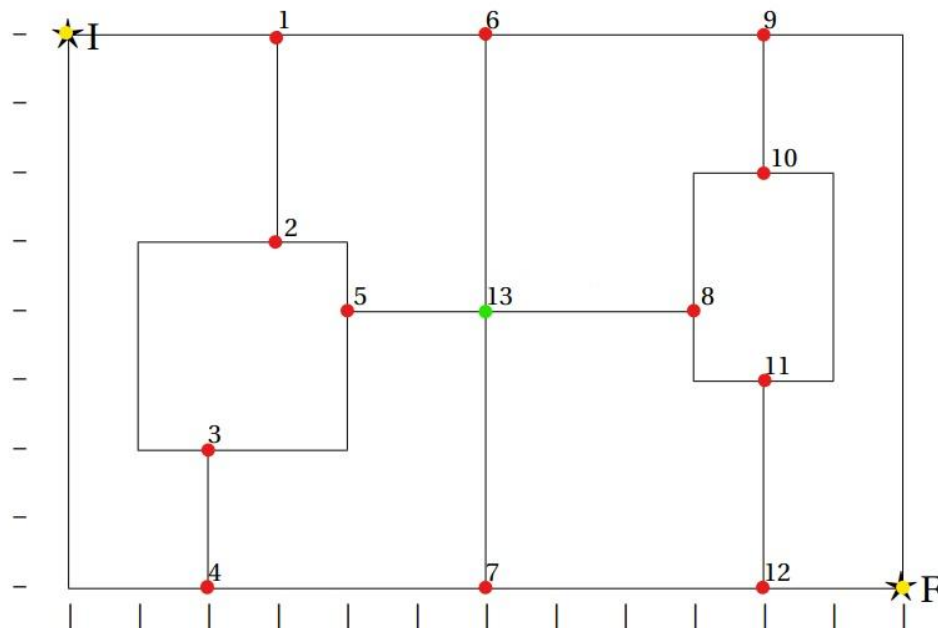


Fig.2 : Figura de classificação de paridade do grau dos vértices.

2. Modelo

Para representar os vértices nas variáveis de decisão, decidimos usar o conjunto $V = \{I, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, F\}$. Não contabilizamos o vértice 13 neste conjunto pois o grau deste vértice já é par, não sendo necessário adicionar-lhe nenhuma aresta.

2.1. Variáveis de Decisão

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

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

Como as arestas podem ser percorridas em qualquer direção assumimos que

$x_{ij} = x_{ji}$ e então apenas representamos x_{ij} : $i < j$, $i, j \in V$, considerando o vértice I (inicial) e o vértice F (final), o menor e maior elemento do conjunto V, respetivamente.

```

bin xI1, xI2, xI3, xI4, xI5, xI6, xI7, xI8, xI9, xI_10, xI_11, xI_12, xIF,
    x12, x13, x14, x15, x16, x17, x18, x19, x1_10, x1_11, x1_12, x1F,
    x23, x24, x25, x26, x27, x28, x29, x2_10, x2_11, x2_12, x2F,
    x34, x35, x36, x37, x38, x39, x3_10, x3_11, x3_12, x3F,
    x45, x46, x47, x48, x49, x4_10, x4_11, x4_12, x4F,
    x56, x57, x58, x59, x5_10, x5_11, x5_12, x5F,
    x67, x68, x69, x6_10, x6_11, x6_12, x6F,
    x78, x79, x7_10, x7_11, x7_12, x7F,
    x89, x8_10, x8_11, x8_12, x8F,
    x9_10, x9_11, x9_12, x9F,
    x10_11, x10_12, x10F,
    x11_12, x11F,
    x12F;

```

Fig.3 : Variáveis de Decisão.

2.2. Parâmetros

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

Comprimento de todas as arestas:

		x	0	3	3	2	2	4	6	6	9	10	10	10	10	6	12
		y	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

Fig.4 : Distâncias entre todos os vértices.

2.3. Função Objetivo

Minimizar o comprimento total das arestas a adicionar.

$\min: \sum(c_{ij} \times x_{ij}), i, j \in V, i < j$

```

/* Objective function */
min: 3 xI1 + 4.24 xI2 + 6.32 xI3 + 8.25 xI4 + 5.66 xI5 + 6.00 xI6 + 10.00 xI7 + 9.85 xI8 + 10.00 xI9 + 10.20 xI_10 + 11.18 xI_11 + 12.81 xI_12 + 14.42 xIF +
    3.00 x12 + 6.08 x13 + 8.06 x14 + 4.12 x15 + 3.00 x16 + 8.54 x17 + 7.21 x18 + 7.00 x19 + 7.28 x1_10 + 8.60 x1_11 + 10.63 x1_12 + 12.04 x1F +
    3.16 x23 + 5.10 x24 + 1.41 x25 + 4.24 x26 + 5.83 x27 + 6.08 x28 + 7.62 x29 + 7.07 x2_10 + 7.28 x2_11 + 8.60 x2_12 + 10.30 x2F +
    2.00 x34 + 2.83 x35 + 7.21 x36 + 4.47 x37 + 7.28 x38 + 10.00 x39 + 8.94 x3_10 + 8.06 x3_11 + 8.25 x3_12 + 10.20 x3F +
    4.47 x45 + 8.94 x46 + 4.00 x47 + 8.06 x48 + 11.31 x49 + 10.00 x4_10 + 8.54 x4_11 + 8.00 x4_12 + 10.00 x4F +
    4.47 x56 + 4.47 x57 + 5.00 x58 + 7.21 x59 + 6.32 x5_10 + 6.08 x5_11 + 7.21 x5_12 + 8.94 x5F +
    8.00 x67 + 5.00 x68 + 4.00 x69 + 4.47 x6_10 + 6.40 x6_11 + 8.94 x6_12 + 10.00 x6F +
    5.00 x78 + 8.94 x79 + 7.21 x7_10 + 5.00 x7_11 + 4.00 x7_12 + 6.00 x7F +
    4.12 x89 + 2.24 x8_10 + 1.41 x8_11 + 4.12 x8_12 + 5.00 x8F +
    2.00 x9_10 + 5.00 x9_11 + 8.00 x9_12 + 8.25 x9F +
    3.00 x10_11 + 6.00 x10_12 + 6.32 x10F +
    3.00 x11_12 + 3.61 x11F +
    2.00 x12F;

```

Fig.5 : Função objetivo.

2.4. Restrições

Cada vértice pertencente a **V** deve ter uma e somente uma nova aresta ligada a si, de modo a corrigir a paridade dos graus dos vértices pertencentes a **V**.

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

/* Variable bounds */

```

VI:      xI1  + xI2  + xI3  + xI4  + xI5  + xI6  + xI7  + xI8  + xI9  + xI_10 + xI_11 + xI_12 + xIF = 1;
V1: xI1  +      xI2  + xI3  + xI4  + xI5  + xI6  + xI7  + xI8  + xI9  + xI_10 + xI_11 + xI_12 + xIF = 1;
V2: xI2  + xI2  +      xI3  + xI4  + xI5  + xI6  + xI7  + xI8  + xI9  + xI_10 + xI_11 + xI_12 + xIF = 1;
V3: xI3  + xI3  + xI3  +      xI4  + xI5  + xI6  + xI7  + xI8  + xI9  + xI_10 + xI_11 + xI_12 + xIF = 1;
V4: xI4  + xI4  + xI4  + xI4  +      xI5  + xI6  + xI7  + xI8  + xI9  + xI_10 + xI_11 + xI_12 + xIF = 1;
V5: xI5  + xI5  + xI5  + xI5  + xI5  +      xI6  + xI7  + xI8  + xI9  + xI_10 + xI_11 + xI_12 + xIF = 1;
V6: xI6  + xI6  + xI6  + xI6  + xI6  + xI6  +      xI7  + xI8  + xI9  + xI_10 + xI_11 + xI_12 + xIF = 1;
V7: xI7  + xI7  + xI7  + xI7  + xI7  + xI7  + xI7  +      xI8  + xI9  + xI_10 + xI_11 + xI_12 + xIF = 1;
V8: xI8  + xI8  + xI8  + xI8  + xI8  + xI8  + xI8  + xI8  +      xI9  + xI_10 + xI_11 + xI_12 + xIF = 1;
V9: xI9  + xI9  + xI9  + xI9  + xI9  + xI9  + xI9  + xI9  + xI9  +      xI_10 + xI_11 + xI_12 + xIF = 1;
V10: xI_10 + xI_10 + xI_10 + xI_10 + xI_10 + xI_10 + xI_10 + xI_10 + xI_10 + xI_10 +      xI_11 + xI_12 + xIF = 1;
V11: xI_11 + xI_11 + xI_11 + xI_11 + xI_11 + xI_11 + xI_11 + xI_11 + xI_11 + xI_11 + xI_11 +      xI_12 + xIF = 1;
V12: xI_12 + xI_12 + xI_12 + xI_12 + xI_12 + xI_12 + xI_12 + xI_12 + xI_12 + xI_12 + xI_12 + xI_12 +      xIF = 1;
VF: xIF  + xIF  + xIF  + xIF  + xIF  + xIF  + xIF  + xIF  + xIF  + xIF  + xIF  + xIF  + xIF  + xIF = 1;

```

Fig.6 : Restrições.

3. Solução ótima

Ao correr o modelo no programa LPSolve, obtivemos os seguintes resultados.

Source Matrix Options Result			
Objective Constraints Sensitivity			
Variables	MILP Feasible	MILP Better	result
	19.65	19.12	19.12
xI1	1	1	1
x26	0	1	1
x34	1	1	1
x57	0	1	1
x8_11	0	1	1
x9_10	0	1	1
x12F	1	1	1
xI2	0	0	0
xI3	0	0	0
xI4	0	0	0
xI5	0	0	0
xI6	0	0	0
xI7	0	0	0
xI8	0	0	0

Fig.7 : Output obtido do programa LPSolve

De acordo com este output, chegamos a uma solução ótima em que adicionamos ao grafo as arestas xI1, x26, x34, x57, x8_11, x9_10, x12F com um comprimento total de 19,12 unidades. Estas arestas adicionais criam vários possíveis caminhos Eulerianos ótimos que permitem ao drone percorrer todas as linhas de alta tensão começando no vértice **I** e acabando no vértice **F**.

Este seria o aspeto do mapa com as arestas adicionadas:

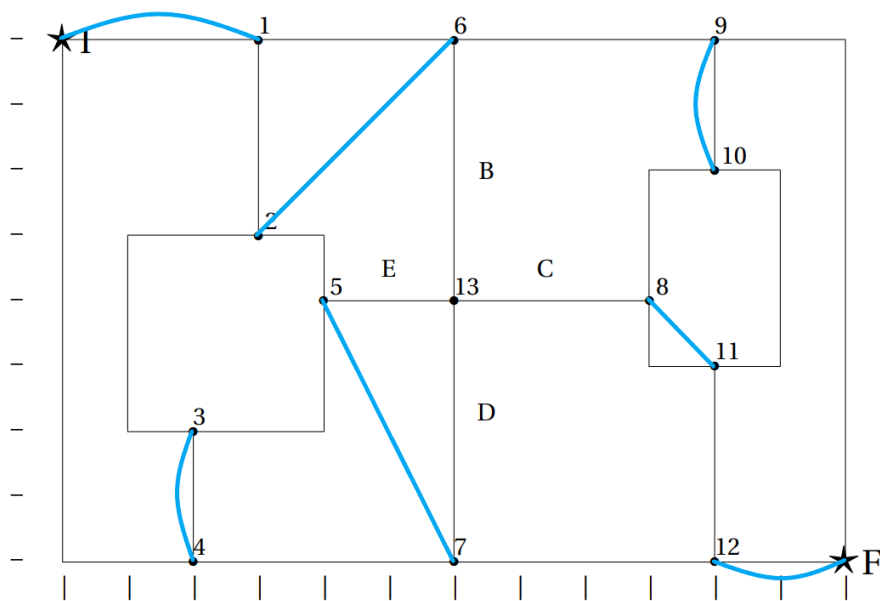


Fig.8 : Mapa com arestas adicionais.

Com este mapa, temos um grafo onde é possível deduzir um possível caminho para o percurso do drone, uma vez que a paridade do grau dos vértices obedece às condições de existência de um caminho Euleriano. Como o caminho terá de passar uma só vez em todas as arestas deste mapa, a distância percorrida pelo drone irá ser sempre a mesma independentemente dos diferentes caminhos Eulerianos possíveis neste mapa. Existem várias possibilidades de caminhos nesta rede de linhas, um exemplo de um desses possíveis caminhos é o seguinte:

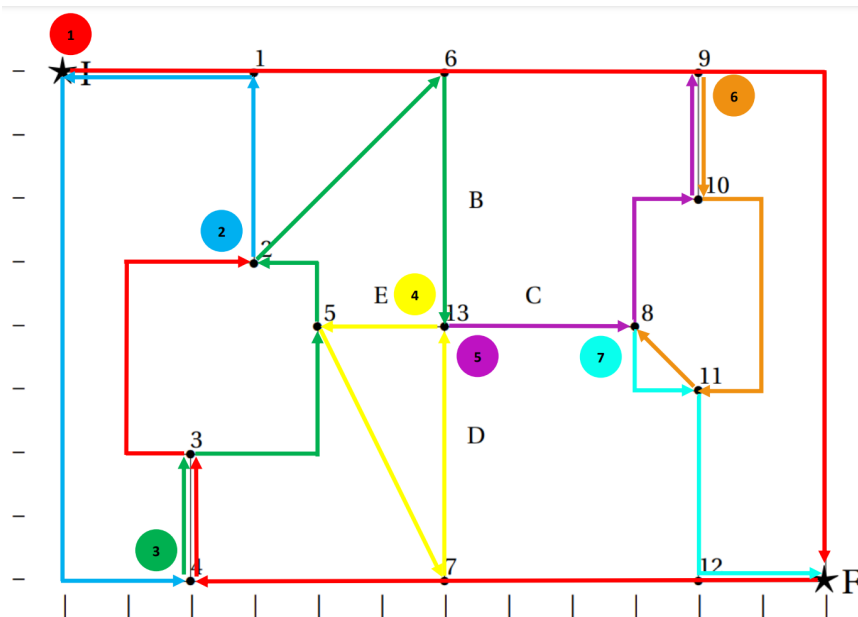

$$I \rightarrow 1 \rightarrow 6 \rightarrow 9 \rightarrow F \rightarrow 12 \rightarrow 7 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow I \rightarrow 4 \rightarrow 3 \rightarrow 5 \rightarrow 2 \rightarrow 6 \rightarrow 13 \rightarrow 5 \rightarrow 7 \rightarrow 13 \rightarrow 8 \rightarrow 10 \rightarrow 9 \rightarrow 10 \rightarrow 11 \rightarrow 8 \rightarrow 11 \rightarrow 12 \rightarrow F$$

Fig.9 : Exemplo de caminho possível.

Neste exemplo do caminho ótimo, o drone percorre em primeiro lugar a linha **vermelha**, depois a **azul**, **verde**, **amarela**, **roxa**, **laranja** e **ciano**. Independentemente do caminho escolhido, o drone terá de percorrer a distância de $85 + 19,12 = 104,12$ unidades, ou seja, a soma do comprimento total das linhas que o drone tem de percorrer (85), com o comprimento total das arestas que adicionámos ao grafo (19,12).

4. Validação do modelo

Para a validação do modelo decidimos desenvolver uma tabela de excel em que cada célula representa o valor de cada variável x_{ij} nas equações das restrições.

Podemos observar a tabela na figura que se segue.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	V	xij	I	1	2	3	4	5	6	7	8	9	10	11	12	F	Arestas adicionadas	Restrição verificada?
2	V1	I		1	0	0	0	0	0	0	0	0	0	0	0	0	1	SIM
3	V1	1	1		0	0	0	0	0	0	0	0	0	0	0	0	1	SIM
4	V2	2	0	0		0	0	0	1	0	0	0	0	0	0	0	1	SIM
5	V3	3	0	0	0		1	0	0	0	0	0	0	0	0	0	1	SIM
6	V4	4	0	0	0	1		0	0	0	0	0	0	0	0	0	1	SIM
7	V5	5	0	0	0	0	0		0	1	0	0	0	0	0	0	1	SIM
8	V6	6	0	0	1	0	0	0		0	0	0	0	0	0	0	1	SIM
9	V7	7	0	0	0	0	0	1	0		0	0	0	0	0	0	1	SIM
10	V8	8	0	0	0	0	0	0	0	0		0	0	1	0	0	1	SIM
11	V9	9	0	0	0	0	0	0	0	0	0		1	0	0	0	1	SIM
12	V10	10	0	0	0	0	0	0	0	0	0	1		0	0	0	1	SIM
13	V11	11	0	0	0	0	0	0	0	1	0	0	0		0	0	1	SIM
14	V12	12	0	0	0	0	0	0	0	0	0	0	0	0		1	1	SIM
15	VF	F	0	0	0	0	0	0	0	0	0	0	0	0	1		1	SIM

Fig.10 : Tabela Excel para validação do modelo.

Nesta tabela, as linhas no intervalo [2,15] representam cada equação das restrições apresentadas anteriormente. Cada célula Q_k (coluna 'Arestas adicionadas') apresenta o somatório dos valores das variáveis presentes na linha k ($k \in [2, 15]$), e cada célula R_k (coluna 'Restrição verificada?') verifica a condição $Q_k = 1$.

Deste modo, conseguimos traduzir as restrições expressas matematicamente para o excel.

Conforme verificado na coluna R da tabela, todas as restrições do modelo foram satisfeitas quando usamos a solução obtida, ou seja, foi adicionada apenas uma aresta a cada vértice.

Ao substituir os valores da solução obtida na função objetivo, obtemos o seguinte:

$$z = 3 \times 1 + 2 \times 1 + 4.24 \times 1 + 4.47 \times 1 + 2 \times 1 + 1.41 \times 1 + 2 \times 1 \Leftrightarrow \\ \Leftrightarrow z = 19.12$$

Este resultado é igual à soma das distâncias euclidianas entre os pontos das arestas obtidas na nossa solução, bem como confirma o resultado obtido no LPSolve.