



UNIVERSIDADE DO MINHO  
LICENCIATURA EM ENGENHARIA INFORMÁTICA

## **INVESTIGAÇÃO OPERACIONAL**

### **Trabalho Prático 1** **DRONE**



Lídia Sousa  
a93205



Diogo Rebelo  
a93278



Bohdan Malanka  
a93300



Henrique Alvelos  
a93316

20 de julho de 2022

## Conteúdo

<b>1</b>	<b>Introdução</b>	<b>3</b>
<b>2</b>	<b>Questão 0</b>	<b>4</b>
<b>3</b>	<b>Questão 1</b>	<b>5</b>
3.1	Formulação do Problema . . . . .	5
3.1.1	Descrição do Problema . . . . .	5
3.1.2	Objetivo Principal . . . . .	5
3.1.3	Dados . . . . .	5
3.1.4	Variáveis de Decisão . . . . .	5
3.1.5	Restrições . . . . .	6
3.1.6	Ideia subjacente na formulação do problema . . . . .	6
<b>4</b>	<b>Questão 2</b>	<b>7</b>
<b>5</b>	<b>Questão 3</b>	<b>11</b>
<b>6</b>	<b>Questão 4</b>	<b>13</b>
6.1	Ficheiro de Output no Terminal . . . . .	13
6.2	Output na IDE . . . . .	14
<b>7</b>	<b>Questão 5</b>	<b>15</b>
<b>8</b>	<b>Questão 6</b>	<b>17</b>
8.0.1	Validação Analítica . . . . .	17
8.0.2	Validação Intuitiva . . . . .	18
<b>9</b>	<b>Conclusão e Avaliação crítica</b>	<b>19</b>
<b>10</b>	<b>Referências Bibliográficas</b>	<b>19</b>

## Lista de Figuras

1	Mapa de linhas de alta tensão (respetivo grafo inicial) . . . . .	3
2	Rede após remoção de arestas . . . . .	4
3	Ficheiro de Output no Terminal . . . . .	13
4	Ficheiro de Output na IDE . . . . .	14
5	Grafo final com emparelhamentos ótimos . . . . .	15
6	Exemplo de Caminho Euleriano . . . . .	16

# 1 Introdução

O presente relatório incorpora o conjunto de **itens primordiais** do desenvolvimento do primeiro trabalho prático, no âmbito da Unidade Curricular de **Investigação Operacional**. O respetivo enunciado propõe a **formulação de um modelo** de programação linear que permita responder a um problema que envolve a **movimentação de um drone**.

De um modo mais detalhado, o problema é semelhante ao **Problema do Carteiro Chinês (PCC)**, um problema de inspeção de rotas que consiste em encontrar um **caminho mais curto** com a visita de cada aresta de um grafo não direcionado (conectado), pelo menos uma vez.

Analogamente, temos um veículo não tripulado (drone), que tem de se movimentar em linhas de alta tensão, para verificar a existência de vegetação a interferir com essas linhas. O reposicionamento do drone é feito pelo caminho mais curto, de acordo com o que habitualmente se designa por “distância euclidiana”, não sendo necessário seguir as linhas de alta tensão para o efeito. Assim sendo, tem-se como **objetivo principal** que o drone percorra **todas as arestas** do respetivo grafo, pelo menos uma vez, **minimizando a distância total** percorrida. Neste contexto, tem-se bem presente conceito de “**caminho euleriano**”, introduzido por Leonard Euler, já que se visitam todas as arestas pelo menos uma vez, começando num vértice I e terminado num vértice F do grafo (pontos de origem e destino são diferentes).

Para a resolução deste problema, socorre-se à utilização do software de programação linear **LP-Solve**, muito conhecido neste âmbito de programação.

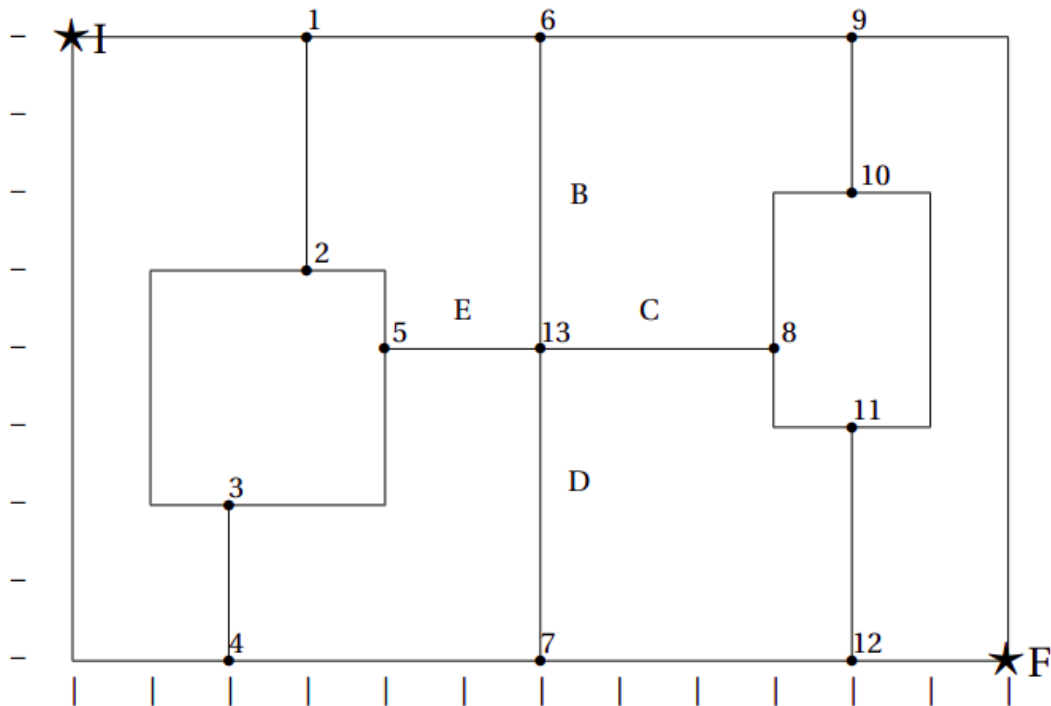


Figura 1: Mapa de linhas de alta tensão (respetivo grafo inicial)



## 3 Questão 1

### 3.1 Formulação do Problema

Uma das etapas mais importantes num problema de programação linear é a formulação do mesmo, já que permite **perceber o problema** de um modo mais profundo, nomeadamente, reunir os **dados**, definir as **variáveis de decisão** relevantes e o **conjunto das restrições**. Parte da secção de Descrição do Problema já surgiu inevitavelmente na introdução, todavia, é relevante incluí-la nesta formulação.

#### 3.1.1 Descrição do Problema

Através do problema proposto, facilmente se identificam **regras gerais de funcionamento e recursos disponíveis**. Como referido anteriormente, trata-se de um problema onde um veículo não tripulado tem de proceder à inspeção de linhas de transporte de energia elétrica de alta tensão, com o intuito de verificar a existência de vegetação que possa interferir com essas linhas. O problema relaciona-se com o conceito de “**caminho euleriano**” (em que os pontos inicial e final são diferentes), já que temos de passar em cada aresta **pelo menos uma vez**.

Assim sendo, é sabido, por definição, que um grafo que contenha um **circuito euleriano** é um **grafo euleriano**. Nestes grafos, o **grau** de cada vértice deve ser **par** e o grafo deve ser **conexo**. Contudo, neste trabalho, pretende-se que o grafo que resulta da adição de arestas seja um grafo com um **caminho euleriano**, ou seja, existem no máximo dois vértices de grau ímpar (o de partida e chegada), sendo os restantes de grau par - como refere o corolário deste Teorema de Euler para Grafos.

Então, temos como princípio na resolução do problema que o grau de incidência de cada vértice deve ser sempre par e que existam, no máximo, dois vértices de grau ímpar (**vértices I e F**), tendo sempre em conta que cada aresta tem uma **distância associada**.

#### 3.1.2 Objetivo Principal

O principal objetivo para a resolução deste problema é encontrar a sua **solução ótima**, ou seja, encontrar o conjunto de arestas suplementares a adicionar, de modo a que se tenha um **caminho euleriano de menor custo**, para, posteriormente, calcular a **distância mínima a percorrer pelo drone**, na sua inspeção das linhas de alta tensão.

#### 3.1.3 Dados

Os dados do problema são facilmente **extraídos do próprio enunciado** e não podem ser alterados. Então, os dados deste modelo de programação linear serão as respetivas **distâncias euclidianas entre os vértices** (surgem representadas na matriz dada) - **comprimento das arestas** (linhas de alta tensão) no próprio mapa de rede (de comprimento inteiro, proporcional à dimensão do seu traço em centímetros). Para além disso, está subentendido como dado a configuração da rede (os próprios vértices e arestas).

#### 3.1.4 Variáveis de Decisão

Estas variáveis de decisão são o **conjunto de valores** que se traduzem em decisões a implementar no sistema real. Como referido na descrição do problema, sabemos que teremos de **tornar todos os vértices de grau ímpar em par**, com **exceção** de dois vértices (I e F), cujo grau deve tornar-se **ímpar**. Para que isso seja possível, procedemos à **adição de arestas** (a determinar) entre os vértices selecionados para ficarem com grau par. Com os vértices I e F, a ideia é a mesma: adicionar arestas para os tornar ímpares. No fundo, acabamos por **criar percursos adicionais** entre estes vértices.

Neste problema, o veículo em questão não é tripulado e, por isso, as ligações a estabelecer podem ser **ligações diretas**, feitas pelo ar através do caminho mais curto entre esses vértices.

Reunindo estas informações e tendo em conta o que se pretende representar, as variáveis de decisão terão de:

- Transparecer o **local de início e fim** do drone, assim como as **arestas** por onde pode passar, **fixas ou adicionais**;
- Definir se um vértice **foi ou não selecionado** para o percurso (no nosso caso, todos são selecionados à exceção do vértice 5, como surge adiante);
- Estar associadas ao **número de vezes** que certos percursos são feitos;

Da lista de itens anterior, conclui-se que é indispensável existirem variáveis de decisão para que seja possível **escolher os vértices que ficarão com grau par/ímpar**. A natureza do problema leva a que se consiga concluir que estas variáveis de decisão serão **variáveis binárias** que irão definir o **estado de um vértice**: se tiverem valor **1 foram selecionadas** para o percurso, caso tenham valor **0 não serão selecionadas** para o percurso. No fundo, permitimos que certos vértices mudem o seu grau conforme desejamos. No nosso caso, obrigamos a que sejam adicionadas arestas a 14 vértices (explicação surge adiante).

### 3.1.5 Restrições

As restrições permitirão **delimitar o espaço de soluções admissíveis**. No nosso caso concreto, as restrições vão permitir definir o conjunto de arestas que podem ser adicionadas a cada vértice, de modo a **alterar o seu grau**, conforme o pretendido. Estas mesmas restrições devem garantir que a escolha das arestas suplementares seja consistente e que o grafo resultante destas adições seja um grafo com um caminho euleriano, ou seja, deve ser uma solução admissível.

Em relação ao problema, são relevantes os tópicos seguintes:

- Não hajam descontinuidades no percurso do drone;
- O drone **passe por todas as arestas** pelo menos uma vez, como dita o próprio problema;
- Dos itens anteriores, conclui-se que terá de existir um **custo de passar por todas as arestas**, denominado de custo fixo. De seguida, precisaremos de ter um custo que represente o **custo dos percursos adicionais**, denominado de custo adicional, e o qual será adicionado ao custo fixo;
- Finalmente, como dita o corolário já supracitado, terão de existir **exatamente dois vértices de grau ímpar** (I e F).

### 3.1.6 Ideia subjacente na formulação do problema

O grafo que o grupo obteve, após remoção de arestas, não é um grafo euleriano, pois temos mais do que dois vértices cujo grau de entrada é ímpar, como prova a tabela seguinte. Então, de modo a resolver o problema, é necessário tornar o grafo euleriano, através do processo de eulerização, formando ligações artificiais entre vértices, de modo a minimizar o comprimento do percurso a efetuar.

Vértices	I	F	1	2	3	4	5	6	7	8	9	10	11	12	13
Grau de Incidência	2	2	3	3	3	3	2	3	3	3	3	3	3	3	3

Tabela 1: Grau de incidência em cada vértice antes da eulerização.

## 4 Questão 2

### Problema do Caminho Euleriano - Modelo

#### Variáveis de Decisão

Para representar cada aresta, utilizamos  $x_{ij}$ , com  $i$  e  $j$  pertencentes a um domínio inteiro e finito, já que representam os vértices de partida e chegada, respectivamente, da aresta em questão. Cada vértice será representado como  $v_t$ , com  $t$  inteiro e positivo e para o caso dos vértices I e F, temos  $v_i$  e  $v_f$ , respectivamente.

Para além disto, é de importância evitar determinadas situações, nomeadamente:

- A existência de uma aresta com  $i = j$ , que não representaria uma aresta;
- A seleção de arestas iguais, com vértices de partida e chegada invertidos. Por exemplo, as variáveis  $x_{12}$  e  $x_{21}$ , que representam uma mesma aresta. Para contornar isso, utiliza-se  $i < j$ , na representação de arestas.

Assim sendo, seguem-se as declarações do tipo variável:

**$x_{ij}$ :  $i$  representa o vértice de origem,  $j$  representa o vértice de destino. Permite representar uma aresta  $ij$  (variável binária).**

$$i, j \in \{1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13\} \cup \{i, f\}$$

Não foi incluído no conjunto anterior o vértice 5, uma vez que este já possuía grau par. A reunião com os elementos  $\{i, f\}$  serve apenas para destacar o caso das arestas com os vértices I e F (por exemplo:  $x_{i1}, x_{1f}$ )

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

- Para o caso de arestas com vértices que não os de origem e destino:

$$\forall i, j \in \{1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13\}, i \neq j \wedge i < j$$

- Para o caso de arestas com vértices de origem e/ou destino:

- Se a aresta envolve o vértice I:

$$\forall x_{ij}, (i = i) \wedge j \in \{1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, f\}$$

- Se a aresta envolve o vértice F:

$$\forall x_{ij}, (j = f) \wedge i \in \{i, 1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13\}$$

**$v_t$ :  $t$  representa o vértice em questão (variável binária).**

$$t \in \{1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13\} \cup \{i, f\}$$

$$v_t \in \{0, 1\}$$

## Restrições

Em relação ao conjunto de restrições, é coerente e relevante afirmar que se dividem em dois grupos distintos.

### Grupo 1

Grupo de restrições (implícito) necessário para que seja possível escolher quais os vértices que serão convertidos a vértices de grau par ou a vértices de grau ímpar.

$$v_i \leq 1; v_1 \leq 1; v_2 \leq 1; v_3 \leq 1; v_4 \leq 1; v_6 \leq 1; \\ v_7 \leq 1; v_8 \leq 1; v_9 \leq 1; v_{10} \leq 1; v_{11} \leq 1; v_{12} \leq 1; v_{13} \leq 1; v_f \leq 1;$$

Estas primeiras inequações permitem que se possa selecionar ou não os vértices para o percurso. Como abaixo estamos a incluir todos os vértices à exceção dos vértices pares que não são pontos de início ou fim (sendo estes variáveis binárias) estamos a obrigar que todos sejam iguais a 1 (número de variáveis no 1º membro é igual a 14). Logo, a restrições de menor ou igual acima representadas deixam de ser necessárias, podendo ser colocadas em comentário.

$$v_i + v_1 + v_2 + v_3 + v_4 + v_6 + v_7 + v_8 + v_9 + v_{10} + v_{11} + v_{12} + v_{13} + v_f = 14$$

Esta equação permite estabelecer que dos vértices presentes no primeiro membro da equação, queremos que sejam convertidos em vértices de grau par/ímpar.

### Grupo 2

Grupo de restrições que estabelecem o conjunto de arestas possíveis a adicionar a um determinado vértice.

#### Vértice Inicial I

$$x_{i1} + x_{i2} + x_{i3} + x_{i4} + x_{i6} + x_{i7} + x_{i8} + x_{i9} + x_{i10} + x_{i11} + x_{i12} + x_{i13} + x_{if} = v_i;$$

#### Vértice Final F

$$x_{if} + x_{1f} + x_{2f} + x_{3f} + x_{4f} + x_{6f} + x_{7f} + x_{8f} + x_{9f} + x_{10f} + x_{11f} + x_{12f} + x_{13f} = v_f;$$

#### Vértice 1

$$x_{12} + x_{13} + x_{14} + x_{16} + x_{17} + x_{18} + x_{19} + x_{110} + x_{111} + x_{112} + x_{113} + x_{1f} + x_{i1} = v_1;$$

#### Vértice 2

$$x_{12} + x_{23} + x_{24} + x_{26} + x_{27} + x_{28} + x_{29} + x_{210} + x_{211} + x_{212} + x_{213} + x_{2f} + x_{i2} = v_2;$$

#### Vértice 3

$$x_{13} + x_{23} + x_{34} + x_{36} + x_{37} + x_{38} + x_{39} + x_{310} + x_{311} + x_{312} + x_{313} + x_{3f} + x_{i3} = v_3;$$

#### Vértice 4

$$x_{14} + x_{24} + x_{34} + x_{46} + x_{47} + x_{48} + x_{49} + x_{410} + x_{411} + x_{412} + x_{413} + x_{4f} + x_{i4} = v_4;$$

#### Vértice 6

$$x_{16} + x_{26} + x_{36} + x_{46} + x_{67} + x_{68} + x_{69} + x_{610} + x_{611} + x_{612} + x_{613} + x_{6f} + x_{i6} = v_6;$$

#### Vértice 7

$$x_{17} + x_{27} + x_{37} + x_{47} + x_{67} + x_{78} + x_{79} + x_{710} + x_{711} + x_{712} + x_{713} + x_{7f} + x_{i7} = v_7;$$

#### Vértice 8

$$x_{18} + x_{28} + x_{38} + x_{48} + x_{68} + x_{78} + x_{89} + x_{810} + x_{811} + x_{812} + x_{813} + x_{8f} + x_{i8} = v_8;$$



**Vértice 9**

$$x_{19} + x_{29} + x_{39} + x_{49} + x_{69} + x_{79} + x_{89} + x_{910} + x_{911} + x_{912} + x_{913} + x_{9f} + x_{i9} = v_9;$$

**Vértice 10**

$$x_{110} + x_{210} + x_{310} + x_{410} + x_{610} + x_{710} + x_{810} + x_{910} + x_{1011} + x_{1012} + x_{1013} + x_{10f} + x_{i10} = v_{10};$$

**Vértice 11**

$$x_{111} + x_{211} + x_{311} + x_{411} + x_{611} + x_{711} + x_{811} + x_{911} + x_{1011} + x_{1112} + x_{1113} + x_{11f} + x_{i11} = v_{11};$$

**Vértice 12**

$$x_{112} + x_{212} + x_{312} + x_{412} + x_{612} + x_{712} + x_{812} + x_{912} + x_{1012} + x_{1112} + x_{1213} + x_{12f} + x_{i12} = v_{12};$$

**Vértice 13**

$$x_{113} + x_{213} + x_{313} + x_{413} + x_{613} + x_{713} + x_{813} + x_{913} + x_{1013} + x_{1113} + x_{1213} + x_{13f} + x_{i13} = v_{13};$$

**Declaração de Variáveis Binárias**

Finalmente, temos ainda de declarar as variáveis como binárias, acabando por restringi-las a 0 ou 1:

**bin**

$$\begin{aligned} & x_{i1} \ x_{i2} \ x_{i3} \ x_{i4} \ x_{i6} \ x_{i7} \ x_{i8} \ x_{i9} \ x_{i10} \ x_{i11} \ x_{i12} \ x_{i13} \ x_{if} \\ & x_{12} \ x_{13} \ x_{14} \ x_{16} \ x_{17} \ x_{18} \ x_{19} \ x_{110} \ x_{111} \ x_{112} \ x_{113} \ x_{1f} \\ & x_{23} \ x_{24} \ x_{26} \ x_{27} \ x_{28} \ x_{29} \ x_{210} \ x_{211} \ x_{212} \ x_{213} \ x_{2f} \\ & x_{34} \ x_{36} \ x_{37} \ x_{38} \ x_{39} \ x_{310} \ x_{311} \ x_{312} \ x_{313} \ x_{3f} \\ & x_{46} \ x_{47} \ x_{48} \ x_{49} \ x_{410} \ x_{411} \ x_{412} \ x_{413} \ x_{4f} \\ & x_{67} \ x_{68} \ x_{69} \ x_{610} \ x_{611} \ x_{612} \ x_{613} \ x_{6f} \\ & x_{78} \ x_{79} \ x_{710} \ x_{711} \ x_{712} \ x_{713} \ x_{7f} \\ & x_{89} \ x_{810} \ x_{811} \ x_{812} \ x_{813} \ x_{8f} \\ & x_{910} \ x_{911} \ x_{912} \ x_{913} \ x_{9f} \\ & x_{1011} \ x_{1012} \ x_{1013} \ x_{10f} \\ & x_{1112} \ x_{1113} \ x_{11f} \\ & x_{1213} \ x_{12f} \\ & x_{13f} \\ & v_i \ v_1 \ v_2 \ v_3 \ v_4 \ v_6 \ v_7 \ v_8 \ v_9 \ v_{10} \ v_{11} \ v_{12} \ v_{13} \ v_f; \end{aligned}$$


---

## Função Objetivo

Não esquecendo nunca o objetivo do problema - determinar as arestas a adicionar para obter a distância euclidiana de menor custo, para que o drone consiga verificar a possível existência de vegetação a interferir com as linhas de alta tensão -, torna-se fulcral que na função objetivo estejam as distâncias das arestas fixas do mapa inicial, 83 centímetros, assim como a soma das distâncias das arestas adicionais possíveis multiplicadas pela sua respectiva variável binária.

A função objetivo é:

**min:** 83 + 3.00 xi1 + 4.24 xi2 + 6.32 xi3 + 8.25 xi4 + 6.00 xi6 + 10.00 xi7 + 9.85 xi8 + 10.00 xi9 + 10.20 xi10 + 11.18 xi11 + 12.81 xi12 + 7.21 xi13 + 14.42 xif + 3.00 x12 + 6.08 x13 + 8.06 x14 + 3.00 x16 + 8.54 x17 + 7.21 x18 + 7.00 x19 + 7.28 x110 + 8.60 x111 + 10.63 x112 + 5.00 x113 + 12.04 x1f + 3.16 x23 + 5.10 x24 + 4.24 x26 + 5.83 x27 + 6.08 x28 + 7.62 x29 + 7.07 x210 + 7.28 x211 + 8.60 x212 + 3.16 x213 + 10.30 x2f + 2.00 x34 + 7.21 x36 + 4.47 x37 + 7.28 x38 + 10.00 x39 + 8.94 x310 + 8.06 x311 + 8.25 x312 + 4.47 x313 + 10.20 x3f + 8.94 x46 + 4.00 x47 + 8.06 x48 + 11.31 x49 + 10.00 x410 + 8.54 x411 + 8.00 x412 + 5.66 x413 + 10.00 x4f + 8.00 x67 + 5.00 x68 + 4.00 x69 + 4.47 x610 + 6.40 x611 + 8.94 x612 + 4.00 x613 + 10.00 x6f + 5.00 x78 + 8.94 x79 + 7.21 x710 + 5.00 x711 + 4.00 x712 + 4.00 x713 + 6.00 x7f + 4.12 x89 + 2.24 x810 + 1.41 x811 + 4.12 x812 + 3.00 x813 + 5.00 x8f + 2.00 x910 + 5.00 x911 + 8.00 x912 + 5.66 x913 + 8.25 x9f + 3.00 x1011 + 6.00 x1012 + 4.47 x1013 + 6.32 x10f + 3.00 x1112 + 4.12 x1113 + 3.61 x11f + 5.66 x1213 + 2.00 x12f + 7.21 x13f;

Em relação ao custo fixo, ou seja, a soma do comprimento de todas as arestas iniciais, tem-se:

**custo fixo** =  $3 \times 8 + 2 \times 12 + 3 \times 4 + 3 \times 2 + 4 \times 2 + 3 \times 3 = 83$

## 5 Questão 3

Como explícito na Introdução do trabalho, depois de percebermos o problema e de o formularmos corretamente através de programação linear, procedemos ao software *LPSolve*, para encontrar a respetiva solução ótima no contexto do problema. Então, colocamos a função objetivo a minimizar e as restrições que anteriormente definimos. Conseguimos construir, assim, o ficheiro de input que apresentamos a seguir:

```
/* Objective function */
min: 83 + 3.00 x11 + 4.24 x12 + 6.32 x13 + 8.25 x14 + 6.00 x16 + 10.00 x17
      + 9.85 x18 + 10.00 x19 + 10.20 x110 + 11.18 x111 + 12.81 x112
      + 7.21 x113 + 14.42 x1f + 3.00 x12 + 6.08 x13 + 8.06 x14 + 3.00 x16
      + 8.54 x17 + 7.21 x18 + 7.00 x19 + 7.28 x110 + 8.60 x111 + 10.63 x112
      + 5.00 x113 + 12.04 x1f + 3.16 x23 + 5.10 x24 + 4.24 x26 + 5.83 x27
      + 6.08 x28 + 7.62 x29 + 7.07 x210 + 7.28 x211 + 8.60 x212 + 3.16 x213
      + 10.30 x2f + 2.00 x34 + 7.21 x36 + 4.47 x37 + 7.28 x38 + 10.00 x39
      + 8.94 x310 + 8.06 x311 + 8.25 x312 + 4.47 x313 + 10.20 x3f + 8.94 x46
      + 4.00 x47 + 8.06 x48 + 11.31 x49 + 10.00 x410 + 8.54 x411 + 8.00 x412
      + 5.66 x413 + 10.00 x4f + 8.00 x67 + 5.00 x68 + 4.00 x69 + 4.47 x610
      + 6.40 x611 + 8.94 x612 + 4.00 x613 + 10.00 x6f + 5.00 x78 + 8.94 x79
      + 7.21 x710 + 5.00 x711 + 4.00 x712 + 4.00 x713 + 6.00 x7f + 4.12 x89
      + 2.24 x810 + 1.41 x811 + 4.12 x812 + 3.00 x813 + 5.00 x8f
      + 2.00 x910 + 5.00 x911 + 8.00 x912 + 5.66 x913 + 8.25 x9f
      + 3.00 x1011 + 6.00 x1012 + 4.47 x1013 + 6.32 x10f
      + 3.00 x1112 + 4.12 x1113 + 3.61 x11f
      + 5.66 x1312 + 2.00 x12f
      + 7.21 x13f;

/* Variable bounds */
// Group 1
/* Implicit
vi<=1;
v1<=1;
v2<=1;
v3<=1;
v4<=1;
v6<=1;
v7<=1;
v8<=1;
v9<=1;
v10<=1;
v11<=1;
v12<=1;
v13<=1;
vf<=1; */

vi+v1+v2+v3+v4+v6+v7+v8+v9+v10+v11+v12+v13+vf = 14;

// Group 2
x11+x12+x13+x14+x16+x17+x18+x19+x110+x111+x112+x113+x1f=vi;
x1f+x1f+x2f+x3f+x4f+x6f+x7f+x8f+x9f+x10f+x11f+x12f+x13f=vf;
x12+x13+x14+x16+x17+x18+x19+x110+x111+x112+x113+x1f+x11=v1;
x12+x23+x24+x26+x27+x28+x29+x210+x211+x212+x213+x2f+x12=v2;
x13+x23+x34+x36+x37+x38+x39+x310+x311+x312+x313+x3f+x13=v3;
x14+x24+x34+x46+x47+x48+x49+x410+x411+x412+x413+x4f+x14=v4;
x16+x26+x36+x46+x67+x68+x69+x610+x611+x612+x613+x6f+x16=v6;
x17+x27+x37+x47+x67+x78+x79+x710+x711+x712+x713+x7f+x17=v7;
```

```

x18+x28+x38+x48+x68+x78+x89+x810+x811+x812+x813+x8f+xi8=v8;
x19+x29+x39+x49+x69+x79+x89+x910+x911+x912+x913+x9f+xi9=v9;
x110+x210+x310+x410+x610+x710+x810+x910+x1011+x1012+x1013+x10f+xi10=v10;
x111+x211+x311+x411+x611+x711+x811+x911+x1011+x1112+x1113+x11f+xi11=v11;
x112+x212+x312+x412+x612+x712+x812+x912+x1012+x1112+x1213+x12f+xi12=v12;
x113+x213+x313+x413+x613+x713+x813+x913+x1013+x1113+x1213+x13f+xi13=v13;

```

```

/* Binary Variables Declaration */

```

```

bin

```

```

    xi1 xi2 xi3 xi4 xi6 xi7 xi8 xi9 xi10 xi11 xi12 xi13 xif
      x12 x13 x14 x16 x17 x18 x19 x110 x111 x112 x113 x1f
        x23 x24 x26 x27 x28 x29 x210 x211 x212 x213 x2f
          x34 x36 x37 x38 x39 x310 x311 x312 x313 x3f
            x46 x47 x48 x49 x410 x411 x412 x413 x4f
              x67 x68 x69 x610 x611 x612 x613 x6f
                x78 x79 x710 x711 x712 x713 x7f
                  x89 x810 x811 x812 x813 x8f
                    x910 x911 x912 x913 x9f
                      x1011 x1012 x1013 x10f
                        x1112 x1113 x11f
                          x1213 x12f
                            x13f
vi vi v1 v2 v3 v4 v6 v7 v8 v9 v10 v11 v12 v13 vf;

```

## 6 Questão 4

### 6.1 Ficheiro de Output no Terminal



```
C:\Program Files (x86)\LPSolve IDE\LpSolveIDE.exe
Model name: 'LPSolver' - run #1
Objective: Minimize(R0)

SUBMITTED
Model size:      15 constraints,      105 variables,      210 non-zeros.
Sets:            0 GUB,              0 SOS.

Using DUAL simplex for phase 1 and PRIMAL simplex for phase 2.
The primal and dual simplex pricing strategy set to 'Devex'.

Relaxed solution      101.65 after      30 iter is B&B base.
Feasible solution     101.65 after      30 iter,      0 nodes (gap 0.0%)
Optimal solution      101.65 after      30 iter,      0 nodes (gap 0.0%).

Relative numeric accuracy ||*|| = 5.55112e-017

MEMO: lp_solve version 5.5.2.11 for 32 bit OS, with 64 bit REAL variables.
In the total iteration count 30, 15 (50.0%) were bound flips.
There were 0 refactorizations, 0 triggered by time and 0 by density.
... on average 15.0 major pivots per refactorization.
The largest [LUSOL v2.2.1.0] fact(B) had 16 NZ entries, 1.0x largest basis.
The maximum B&B level was 1, 0.0x MIP order, 1 at the optimal solution.
The constraint matrix inf-norm is 1, with a dynamic range of 1.
Time to load data was 0.002 seconds, presolve used 0.005 seconds,
... 0.013 seconds in simplex solver, in total 0.020 seconds.
```

Figura 3: Ficheiro de Output no Terminal

## 6.2 Output na IDE

Como referido ao longo do trabalho, após a execução do input por nós construído, a ferramenta apresenta-nos a solução ótima para o nosso problema de programação linear, dizendo-nos quais as ligações adicionais de menor distância que devemos considerar no percurso do drone e tendo um caminho euleriano por si percorrido.

Variables	MILP ...	result
x1	1	1
x2	0	0
x3	0	0
x4	0	0
x6	0	0
x7	0	0
x8	0	0
x9	0	0
x10	0	0
x11	0	0
x12	0	0
x13	0	0
xf	0	0
x12	0	0
x13	0	0
x14	0	0
x16	0	0
x17	0	0
x18	0	0
x19	0	0
x110	0	0
x111	0	0

Variables	MILP ...	result
x112	0	0
x113	0	0
x1f	0	0
x23	0	0
x24	0	0
x26	1	1
x27	0	0
x28	0	0
x29	0	0
x210	0	0
x211	0	0
x212	0	0
x213	0	0
x2f	0	0
x34	1	1
x36	0	0
x37	0	0
x38	0	0
x39	0	0
x310	0	0
x311	0	0
x312	0	0
x313	0	0

Variables	MILP ...	result
x3f	0	0
x46	0	0
x47	0	0
x48	0	0
x49	0	0
x410	0	0
x411	0	0
x412	0	0
x413	0	0
x4f	0	0
x67	0	0
x68	0	0
x69	0	0
x610	0	0
x611	0	0
x612	0	0
x613	0	0
x6f	0	0
x78	0	0
x79	0	0
x710	0	0
x711	0	0
x712	0	0

Variables	MILP ...	result
x713	1	1
x7f	0	0
x89	0	0
x810	0	0
x811	1	1
x812	0	0
x813	0	0
x8f	0	0
x910	1	1
x911	0	0
x912	0	0
x913	0	0
x9f	0	0
x1011	0	0
x1012	0	0
x1013	0	0
x10f	0	0
x1112	0	0
x1113	0	0
x11f	0	0
x1213	0	0
x12f	1	1
x13f	0	0

vi	result
v1	1
v2	1
v3	1
v4	1
v6	1
v7	1
v8	1
v9	1
v10	1
v11	1
v12	1
v13	1
vf	1

Figura 4: Ficheiro de Output na IDE

## 7 Questão 5

### Análise de Resultados

Um objetivo implícito era fazer com que o *LPSolve* nos fornecesse o conjunto de arestas a adicionar à rede inicial de modo a obter um caminho euleriano de menor custo. Assim sendo, interpretando a solução obtida, é correto afirmar que a solução ótima é constituída por todos os vértices à exceção do vértice 5. Para além disso, é também constituída por vários emparelhamentos, nomeadamente:

- Vértice I ao 1 - comprimento: 3.00;
- Vértice 2 ao 6 - comprimento: 4.24;
- Vértice 3 ao 4 - comprimento: 2.00;
- Vértice 7 ao 13 - comprimento: 4.00;
- Vértice 8 ao 11 - comprimento: 1.41;
- Vértice 9 ao 10 - comprimento: 2.00;
- Vértice 12 ao F - comprimento: 2.00;
- Total = 18.65.

Através da matriz fornecida, sabe-se a distância euclidiana de cada emparelhamento. Na alínea anterior, obteve-se a solução ótima que corresponde à distância (em centímetros) percorrida pelo drone em todo o seu percurso, ao longo da verificação de todas as linhas de alta tensão. Logo, o drone percorre 101.65 centímetros. Este é claramente o percurso que gasta menos recursos (que percorre a menor distância total, tendo em conta todas as restrições em termos de caminho). Através do somatório das distância euclidianas dos emparelhamentos anteriores, obtemos exatamente o valor 18.65, o qual traduz o gasto mínimo de recursos.

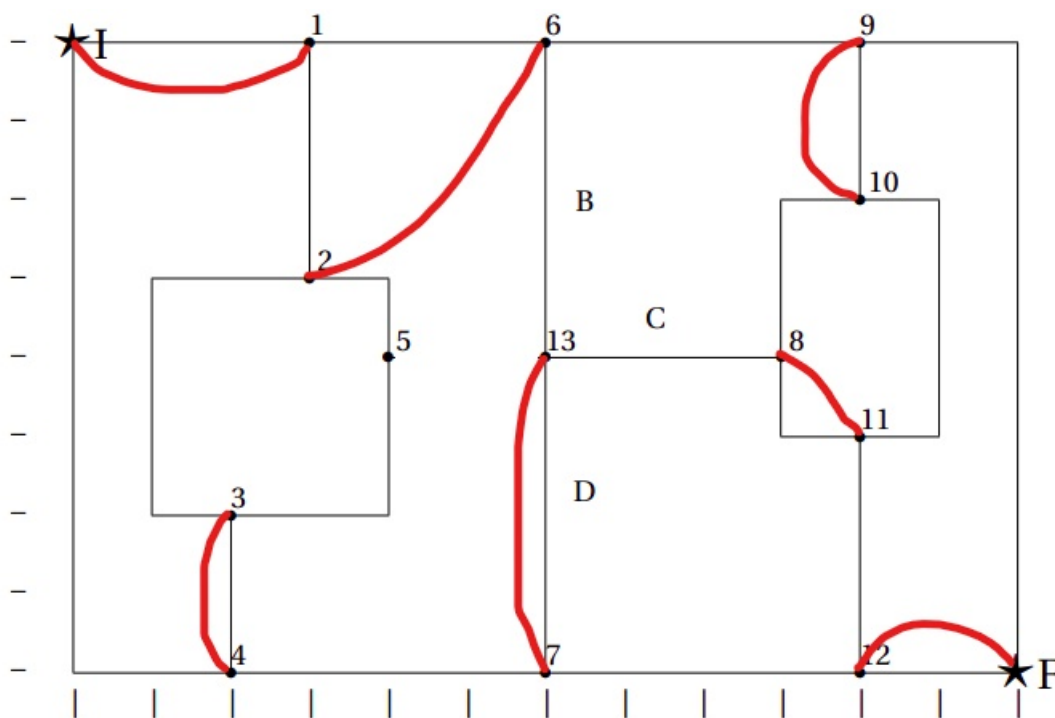
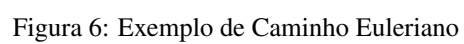


Figura 5: Grafo final com emparelhamentos ótimos

Através desta solução ótima, podemos tentar encontrar um caminho euleriano (possível), deixando abaixo todo o percurso a seguir pelo drone, como forma de exemplo:





## 8 Questão 6

### Validação do Modelo

Para validar este modelo, o grupo decidiu proceder de duas abordagens diferentes, uma mais intuitiva e outra mais analítica.

#### 8.0.1 Validação Analítica

Para o fazermos, temos de analisar a solução que o software utilizado nos forneceu, efetuar a substituição dos valores de cada variável na função objetivo e nas restrições e verificar se é coerente e válido o seu resultado.

A função objetivo é:

**min:**  $83 + 3.00 x_{i1} + 4.24 x_{i2} + 6.32 x_{i3} + 8.25 x_{i4} + 6.00 x_{i6} + 10.00 x_{i7} + 9.85 x_{i8} + 10.00 x_{i9} + 10.20 x_{i10} + 11.18 x_{i11} + 12.81 x_{i12} + 7.21 x_{i13} + 14.42 x_{if} + 3.00 x_{i2} + 6.08 x_{i3} + 8.06 x_{i4} + 3.00 x_{i6} + 8.54 x_{i7} + 7.21 x_{i8} + 7.00 x_{i9} + 7.28 x_{i10} + 8.60 x_{i11} + 10.63 x_{i12} + 5.00 x_{i13} + 12.04 x_{if} + 3.16 x_{i23} + 5.10 x_{i24} + 4.24 x_{i26} + 5.83 x_{i27} + 6.08 x_{i28} + 7.62 x_{i29} + 7.07 x_{i210} + 7.28 x_{i211} + 8.60 x_{i212} + 3.16 x_{i213} + 10.30 x_{i2f} + 2.00 x_{i34} + 7.21 x_{i36} + 4.47 x_{i37} + 7.28 x_{i38} + 10.00 x_{i39} + 8.94 x_{i310} + 8.06 x_{i311} + 8.25 x_{i312} + 4.47 x_{i313} + 10.20 x_{i3f} + 8.94 x_{i46} + 4.00 x_{i47} + 8.06 x_{i48} + 11.31 x_{i49} + 10.00 x_{i410} + 8.54 x_{i411} + 8.00 x_{i412} + 5.66 x_{i413} + 10.00 x_{i4f} + 8.00 x_{i67} + 5.00 x_{i68} + 4.00 x_{i69} + 4.47 x_{i610} + 6.40 x_{i611} + 8.94 x_{i612} + 4.00 x_{i613} + 10.00 x_{i6f} + 5.00 x_{i78} + 8.94 x_{i79} + 7.21 x_{i710} + 5.00 x_{i711} + 4.00 x_{i712} + 4.00 x_{i713} + 6.00 x_{i7f} + 4.12 x_{i89} + 2.24 x_{i810} + 1.41 x_{i811} + 4.12 x_{i812} + 3.00 x_{i813} + 5.00 x_{i8f} + 2.00 x_{i910} + 5.00 x_{i911} + 8.00 x_{i912} + 5.66 x_{i913} + 8.25 x_{i9f} + 3.00 x_{i1011} + 6.00 x_{i1012} + 4.47 x_{i1013} + 6.32 x_{i10f} + 3.00 x_{i1112} + 4.12 x_{i1113} + 3.61 x_{i11f} + 5.66 x_{i1213} + 2.00 x_{i12f} + 7.21 x_{i13f};$

Efetuando a substituição dos valores dados pelos LPSolver, vem:

**min:**  $83 + 3.00 + 4.24 + 2.00 + 4.00 + 1.41 + 2.00 + 2.00 = 101.65$

Confrontando com as respectivas restrições, temos:

$v_i + v_l + v_2 + v_3 + v_4 + v_6 + v_7 + v_8 + v_9 + v_{10} + v_{11} + v_{12} + v_{13} + v_f = 14 \Leftrightarrow$   
 $\Leftrightarrow 1+1+1+1+1+1+1+1+1+1+1+1+1 = 14 \Leftrightarrow \text{True}$   
 $x_{i1} + x_{i2} + x_{i3} + x_{i4} + x_{i6} + x_{i7} + x_{i8} + x_{i9} + x_{i10} + x_{i11} + x_{i12} + x_{i13} + x_{if} = v_i \Leftrightarrow$   
 $\Leftrightarrow 1+0+0+0+0+0+0+0+0+0+0+0+0 = 1 \Leftrightarrow \text{True}$   
 $x_{if} + x_{i1f} + x_{i2f} + x_{i3f} + x_{i4f} + x_{i6f} + x_{i7f} + x_{i8f} + x_{i9f} + x_{i10f} + x_{i11f} + x_{i12f} + x_{i13f} = v_f \Leftrightarrow$   
 $\Leftrightarrow 0+0+0+0+0+0+0+0+0+0+0+0+1 = 1 \Leftrightarrow \text{True}$   
 $x_{i2} + x_{i3} + x_{i4} + x_{i6} + x_{i7} + x_{i8} + x_{i9} + x_{i10} + x_{i11} + x_{i12} + x_{i13} + x_{if} + x_{i1} = v_1 \Leftrightarrow$   
 $\Leftrightarrow 0+0+0+0+0+0+0+0+0+0+0+0+1 = 1 \Leftrightarrow \text{True}$   
 $x_{i2} + x_{i3} + x_{i4} + x_{i6} + x_{i7} + x_{i8} + x_{i9} + x_{i10} + x_{i11} + x_{i12} + x_{i13} + x_{if} + x_{i2} = v_2 \Leftrightarrow$   
 $\Leftrightarrow 0+0+0+1+0+0+0+0+0+0+0+0+0 = 1 \Leftrightarrow \text{True}$   
 $x_{i3} + x_{i4} + x_{i6} + x_{i7} + x_{i8} + x_{i9} + x_{i10} + x_{i11} + x_{i12} + x_{i13} + x_{if} + x_{i3} = v_3 \Leftrightarrow$   
 $\Leftrightarrow 0+0+1+0+0+0+0+0+0+0+0+0+0 = 1 \Leftrightarrow \text{True}$   
 $x_{i4} + x_{i6} + x_{i7} + x_{i8} + x_{i9} + x_{i10} + x_{i11} + x_{i12} + x_{i13} + x_{if} + x_{i4} = v_4 \Leftrightarrow$   
 $\Leftrightarrow 0+0+1+0+0+0+0+0+0+0+0+0+0 = 1 \Leftrightarrow \text{True}$   
 $x_{i6} + x_{i7} + x_{i8} + x_{i9} + x_{i10} + x_{i11} + x_{i12} + x_{i13} + x_{if} + x_{i6} = v_6 \Leftrightarrow$   
 $\Leftrightarrow 0+1+0+0+0+0+0+0+0+0+0+0+0 = 1 \Leftrightarrow \text{True}$   
 $x_{i7} + x_{i8} + x_{i9} + x_{i10} + x_{i11} + x_{i12} + x_{i13} + x_{if} + x_{i7} = v_7 \Leftrightarrow$   
 $\Leftrightarrow 0+0+0+0+0+0+0+0+0+1+0+0+0 = 1 \Leftrightarrow \text{True}$   
 $x_{i8} + x_{i9} + x_{i10} + x_{i11} + x_{i12} + x_{i13} + x_{if} + x_{i8} = v_8 \Leftrightarrow$   
 $\Leftrightarrow 0+0+0+0+0+0+0+1+0+0+0+0+0 = 1 \Leftrightarrow \text{True}$

$$\begin{aligned}
& x_{19}+x_{29}+x_{39}+x_{49}+x_{69}+x_{79}+x_{89}+x_{910}+x_{911}+x_{912}+x_{913}+x_{9f}+x_{i9}=v_9 \Leftrightarrow \\
& \Leftrightarrow 0+0+0+0+0+0+0+1+0+0+0+0+0 = 1 \Leftrightarrow \text{True} \\
& x_{110}+x_{210}+x_{310}+x_{410}+x_{610}+x_{710}+x_{810}+x_{910}+x_{1011}+x_{1012}+x_{1013}+x_{10f}+x_{i10}=v_{10} \\
& \Leftrightarrow 0+0+0+0+0+0+0+1+0+0+0+0+0 = 1 \Leftrightarrow \text{True} \\
& x_{111}+x_{211}+x_{311}+x_{411}+x_{611}+x_{711}+x_{811}+x_{911}+x_{1011}+x_{1112}+x_{1113}+x_{11f}+x_{i11}=v_{11} \\
& \Leftrightarrow 0+0+0+0+0+0+1+0+0+0+0+0+0 = 1 \Leftrightarrow \text{True} \\
& x_{112}+x_{212}+x_{312}+x_{412}+x_{612}+x_{712}+x_{812}+x_{912}+x_{1012}+x_{1112}+x_{1213}+x_{12f}+x_{i12}=v_{12} \\
& \Leftrightarrow 0+0+0+0+0+0+0+0+0+0+0+1+0 = 1 \Leftrightarrow \text{True} \\
& x_{113}+x_{213}+x_{313}+x_{413}+x_{613}+x_{713}+x_{813}+x_{913}+x_{1013}+x_{1113}+x_{1213}+x_{13f}+x_{i13}=v_{13} \\
& \Leftrightarrow 0+0+0+0+0+1+0+0+0+0+0+0+0 = 1 \Leftrightarrow \text{True}
\end{aligned}$$

Assim, conseguimos perceber que a solução encontrada é assim uma solução admissível válida, pois respeita o conjunto de restrições estabelecidas, satisfazendo-as.

### 8.0.2 Validação Intuitiva

Em relação ao problema, o grupo foi-se apercebendo de particularidades específicas, inclusive ao longo da formulação do próprio modelo. Essas particularidades verificam-se na solução encontrada. Seguem-se abaixo essas mesmas particularidades:

- **É possível encontrar um caminho euleriano na solução ótima fornecida:** o grafo encontrado pela ferramenta possui, como se viu, um caminho proposto euleriano, pelo que os pontos de início (I) e fim (F) são efetivamente diferentes, tendo, a particularidade de coincidir com os vértices de grau ímpar, na solução encontrada. Para além disso, estes vértices são os únicos com grau ímpar, todos os outros têm grau par.
- **É possível percorrer todas as arestas pelo menos uma vez:** olhando para a solução ótima, percebemos que ao adicionarmos o custo fixo na função objetivo garantimos que todas as arestas fixas são percorridas. Somando-o com o custo adicional, garantimos que as arestas que forem adicionadas também são percorridas. Então, observamos que distância total percorrida pelo drone foi de 101.65 (valor resultante da soma de cada custo: 83 (custo fixo) com 18.65 (custo adicional)).

## 9 Conclusão e Apreciação crítica

São várias as conclusões importantes a que o grupo chegou. Nem sempre é fácil modelar um problema de programação linear e isso é um processo de reflexão. Algo que permitiu comprovar esta característica da modelação foi a reflexão do grupo em perceber que certas restrições (colocadas em comentário no ficheiro `.lp`) acabavam por ser desnecessárias e retirando-as, por exemplo. Neste mesmo caso, o grupo nunca pensou que fosse necessária a utilização de bastantes variáveis, foi, inclusive, algo que compreendemos ao longo da modelação, e é só mais uma das razões pelas quais a formulação se torna tão necessária. A utilização da programação linear para a resolução de problemas reais é assim muito útil e ajuda a obter soluções práticas.

Neste contexto, em relação às dificuldades encontradas, deparamo-nos com um “impasse” no raciocínio adotado em relação à “conversão” do grau dos vértices pares em ímpares: não tínhamos ideia do conjunto de restrições que tornariam essa conversão possível pois estávamos muito focados na conversão dos vértices de grau ímpar em par, esquecendo-nos que esse processo tem exatamente a mesma lógica que o processo inverso: a adição de arestas. Então, bastaria também adicionar arestas aos vértices de início e fim para os tornar em vértices de grau ímpar.

Ao longo do trabalho, seguimos uma abordagem possível, contudo, temos perfeita noção de que outras abordagens estariam igualmente corretas.

De um modo sumário, o trabalho realizado contribuiu para aprofundar o conhecimento alusivo à programação linear, através da aplicação prática dos conceitos e respetiva utilização e manipulação de software útil nessa mesma área. O conjunto de objetivos inicialmente estabelecidos foi cumprido e o projeto prático, na perspetiva do grupo, foi bem conseguido.

## 10 Referências Bibliográficas

I. Carvalho, J. M. V. D. (n.d.). Modelos de Investigação Operacional  
(Apêndice disponibilizado pelo docente)