



**Universidade do Minho**  
Escola de Engenharia

# Investigação Operacional

Relatório da Fase 1 - Drone

Grupo: 93273

20/03/2022

LEI – 2º Ano – 2º Semestre



Carlos Filipe Almeida  
Dias - A93185



José Pedro Martins  
Magalhães - A93273



Francisco Reis  
Izquierdo - A93241



Duarte Augusto  
Rodrigues Lucas -  
A89526

# Conteúdo

1	Introdução .....	3
➤	Objetivo Principal: .....	3
2	Grafo do Grupo: .....	4
3	Formulação do Problema .....	5
➤	Descrição do Problema: .....	5
➤	Objetivo: .....	5
➤	Variáveis de Decisão: .....	6
➤	Parâmetros: .....	6
➤	Função Objetivo: .....	6
➤	Restrições: .....	7
4	Modelo de Programação Linear .....	7
➤	Variáveis de Decisão: .....	7
➤	Parâmetros: .....	8
➤	Função Objetivo: .....	8
➤	Restrições: .....	8
5	LP Solve .....	9
➤	Ficheiro de Input: .....	9
➤	Ficheiro de Output: .....	10
6	Solução Ótima .....	12
7	Análise Final .....	14
➤	Interpretação da Solução Ótima: .....	14
➤	Validação do Modelo: .....	14
➤	Conclusão: .....	15

# Índice de figuras

Figura 1: Distâncias entre Todos os Vértices .....	8
Figura 2: Tabela das distâncias Euclidianas .....	8
Figura 3: Grafo com arestas adicionadas .....	12
Figura 4: Possível percurso .....	13
Figura 5: Resultado das restrições .....	14

# 1 Introdução

## ➤ Objetivo Principal:

No âmbito da disciplina de Investigação Operacional, foi-nos proposto o desenvolvimento de um modelo de programação linear com o intuito de resolver uma problemática dada pela equipa docente. O problema proposto foi estudado e analisado de forma a contemplar os vários passos à resolução do mesmo, de forma a ir ao encontro do que fora pedido pelos docentes. Além disso, toda a modulação do modelo que é detalhado ao longo do presente relatório, foi estruturado e planificado tendo em conta o que foi abordado nas aulas teóricas e práticas, como também os tópicos referenciados nas mesmas e que se mostraram ser fulcrais para a construção do modelo.

Posto isto, o primeiro passo a tomar foi a projeção do grafo presente no enunciado correspondendo ao grupo de trabalho, cujo número é identificado pelo maior número dos elementos do grupo. Este grafo é tido em conta ao previamente mencionado no que toca à remoção das arestas *BCDE*, uma das etapas que é pedida no enunciado, sendo o resultado da mesma visível no tópico abaixo.

Além disso, neste relatório estão detalhados todos os passos relativos à formalização do problema, bem como todos os passos referentes à modelação do problema. Com isto, de forma a puder obter soluções com grande precisão e credíveis, foi usado o software *LPSOLVE*, ferramenta proposta e que é usada nas aulas práticas da disciplina. Por último, é feito um levantamento dos resultados obtidos, isto é o significado da solução obtida.

## 2 Grafo do Grupo:

Tal como foi abordado na introdução, está presente abaixo uma figura representativa do grafo pertencente grupo de trabalho, no qual o maior número dos elementos do grupo corresponde ao número **93273**, no qual dadas as regras impostas da eliminação de arestas, foi apenas retirada a aresta cuja ligação era os vértices 8 e 9, isto é a aresta **C**.

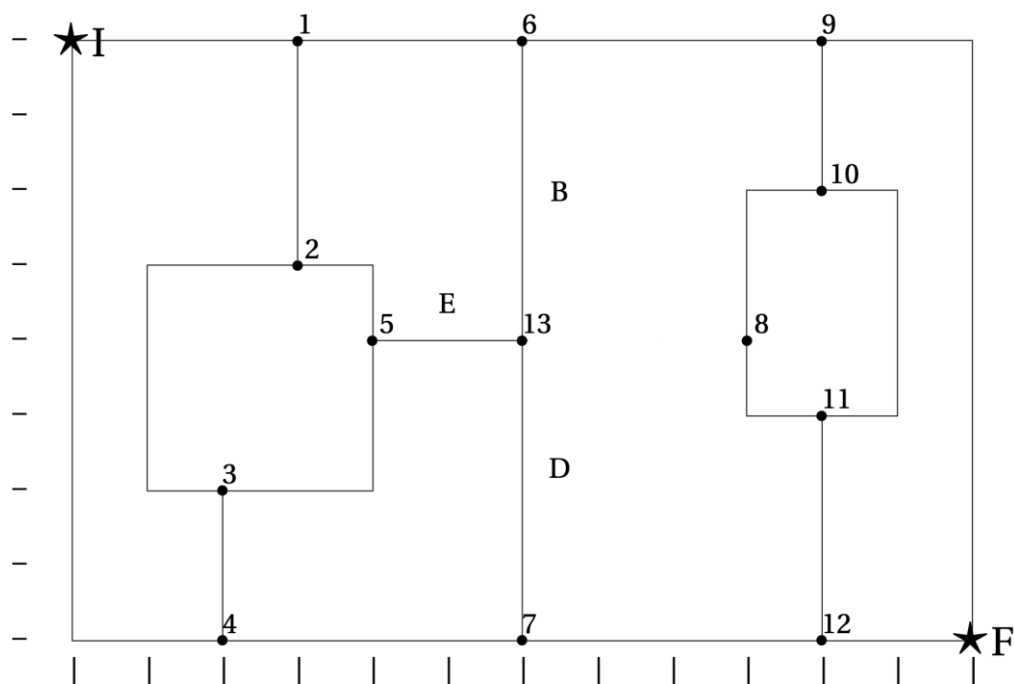


Figura 1: Grafo do grupo (93273)

## 3 Formulação do Problema

### ➤ Descrição do Problema:

O ambiente descrito do problema é o de um drone cuja funcionalidade é a de inspecionar linhas de alta tensão de forma a verificar a presença de vegetação que interfere com as linhas. Além disso, para reposicionar o drone para recomençar a inspeção de uma nova linha não é necessário seguir as linhas bastando fazer o percurso mais curto através do ar. Com isto, o conjunto das linhas de tensão pode ser descrito pelo grafo (arestas do grafo) previamente salientado no qual é de notar a presença de dois postos, um de partida **I** e um de chegada **F**, no qual entre os dois, o drone deve percorrer todas as linhas de tensão, cujas distâncias euclidianas nos são fornecidas.

### ➤ Objetivo:

Dada a descrição do problema, temos como objetivo que o drone deve percorrer um dado percurso de modo a iterar sobre todas as linhas de tensão pelo menos uma vez, indo do ponto de partida até ao ponto de chegada e para tal recorreremos à Teoria dos Grafos, de forma a encontrar uma solução viável, no qual temos o seguinte conceito de circuito Euleriano:

*“Caso um grafo seja ligado e todos os seus vértices tiverem grau par, então garante-se um circuito Euleriano, isto é, que passa apenas uma vez em todas as arestas”.*

Contudo é notório a presença de um grafo não Euleriano, para o qual a única forma de tornar os seus vértices ímpares em vértices pares, é através da inserção de arestas aos mesmo, indo de encontro ao conceito de “ligações aéreas” descritas no tópico anterior. É de extrema consideração que os pontos de partida **I** e de chegada **F**, devem ter grau ímpar uma vez que são pontos distintos, algo relevante na Teoria dos Grafos. Além disso, queremos também minimizar o custo destas ligações adicionais de forma a minimizar a distância percorrida pelo drone.

Assim, dado o descrito, recaímos num problema de minimização, tendo como objetivo encontrar a solução ótima, que consiste em demonstrar que novas arestas devem ser impostas ao grafo de forma a torná-lo Euleriano e que permita minimizar o custo dessas mesmas ligações.

### ➤ Variáveis de Decisão:

Tal como foi abordado previamente, queremos então tornar cada vértice de grau ímpar num vértice de grau par através da inserção de novas arestas (e o contrário nos dois casos de exceção, ponto de partida e ponto de chegada), no entanto é de ressaltar que esta inserção pode resultar numa repetição de aresta, uma vez que queremos o objetivo descrito. Contudo, queremos minimizar o número de ligações e o custo das mesmas, havendo para cada vértice a converter, várias combinações com outros vértices, isto é, novas arestas a formar. Assim, conseguimos identificar o possível emparelhamento entre vértices, ou seja, as possíveis arestas a adicionar, como **as variáveis de decisão**. No que concerne ao nosso grafo, teremos então de ter em conta as várias possibilidades de emparelhamento para cada vértice indo de encontro ao objetivo, no qual o mesmo emparelhamento será representado no formato de  $X_{ij}$ , onde  $i$  representa o vértice que queremos emparelhar e  $j$  representa o vértice admissível de emparelhar. Além disso, estas variáveis de decisão serão tratadas como **binárias**, uma vez que na escolha do melhor emparelhamento, os vértices admissíveis são ou não o melhor emparelhamento para o vértice em estudo, tendo por isso uma conotação binária.

### ➤ Parâmetros:

Postas as variáveis de decisão, de forma a pudermos chegar ao cerne do problema, isto é, a definição do modelo de programação linear adjacente a este problema, teremos também de analisar quais são os parâmetros que irão refletir a informação necessária para a resolução do mesmo. Assim, é de ressaltar que estes parâmetros são os valores associados às distâncias euclidianas propostas aquando do enunciado, uma vez que é a soma das que fazem parte da solução que queremos minimizar, sendo estes valores dados e inalteráveis.

### ➤ Função Objetivo:

Impostas as variáveis de decisão e as restrições, resta agora abordar a função objetivo, no qual dado que temos como objetivo minimizar o custo das arestas a adicionar, esta função será a minimização da soma das distâncias euclidianas associadas às respetivas variáveis de decisão, isto é, às arestas respetivas. Além disso, a função objetivo dará como resultado a solução ótima, sendo esta solução a distância adicional que o drone vai percorrer por forma a atingir o objetivo proposto.

### ➤ Restrições:

Uma vez identificadas as variáveis de decisão, temos também de perceber como as mesmas se irão correlacionar, no intuito de encontrar o melhor emparelhamento. Assim, cada restrição será linear, uma vez que estamos num modelo de programação linear, sendo descrita pela soma dos emparelhamentos admissíveis ao vértice em estudo na mesma restrição igualando o valor de 1, tal como descrito no tópico sobre os parâmetros, uma vez que cada restrição irá admitir apenas o melhor emparelhamento ao vértice e que não são feitos emparelhamentos comuns.

## 4 Modelo de Programação Linear

### ➤ Variáveis de Decisão:

De modo a atingir a resolução pretendida, foi necessário acrescentar ao grafo disponibilizado arestas complementares,  $X_{ij}$ , de modo a torná-lo um grafo *Euleriano*, tal como foi previamente abordado e detalhado.

De modo a cumprir o objetivo supramencionado foi necessário ter em conta outros aspetos, nenhuma aresta poderia ser  $i = j$ , pois não faria sentido criar uma aresta com início e fim iguais, não seria possível repetir arestas já criadas e cada representação de uma possível aresta teria um valor binário, isto é, 0 ou 1.

$X_{ij}$  = Uma aresta que pode ou não existir unindo  $i$  e  $j$ ,  $i, j \in V, i < j$ .

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

$\forall i, j \in \{1,2,3,4,5,6,7,8,9,10,11,12,13, I, F\}$

## ➤ Parâmetros:

Os parâmetros disponibilizados revelam-se inalteráveis, sendo estes a distâncias entre 2 vértices,  $i$  e  $j$ , podendo esta coincidir com as linhas de tensão ou não,  $d_{ij}$ .

		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

Figura 2: Tabela das distâncias Euclidianas

## ➤ Função Objetivo:

A função objetivo é de minimização do comprimento das arestas a adicionar.

$$\min z = \sum_{i=1}^6 \sum_{j=i+1}^7 d_{ij} \times x_{ij} + \sum_{i=1}^7 \sum_{j=9}^{13} d_{ij} \times x_{ij} + \sum_{i=9}^{12} \sum_{j=i+1}^{13} d_{ij} \times x_{ij} + \sum_{i=1}^7 d_{fi} \times x_{fi} + \sum_{i=9}^{13} d_{fi} \times x_{fi} + d_{fi} \times x_{fi} + \sum_{j=1}^7 d_{ij} \times x_{ij} + \sum_{j=9}^{13} d_{ij} \times x_{ij}$$

## ➤ Restrições:

As restrições aplicadas garantem que cada vértice se torne válido, como anteriormente explicado. Deste modo, garantimos também que cada vértice pertence apenas a uma aresta criada.

$$\forall_{ij} \in \{1,2,3,4,5,6,7,8,9,10,11,12,13,I,F\}: \sum x_{ij} = 1, i < j$$



## 5 LP Solve

### ➤ Ficheiro de Input:

```
/* Objective function */
min: 14.42 xfi + 3 xil + 4.24 xi2 + 6.32 xi3 + 8.25 xi4 + 5.66 xi5 + 6 xi6 + 10 xi7 + 10 xi9 + 10.20 xi_10 + 11.18 xi_11 + 12.81 xi_12 + 7.21 xi_13 +
3 x12 + 6.08 x13 + 8.06 x14 + 4.12 x15 + 3 x16 + 8.54 x17 + 7 x19 + 7.28 x1_10 + 8.6 x1_11 + 10.63 x1_12 + 5 x1_13 + 12.04 xfi + 3.16 x23 + 5.1 x24 +
1.41 x25 + 4.24 x26 + 5.83 x27 + 7.62 x29 + 7.07 x2_10 + 7.28 x2_11 + 8.6 x2_12 + 3.16 x2_13 + 10.3 xfi2 + 2 x34 + 2.83 x35 + 7.21 x36 + 4.47 x37
10 x39 + 8.94 x3_10 + 8.06 x3_11 + 8.25 x3_12 + 4.47 x3_13 + 10.20 xfi3 + 4.47 x45 + 8.94 x46 + 4 x47 + 11.31 x49 + 10 x4_10 + 8.54 x4_11 + 8 x4_12 +
5.66 x4_13 + 10 xfi4 + 4.47 x56 + 4.47 x57 + 7.21 x59 + 6.32 x5_10 + 6.08 x5_11 + 7.21 x5_12 + 2 x5_13 + 8.94 xfi5 + 8 x67 + 4 x69 + 4.47 x6_10 +
6.40 x6_11 + 8.94 x6_12 + 4 x6_13 + 10 xfi6 + 8.94 x79 + 7.21 x7_10 + 5 x7_11 + 4 x7_12 + 4 x7_13 + 6 xfi7 + 2 x9_10 + 5 x9_11 + 8 x9_12 + 5.66 x9_13 + 8.25 xfi9
+ 3 x10_11 + 6 x10_12 + 4.47 x10_13 + 6.32 xfi3 + 3 x11_12 + 4.12 x11_13 + 3.61 xfi11 + 5.66 x12_13 +
2 xfi2 + 7.21 xfi3;

/* Restrições */
verticeF : xfi + xfi + xfi2 + xfi3 + xfi4 + xfi5 + xfi6 + xfi7 + xfi9 + xfi_10 + xfi_11 + xfi_12 + xfi_13 = 1;
verticeI : xfi + xil + xi2 + xi3 + xi4 + xi5 + xi6 + xi7 + xi9 + xi_10 + xi_11 + xi_12 + xi_13 = 1;
vertice1 : xfi + xil + xi2 + xi3 + xi4 + xi5 + xi6 + xi7 + xi9 + xi_10 + xi_11 + xi_12 + xi_13 = 1;
vertice2 : xfi2 + xi2 + xi2 + xi23 + xi24 + xi25 + xi26 + xi27 + xi29 + xi2_10 + xi2_11 + xi2_12 + xi2_13 = 1;
vertice3 : xfi3 + xi3 + xi3 + xi23 + xi34 + xi35 + xi36 + xi37 + xi39 + xi3_10 + xi3_11 + xi3_12 + xi3_13 = 1;
vertice4 : xfi4 + xi4 + xi4 + xi24 + xi34 + xi45 + xi46 + xi47 + xi49 + xi4_10 + xi4_11 + xi4_12 + xi4_13 = 1;
vertice5 : xfi5 + xi5 + xi5 + xi25 + xi35 + xi45 + xi56 + xi57 + xi59 + xi5_10 + xi5_11 + xi5_12 + xi5_13 = 1;
vertice6 : xfi6 + xi6 + xi6 + xi26 + xi36 + xi46 + xi56 + xi67 + xi69 + xi6_10 + xi6_11 + xi6_12 + xi6_13 = 1;
vertice7 : xfi7 + xi7 + xi7 + xi27 + xi37 + xi47 + xi57 + xi67 + xi79 + xi7_10 + xi7_11 + xi7_12 + xi7_13 = 1;
vertice9 : xfi9 + xi9 + xi9 + xi29 + xi39 + xi49 + xi59 + xi69 + xi79 + xi9_10 + xi9_11 + xi9_12 + xi9_13 = 1;
vertice10 : xfi_10 + xi_10 + xi_10 + xi2_10 + xi3_10 + xi4_10 + xi5_10 + xi6_10 + xi7_10 + xi9_10 + xi10_11 + xi10_12 + xi10_13 = 1;
vertice11 : xfi_11 + xi_11 + xi_11 + xi2_11 + xi3_11 + xi4_11 + xi5_11 + xi6_11 + xi7_11 + xi9_11 + xi10_11 + xi11_12 + xi11_13 = 1;
vertice12 : xfi_12 + xi_12 + xi_12 + xi2_12 + xi3_12 + xi4_12 + xi5_12 + xi6_12 + xi7_12 + xi9_12 + xi10_12 + xi11_12 + xi12_13 = 1;
vertice13 : xfi_13 + xi_13 + xi_13 + xi2_13 + xi3_13 + xi4_13 + xi5_13 + xi6_13 + xi7_13 + xi9_13 + xi10_13 + xi11_13 + xi12_13 = 1;
```

```
bin xfi, xfi1, xfi2, xfi3, xfi4, xfi5, xfi6, xfi7, xfi9, xfi_10, xfi11, xfi_12, xfi_13;
bin xil, xi2, xi3, xi4, xi5, xi6, xi7, xi9, xi_10, xi_11, xi_12, xi_13;
bin x12, x13, x14, x15, x16, x17, x19, x1_10, x1_11, x1_12, x1_13;
bin x23, x24, x25, x26, x27, x29, x2_10, x2_11, x2_12, x2_13;
bin x34, x35, x36, x37, x39, x3_10, x3_11, x3_12, x3_13;
bin x45, x46, x47, x49, x4_10, x4_11, x4_12, x4_13;
bin x56, x57, x59, x5_10, x5_11, x5_12, x5_13;
bin x67, x69, x6_10, x6_11, x6_12, x6_13;
bin x9_10, x9_11, x9_12, x9_13;
bin x10_11, x10_12, x10_13;
bin x11_12, x11_13;
bin x12_13;
```

➤ Ficheiro de Output:

Variables ▼	MILP ...	result
	16,41	16,41
xi_13	0	0
xi_12	0	0
xi_11	0	0
xi_10	0	0
xi9	0	0
xi7	0	0
xi6	0	0
xi5	0	0
xi4	0	0
xi3	0	0
xi2	0	0
xi1	1	1
xf_13	0	0
xf_12	0	0
xf_11	1	1
xf_10	0	0
xfi	0	0
xf9	0	0
xf7	0	0
xf6	0	0
xf5	0	0
xf4	0	0
xf3	0	0
xf2	0	0
xf13	0	0

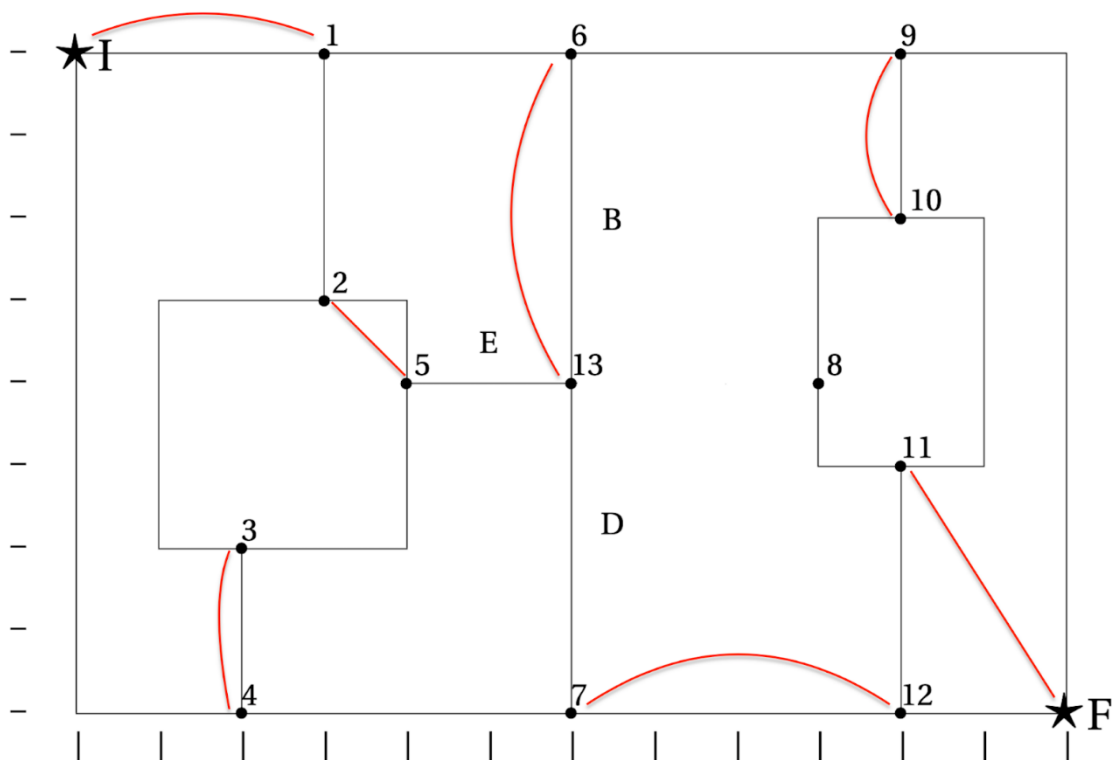
Variables ▼	MILP ...	result
xf12	0	0
xf11	0	0
xf1	0	0
x9_13	0	0
x9_12	0	0
x9_11	0	0
x9_10	1	1
x7_13	0	0
x7_12	1	1
x7_11	0	0
x7_10	0	0
x79	0	0
x6_13	1	1
x6_12	0	0
x6_11	0	0
x6_10	0	0
x69	0	0
x67	0	0
x5_13	0	0
x5_12	0	0
x5_11	0	0
x5_10	0	0
x59	0	0
x57	0	0
x56	0	0
x4_13	0	0

Variables ▼	MILP ...	result
x4_12	0	0
x4_11	0	0
x4_10	0	0
x49	0	0
x47	0	0
x46	0	0
x45	0	0
x3_13	0	0
x3_12	0	0
x3_11	0	0
x3_10	0	0
x39	0	0
x37	0	0
x36	0	0
x35	0	0
x34	1	1
x2_13	0	0
x2_12	0	0
x2_11	0	0
x2_10	0	0
x29	0	0
x27	0	0
x26	0	0
x25	1	1
x24	0	0
x23	0	0
x1_13	0	0

x1_12	0	0
x1_11	0	0
x1_10	0	0
x19	0	0
x17	0	0
x16	0	0
x15	0	0
x14	0	0
x13	0	0
x12_13	0	0
x12	0	0
x11_13	0	0
x11_12	0	0
x10_13	0	0
x10_12	0	0
x10_11	0	0

## 6 Solução Ótima

Respeitando o resultado obtido pelo LP Solve após a introdução de todas as restrições estipuladas e da função objetivo designada pelo grupo de trabalho, sabemos quais as arestas deveremos construir entre vértices de modo a garantir a paridade correta dos mesmos e consequentemente encontrar um percurso ótimo e o seu custo.



*Figura 3: Grafo com arestas adicionadas*

Após a representação das arestas adicionais é possível verificar que os vértices de início e fim são constituídos por um número ímpar de arestas, e os restantes vértices são pares, respeitando todas as normas necessárias para que este seja um grafo Euleriano, sendo possível então percorrer todas as arestas apenas uma vez.

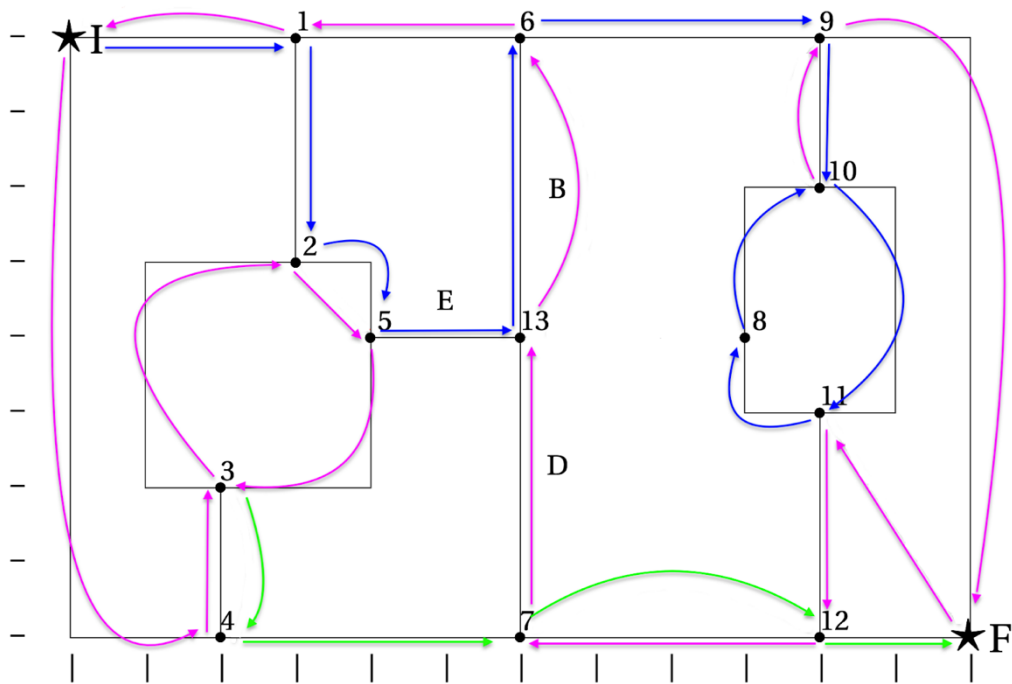


Figura 4: Possível percurso

Neste caso o percurso seria azul → rosa → verde:

I → 1 → 2 → 5 → 13 → 6 → 9 → 10 → 11 → 8 → 10 → 9 → F → 11 → 12 → 7 → 13  
→ 6 → 1 → I → 4 → 3 → 2 → 5 → 3 → 4 → 7 → 12 → F

Deste modo é possível concluir a distância total percorrida:

$$3 + 3 + 2 + 2 + 4 + 4 + 2 + 5 + 2 + 3 + 2 + 10 + 3.61 + 3 + 4 + 4 + 4 + 3 + 3 \\ + 10 + 2 + 6 + 1.41 + 4 + 2 + 4 + 4 + 2 = 102.02$$

## 7 Análise Final

### ➤ Interpretação da Solução Ótima:

Após o levantamento dos resultados propostos pelo LP Solve, verifica-se a presença do valor 1 na escolha das variáveis no que concerne ao emparelhamento dos vértices do grafo, de forma a tornar o mesmo Euleriano. Assim, é notório a escolha de 7 emparelhamentos, formando algumas arestas que se sobrepõem às linhas de tensão, mas com isto, a solução gerada é ótima, no contexto que minimiza a distância do percurso adicional a percorrer pelo drone, para além do caminho que é a constituição inicial do grafo. As variáveis cujo valor é 0, correspondem a variáveis que não foram escolhidas, uma vez que não iriam gerar a solução ótima pretendida.

### ➤ Validação do Modelo:

De modo a validar o modelo proposto, foi necessário verificar os seguintes parâmetros:

- Cada aresta associada por nós não poderá conter vértices repetidos;

Constraints	MILP ...	result
	16,41	16,41
verticeF	1	1
verticeI	1	1
vertice1	1	1
vertice2	1	1
vertice3	1	1
vertice4	1	1
vertice5	1	1
vertice6	1	1
vertice7	1	1
vertice9	1	1
vertice10	1	1
vertice11	1	1
vertice12	1	1
vertice13	1	1

Figura 5: Resultado das restrições

- É possível efetuar o percurso todo e percorrer apenas uma vez cada aresta;
- Os vértices de início e fim terão de ser constituídos por um número ímpar de arestas e os restantes vértices terão de ser constituídos por um número par de arestas;
- Ter a certeza que é uma solução ótima, em vez de possível, isto é, seria possível chegar a outras soluções aumentando o número de arestas e respeitando a paridade de cada vértice, mas nesse caso não seria uma solução ótima;
- O valor da distância percorrida é admissível, isto é, este valor não poderia ser negativo nem nulo, e por outro lado o valor teria de ser maior do que a soma de todas as arestas do grafo.

### ➤ Conclusão:

Este trabalho tinha como objetivo encontrar a solução ótima de um caminho desde um ponto inicial até um ponto final que passe por todas as linhas de alta tensão e minimize a distância percorrida pelo drone. Tal objetivo foi considerado alcançado pela equipa na medida em descobrimos aquela que cremos ser a solução ótima para este problema, uma vez que é aquela que inclui menos variáveis e arestas, maximizando assim a eficiência com base na Teoria de Grafos. Conseguimos também ao longo do projeto ganhar uma compreensão mais aprofundada ao nível da resolução de problemas de minimização assim como um ganho valioso de experiência na formulação de funções objetivo, identificação de restrições e conceção de variáveis de decisão.