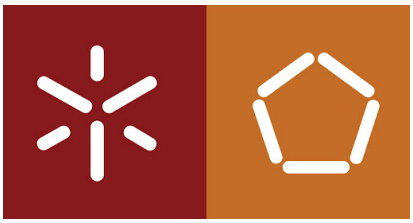

Universidade do Minho

Licenciatura em Engenharia Informática

Investigação Operacional
Trabalho Prático 1

Bernardo Saraiva (A93189)
José Gonçalves (A93204)
Rui Moreira (A93232)
Daniel Azevedo (A93324)
Pedro Araújo (A70699)

25/03/2022

Conteúdo

1 Introdução

O presente relatório, realizado no âmbito da disciplina de Investigação Operacional, visa expor a solução obtida para o problema da minimização do percurso de um *drone*, formulado no decorrer do relatório. Com este intuito, é usada a técnica de programação linear suportada pelo LPSolve, bem como alguns conceitos de teoria de grafos.

2 Alterações ao grafo inicial

De modo a representar todo o percurso a ser realizado pelo drone, é necessário recorrer ao grafo apresentado na Figura 1.

Dado que o maior número de inscrição entre os diversos elementos do grupo é o 93324, foram removidas do grafo original as arestas D e E, como demonstrado na Figura 1.

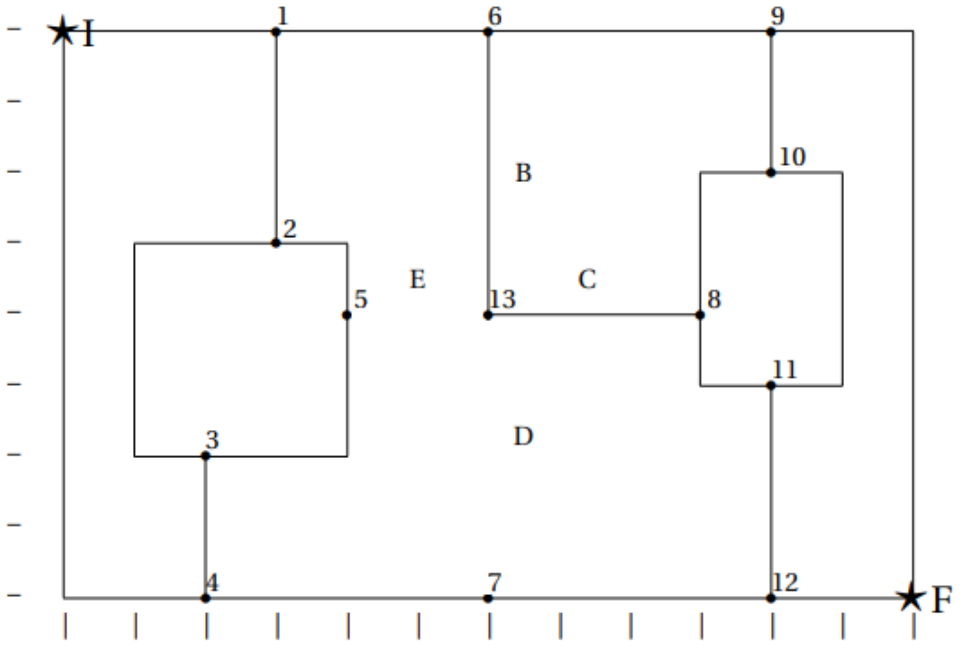

Figura 1: Grafo alterado de acordo com as instruções

3 Formulação do Problema

Num contexto real, no problema do (drone) pretende-se traçar um percurso que comece em qualquer ponto do grafo e retorne a esse mesmo ponto de forma a percorrer todas as arestas do grafo uma e uma única vez, sendo este conceito denominado na teoria de grafos como circuito Euleriano. Contudo, neste problema em concreto deparamo-nos com uma situação um pouco distinta, já que, o nodo inicial e final do percurso são diferentes e por esta razão encontramo-nos perante um problema de caminho Euleriano.

Uma vez que se torna necessário reposicionar o drone para recomeçar a inspeção de uma nova linha, o presente problema possibilita a criação de um caminho aéreo que não siga estritamente as arestas presentes no grafo, pelo que se terá que ligar os pontos de origem e fim através de um novo segmento de reta e calcular a distancia euclidiana para esse caso.

3.1 Objetivo

O objetivo do presente problema é determinar o percurso em que todas as linhas são percorridas pelo menos uma vez, de modo ao drone possa averiguar se existe vegetação a interferir com as linhas, minimizando a distância total percorrida.

3.2 Variáveis de Decisão

De modo a solucionar o problema do caminho Euleriano é necessário acrescentar algumas arestas ao grafo inicial, uma vez que todos os vértices necessitam ser pares, à exceção do vértice inicial e final. Estas arestas adicionais tornam o grafo admissível à resolução do problema e possibilitam o cálculo da solução ótima.

Sendo assim, é necessário começar por definir as variáveis de decisão. As variáveis de decisão irão representar as ligações adicionais existentes entre os vértices ímpares e o vértice inicial e final, de forma a criar um grafo em que todos os seus vértices tem grau par excepto os vértices anteriormente referidos. Consequentemente, estas variáveis serão **variáveis binárias** que indicam se a mesma é selecionada e adicionada ao percurso (valor 1) ou caso contrário (valor 0).

3.3 Função Objetivo

Visto que existe o objetivo de minimizar a distância percorrida, a função objetivo será obrigatoriamente de minimização (minimiza o custo da aresta a escolher) e dará como resultado a distancia total percorrida, ou seja, o custo de percorrer todos os vértices mais o custo dos caminhos "extra"a percorrer pelo *drone*.

3.4 Restrições

De forma a garantir a transformação de todos os vértices numéricos (1,2,...) em vértices pares e o vértice inicial e final (I e J) em vértices ímpares, existe a necessidade de garantir que o emparelhamento escolhido não contém arestas diferentes com vértices em comum. Caso contrário, poderia acontecer de o vértice continuar a ser de grau ímpar, não resolvendo o problema inicial. Assim, para cada vértice foi necessária a criação de uma restrição que impede a seleção de duas variáveis de decisão com valores de índice em comum.

4 Modelo de programação Linear

4.1 Variáveis de Decisão

Como variáveis de decisão, foi utilizada uma notação que indica um caminho entre um par de vértices. Deste modo, representou-se o caminho referido através de uma notação x_{ij}, sendo i e j os identificadores numéricos da origem e destino, respectivamente.

É de referir que, como os vértices 5, 7 e 13 já são vértices com grau par, sendo que não fazem parte do conjunto. De notar também, que se incluiu o I e o F visto existir a necessidade de transformar estes em vértices ímpares.

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

Sendo que:

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

De salvar que se tem em atenção que $i < j$, de forma a evitar a ambiguidade em arestas como x_{12} e x_{21} (ambas as formas representam a mesma aresta).

4.2 Parâmetros

Neste problema, o principal parâmetro é a distância Euclidiana entre dois vértices i e j, C_{ij} (que poderá coincidir com uma linha de alta tensão).

4.3 Função Objetivo

Visto que se pretende que o drone percorra a menor distância possível, a função objetivo irá consistir numa minimização dos diversos custos, o custo de percorrer

todas as arestas (valor 79) mais o custo associado às arestas "extras"selecionadas. Assim a função objetivo terá a seguinte forma:

$$min\ z = \sum C_{ij} \cdot x_{ij}$$

em que C_{ij} corresponde ao custo de percorrer a aresta ij, por sua vez x_{ij} é uma variável de decisão binária que indica se a aresta ij foi ou não escolhida.

De notar que:

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

4.4 Restrições

Como referido acima, para garantir que se transforma os vértices ímpares em vértices pares, e nos casos dos vértices inicial e final em vértices ímpares, a soma de todas as possibilidades de caminhos incidentes em cada vértice terá que ser igual a 1. Isto garante que, devido ao facto de se estar a somar todas as variáveis binárias a esses caminhos, apenas uma delas será 1 e todas as restantes serão iguais a 0.

Por exemplo para o Vértice 1:

$$V1 : x_{12} + x_{13} + x_{14} + x_{16} + x_{18} + x_{19} + x_{1\_10} + x_{1\_11} + x_{1\_12} + x_{1\_I} + x_{1\_F} = 1;$$

5 Ficheiro de input

```
/* Objective function */
min: 79 +
     3 x12 + 6.08 x13 + 8.06 x14 +    3 x16 + 7.21 x18 +    7 x19 + 7.28 x1_10 + 8.60 x1_11 + 10.63 x1_12 +      3 x1_I + 12.04 x1_F +
             3.16 x23 + 5.10 x24 + 4.24 x26 + 6.08 x28 + 7.62 x29 + 7.07 x2_10 + 7.28 x2_11 +  8.60 x2_12 +   4.24 x2_I + 10.30 x2_F +
                              2 x34 + 7.21 x36 + 7.28 x38 +   10 x39 + 8.94 x3_10 + 8.06 x3_11 +  8.25 x3_12 +   6.32 x3_I + 10.20 x3_F +
                                      8.94 x46 + 8.06 x48 + 11.31x49 +   10 x4_10 + 8.54 x4_11 +     8 x4_12 +   8.25 x4_I +    10 x4_F +
                                                   5 x68 +    4 x69 + 4.47 x6_10 + 6.40 x6_11 +  8.94 x6_12 +      6 x6_I +    10 x6_F +
                                                           4.12 x89 + 2.24 x8_10 + 1.41 x8_11 +  4.12 x8_12 +   9.85 x8_I +     5 x8_F +
                                                                      2 x9_10 +    5 x9_11 +     8 x9_12 +     10 x9_I +  8.25 x9_F +
                                                                              3 x10_11 +    6 x10_12 + 10.20 x10_I + 6.32 x10_F +
                                                                                          3 x11_12 + 11.18 x11_I + 3.61 x11_F +
                                                                                                12.81 x12_I +    2 x12_F +
                                                                                                             14.42 xI_F;
```

Figura 2: Função Objectivo

```
/* Variable bounds */
/*V5 V7 V13 são pares, logo não entram*/
V1:          x12   + x13   + x14   + x16   + x18   + x19   + x1_10  + x1_11 + x1_12  + x1_I  + x1_F  = 1;
V2:  x12   +         x23   + x24   + x26   + x28   + x29   + x2_10  + x2_11 + x2_12  + x2_I  + x2_F  = 1;
V3:  x13   + x23   +         x34   + x36   + x38   + x39   + x3_10  + x3_11 + x3_12  + x3_I  + x3_F  = 1;
V4:  x14   + x24   +  x34  +         x46   + x48   + x49   + x4_10  + x4_11 + x4_12  + x4_I  + x4_F  = 1;
V6:  x16   + x26   + x36   + x46   +         x68   + x69   + x6_10  + x6_11 + x6_12  + x6_I  + x6_F  = 1;
V8:  x18   + x28   + x38   + x48   + x68   +         x89   + x8_10  + x8_11 + x8_12  + x8_I  + x8_F  = 1;
V9:  x19   + x29   + x39   + x49   + x69   + x89   +         x9_10  + x9_11 + x9_12  + x9_I  + x9_F  = 1;
V10: x1_10 + x2_10 + x3_10 + x4_10 + x6_10 + x8_10 + x9_10 +         x10_11 + x10_12 + x10_I + x10_F = 1;
V11: x1_11 + x2_11 + x3_11 + x4_11 + x6_11 + x8_11 + x9_11 + x10_11 +          x11_12 + x11_I + x11_F = 1;
V12: x1_12 + x2_12 + x3_12 + x4_12 + x6_12 + x8_12 + x9_12 + x10_12 + x11_12          + x12_I + x12_F = 1;
VI:  x1_I  + x2_I  + x3_I  + x4_I  + x6_I  + x8_I  + x9_I  + x10_I  + x11_I + x12_I           + xI_F  = 1;
VF:  x1_F  + x2_F  + x3_F  + x4_F  + x6_F  + x8_F  + x9_F  + x10_F  + x11_F + x12_F + xI_F            = 1;


/*as seguintes variaveis sao binarias*/
bin x12, x13, x14, x16, x18, x19, x1_10, x1_11,   x1_12, x1_I, x1_F,
          x23, x24, x26, x28, x29, x2_10, x2_11,   x2_12, x2_I, x2_F,
               x34, x36, x38, x39, x3_10, x3_11,   x3_12, x3_I, x3_F,
                    x46, x48, x49, x4_10, x4_11,   x4_12, x4_I, x4_F,
                         x68, x69, x6_10, x6_11,   x6_12, x6_I, x6_F,
                              x89, x8_10, x8_11,   x8_12, x8_I, x8_F,
                                   x9_10, x9_11,   x9_12, x9_I, x9_F,
                                          x10_11, x10_12, x10_I, x10_F,
                                                  x11_12, x11_I, x11_F,
                                                       x12_I, x12_F,
                                                              xI_F;
```

Figura 3: Restrições

6 Ficheiro de output

```
Model name:  'LPSolver' - run #1
Objective:   Minimize(R0)

SUBMITTED
Model size:      12 constraints,      66 variables,         132 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              93.65 after        10 iter is B&B base.

Feasible solution             93.65 after        10 iter,        0 nodes (gap 0.0%)

Optimal solution              93.65 after        10 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 10, 0 (0.0%) were bound flips.
      There were 0 refactorizations, 0 triggered by time and 0 by density.
       ... on average 10.0 major pivots per refactorization.
      The largest [LUSOL v2.2.1.0] fact(B) had 13 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 0.003 seconds, presolve used 0.006 seconds,
       ... 0.008 seconds in simplex solver, in total 0.017 seconds.
```

Figura 4: Output apresentado no terminal

Variables	MILP ...	result
	93,65	93,65
x12	0	0
x13	0	0
x14	0	0
x16	0	0
x18	0	0
x19	0	0
x1_10	0	0
x1_11	0	0
x1_12	0	0
x1_I	1	1
x1_F	0	0
x23	0	0
x24	0	0
x26	1	1
x28	0	0
x29	0	0
x2_10	0	0
x2_11	0	0
x2_12	0	0
x2_I	0	0
x2_F	0	0
x34	1	1
x36	0	0
x38	0	0
x39	0	0
x3_10	0	0
x3_11	0	0
x3_12	0	0
x3_I	0	0
x3_F	0	0
x46	0	0
x48	0	0
x49	0	0
x4_10	0	0
x4_11	0	0
x4_12	0	0
x4_I	0	0
x4_F	0	0
x68	0	0
x69	0	0
x6_10	0	0
x6_11	0	0
x6_12	0	0
x6_I	0	0
x6_F	0	0
x89	0	0
x8_10	0	0
x8_11	1	1
x8_12	0	0
x8_I	0	0
x8_F	0	0
x9_10	1	1
x9_11	0	0
x9_12	0	0
x9_I	0	0
x9_F	0	0
x10_11	0	0
x10_12	0	0
x10_I	0	0
x10_F	0	0
x11_12	0	0
x11_I	0	0
x11_F	0	0
x12_I	0	0
x12_F	1	1
xI_F	0	0

Figura 5: Output LPsolve

7 Solução Ótima

Na solução obtida, verifica-se que as variáveis x_{1_I}, x_{26} , x_{34}, x_{8_11}, x_{9_10} , x_{12_F} têm valor igual a 1, o que significa que estas arestas são duplicadas de modo a obter-se um grafo com todos os vértices de grau par excepto o vértice de partida e o vértice de chegada (I e F).

Esta condição verifica-se no seguinte grafo (Figura 6), onde já se encontram adicionadas as arestas correspondestes às variáveis da solução com valor igual a 1.

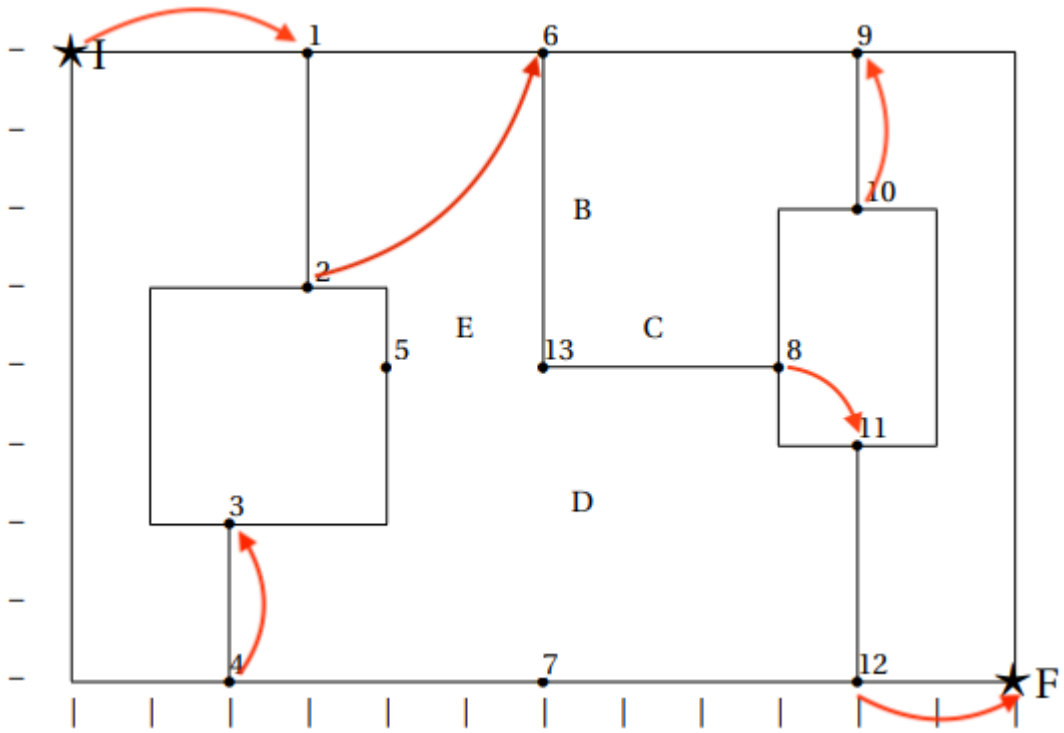

Figura 6: Grafo obtido

Sabendo agora quais as linhas de tensão por onde o drone só passa uma vez, as linhas de tensão que são repetidas e as arestas diagonais a percorrer pelo ar, existem vários percursos possíveis a percorrer. Um possível percurso é identificado na Figura 7.

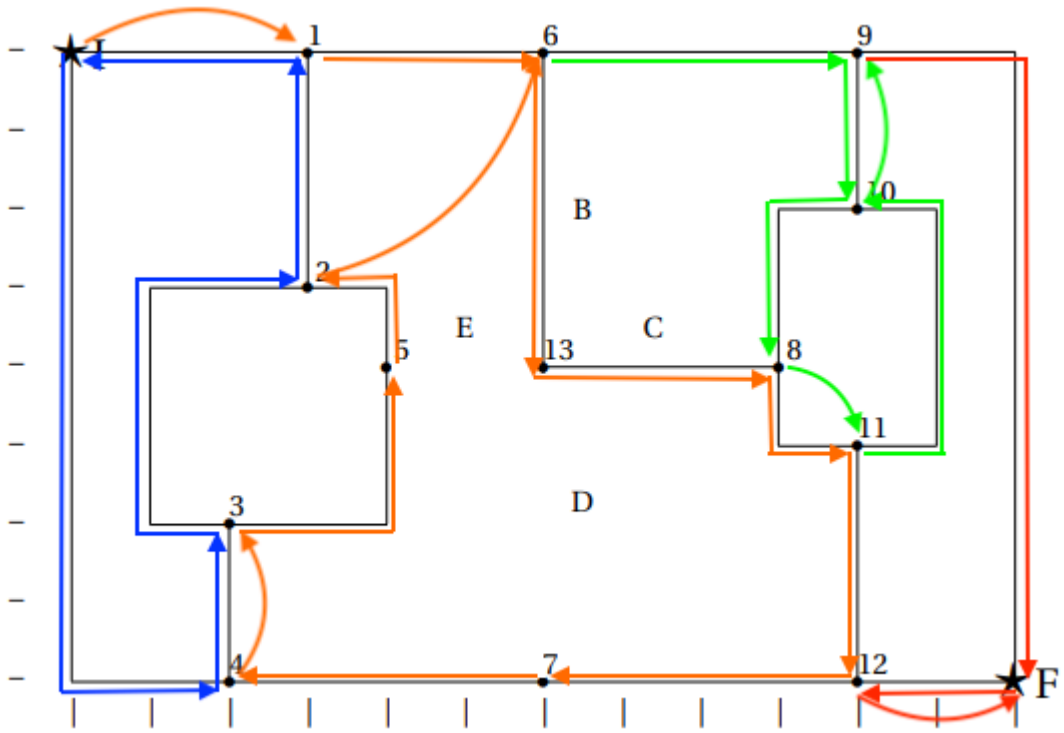

Figura 7: Percurso Possível

Percurso:

Azul → *Laranja* → *Verde* → *Vermelho*

Ou seja:

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

Assim, a distância total percorrida é **93.65**.

8 Validação do modelo

Após o cálculo da solução ótima através LPsolve, é então necessário validar o resultado obtido, de forma a que a verificar a otimalidade. Com esta finalidade, são necessárias as seguintes verificações:

- Todos os vértices do grafo possuem grau par à exeção do nodo inicial e final, como é possível verificar na Figura 6, o que demonstra que as restrições quanto à seleção de arestas aéreas encontram-se corretamente implementadas.

- É possível traçar no grafo obtido após a atribuição das novas arestas, utilizando todas as arestas uma única vez, um caminho de I para F, como é possível determinar através da Figura 7
- A distância percorrida obtida através do modelo é a mínima possível dadas todas as restrições, corresponde portanto a todas as arestas do percurso: $10 + 2 + 6 + 3 + 3 + 3 + 3 + 4 + 3 + 2 + 3 + 4 + 4 + 2 + 4 + 2 + 4.24 + 4 + 2 + 3 + 1.41 + 5 + 2 + 10 + 2 + 2 = 93.65 \equiv$ Solução ótima (obtida através da função objetivo). Para além disto, esta distância é admissível, já que, $93.65 > 79$ (soma total das arestas iniciais), o que reflete a inserção de arestas aéreas.

9 Conclusão

A redução do custo do caminho a percorrer é de grande importância no que toca ao planeamento e gestão de recursos e é usado constantemente no mundo empresarial. Neste trabalho, com base na teoria de grafos e com os conceitos de programação linear, foi possível encontrar uma solução ótima para o problema apresentado, em que o drone percorrerá 93,65.

Em suma, através do presente trabalho para além de ter sido possível consolidar os diversos conhecimentos leccionados durante as aulas, foi também possível solidificar o conceito de Investigação Operacional e perceber a sua importância no contexto do mundo real.