Drone

Trabalho realizado por:

João Figueiredo Martins Peixe dos Santos

Francisco Alves Andrade

Luís Filipe Cruz Sobral

Paulo Silva Sousa

Meriem Khammassi

A89520
João Santos

A89474
Luís Sobral

A85829
Meriem Khammassi

A89465
Paulo Sousa

A89513
Francisco Andrade

Projeto MDIO
2020/2021
Universidade do Minho

Índice

1 Problema

O problema que pretendemos solucionar baseia-se em determinar o menor percurso que um drone deve efetuar para inspecionar linhas de transporte de energia cinética em alta tensão. Assim, todos os vértices do grafo devem ser percorridos uma ou mais vezes em qualquer sentido, com o objetivo de encontrar a melhor solução, ou seja, a que minimize a distância euclidiana total. As linhas têm um comprimento proporcional à dimensão do seu traço em centímetros. É necessário, para provar que existe solução, encontrar um percurso com início e fim na estrela que vá de um vértice para qualquer outro percorrendo todos os vértices.

2 Remoção de arestas

Como o maior número de inscrição do nosso grupo é 89520, removemos as arestas D e E, de acordo com as regras estipuladas no enunciado do trabalho. Assim, a figura do grafo é a seguinte:

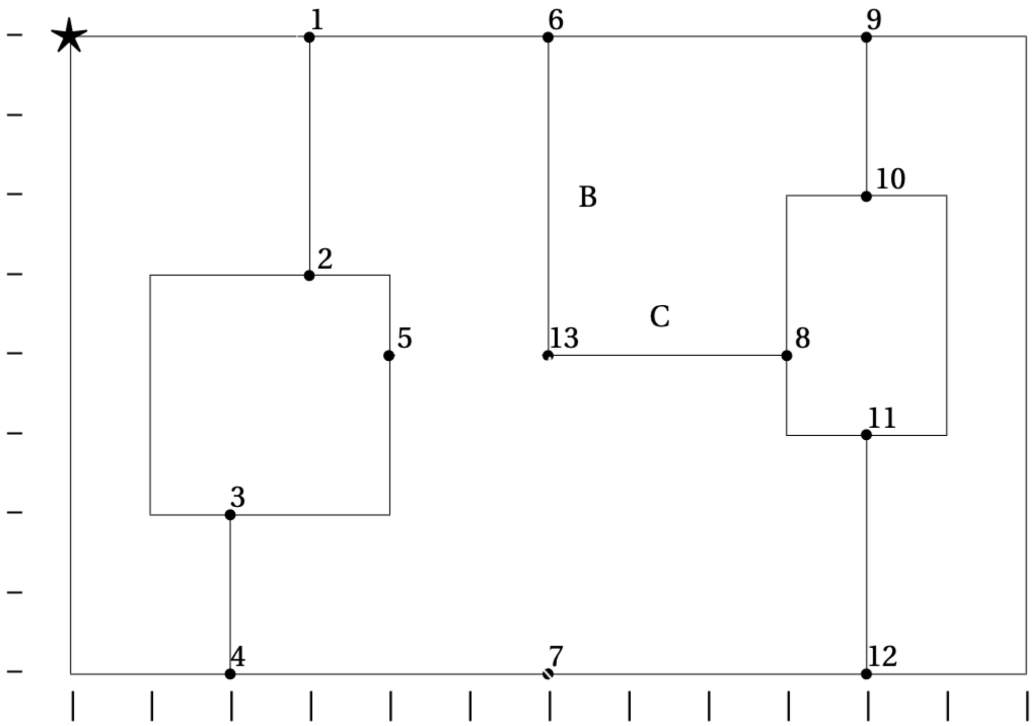

Fig 1 - Grafo do Problema

3 Variáveis de decisão

Por objetivo temos de criar um grafo cujos vértices tenham grau par. Assim, definimos a variável de decisão que representa as ligações adicionais entre os vértices ímpares, de modo a inicializar a foprmulação do modelo de programação linear. Esta é uma variável binária que representa a seleção da aresta, 1 se for selecionada e 0 se não for.

Xij: arestas adicionais entre os vértices i e j.

$$i, j \epsilon \{1, 2, 3, 4, 6, 9, 8, 10, 11, 12\} \tag{1}$$

Tendo em mente tornar o número de incidências dos vértices ímpares em pares, possibilitando a construção de um percurso para o drone, adicionaremos uma aresta aos vértices de grau ímpar, não alterando os vértices de grau par (5, 7 e 13).

4 Dados

Os dados do modelo de programação linear tiveram por base a distância das arestas que foi calculada considerando como escala a dimensão do seu traço em centímetros e o custo de cada ligação entre vértices (representada pela distância euclidiana presente na matriz do enunciado.

5 Restrições

As restrições são as regras de seleção de ligações adicionais. Assim, de modo a garantir a unicidade de um determinado caminho a partir de um vértice, definimos (para os vértices de grau ímpar) uma restrição que certifica que apenas uma das arestas é selecionada. Esta restrição consiste na igualdade entre a soma de todas variáveis binárias (representantes de todas as possibilidades para onde a aresta pode ir)e o valor 1. Desta forma obtemos uma ligação adicional igual a 1 para cada vértice de grau ímpar, sendo que as restantes ligações admissíveis igualam a 0 e não ocorrem.

Vértice 1: x12 + x13 + x14 + x16 + x18 + x19 + x110 + x111 + x112 = 1;
Vértice 2: x12 + x23 + x24 + x26 + x28 + x29 + x210 + x211 + x212 = 1;
Vértice 3: x13 + x23 + x34 + x36 + x38 + x39 + x310 + x311 + x312 = 1;
Vértice 4: x14 + x24 + x34 + x46 + x48 + x49 + x410 + x411 + x412 = 1;
Vértice 6: x16 + x26 + x36 + x46 + x68 + x69 + x610 + x611 + x612 = 1;
Vértice 8: x18 + x28 + x38 + x48 + x68 + x89 + x810 + x811 + x812 = 1;
Vértice 9: x19 + x29 + x39 + x49 + x69 + x89 + x910 + x911 + x912 = 1;
Vértice 10: x110 + x210 + x310 + x410 + x610 + x810 + x910 + x1011 + x1012 = 1;
Vértice 11: x111 + x211 + x311 + x411 + x611 + x811 + x911 + x1011 + x1112 = 1;
Vértice 12: x112 + x212 + x312 + x412 + x612 + x812 + x912 + x1012 + x1112 = 1;

bin x12 x13 x14 x16 x18 x19 x110 x111 x112 x23 x24 x26 x28 x29 x210 x211 x212 x34 x36 x38 x39 x310 x311 x312 x46 x48 x49 x410 x411 x412 x68 x69 x610 x611 x612 x89 x810 x811 x812 x910 x911 x912 x1011 x1012 x1112;

De forma a evitar a repetição de variáveis, declaramos as variáveis como binárias atrevés do bin. Não sendo assim necessário mencionar a aresta x21 quando a aresta x12 já foi declarada.

6 Função Objetivo

A função objetivo traduz o custo das ligações adicionais. Sendo que as linhas de alta tensão têm valor constante, o que podemos otimizar é o custo das ligações. Para determinar a função objetivo, somamos o tamanho de todas as arestas (com o resultado de 79) e adicionamos todas as variáveis binárias a multiplicar pela ditância euclidiana.

Min: 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 x110 + 8.60 x111 + 10.63 x112 + 5.00 x113 + 3.00 x21 + 3.16 x23 + 5.10 x24 + 1.41 x25 + 4.24 x26 + 5.83 x27 + 6.08 x28 + 7.62 x29 + 7.07 x210 + 7.28 x211 + 8.60 x212 + 3.16 x213 + 6.08 x31 + 3.16 x32 + 2.00 x34 + 2.83 x35 + 7.21 x36 + 4.47 x37 + 7.28 x38 + 10.00 x39 + 8.94 x310 + 8.06 x311 + 8.25 x312 + 4.47 x313 + 8.06 x41 + 5.10 x42 + 2.00 x43 + 4.47 x45 + 8.94 x46 + 4.00 x47 + 8.06 x48 + 11.31 x49 + 10.00 x410 + 8.54 x411 + 8.00 x412 + 5.66 x413 + 4.12 x51 + 1.41 x52 2.83 x53 + 4.47 x54 + 4.47 x56 + 4.47 x57 + 5.00 x58 + 7.21 x59 + 6.32 x510 + 6.08 x511 + 7.21 x512 + 2.00 x513 + 3.00 x61 + 4.24 x62 + 7.21 x63 + 8.94 x64 + 4.47 x65 + 8.00 x67 + 5.00 x68 + 4.00 x69 + 4.47 x610 + 6.40 x611 + 8.94 x612+ 4.00 x613 + 8.54 x71 + 5.83 x72 + 4.47 x73 + 4.00 x74 + 4.47 x75 + 8.00 x76 + 5.00 x78 + 8.94 x79 + 7.21 x710 + 5.00 x711 + 4.00 x712 + 4.00 x713 + 7.21 x81 + 6.08 x82 + 7.28 x83 + 8.06 x84 + 5.00 x85 + 5.00 x86 + 5.00 x87 + 4.12 x89 + 2.24 x810 + 1.41 x811 + 4.12 x812 + 3.00 x813 + 7.00 x91 + 7.62 x92 + 10.00 x93 + 11.31 x94 + 7.21 x95 + 4.00 x96 + 8.94 x97 + 4.12 x98 + 2.00 x910 + 5.00 x911 + 8.00 x912 + 5.66 x913 + 7.28 x101 + 7.07 x102 + 8.94 x103 + 10.00 x104 + 6.32 x105 + 4.47 x106 + 7.21 x107 + 2.24 x108 + 2.00 x109 + 3.00 x1011 + 6.00 x1012 + 4.47 x1013 + 8.60 x111 + 7.28 x112 + 8.06 x113 + 8.54 x114 + 6.08 x115 + 6.40 x116 + 5.00 x117 + 1.41 x118 + 5.00 x119 + 3.00 x1110 + 3.00 x1112 + 4.12 x1113 + 10.63 x121 + 8.60 x122 + 8.25 x123 + 8.00 x124 + 7.21 x125 + 8.94 x126 + 4.00 x127 + 4.12 x128 + 8.00 x129 + 6.00 x1210 + 3.00 x1211 + 5.66 x1213 + 5.00 x131 + 3.16 x132 + 4.47 x133 + 5.66 x134 + 2.00 x135 + 4.00 x136 + 4.00 x137 + 3.00 x138 + 5.66 x139 + 4.47 x1310 + 4.12 x1311 + 5.66 x1212 + 79;

7 Ficheiro Input

As restrições e a função objetivo anteriormente definidas foram introduzidas no programa LPSolve.

```
/* Objective function */
min: 79 + 3 x12 + 6.08 x13 + 8.06 x14 + 3 x16 + 7.21 x18 + 7 x19 + 7.28 x110 + 8.6 x111 + 10.63 x112
+ 3.16 x23 + 5.10 x24 + 4.24 x26 + 6.08 x28 + 7.62 x29 + 7.07 x210 + 7.28 x211 + 8.6 x212
+ 2 x34 + 7.21 x36 + 7.28 x38 + 10 x39 + 8.94 x310 + 8.06 x311 + 8.25 x312
+ 8.94 x46 + 8.06 x48 + 11.31 x49 + 10 x410 + 8.54 x411 + 8 x412
+ 5 x68 + 4 x69 + 4.47 x610 + 6.4 x611 + 8.94 x612
+ 4.12 x89 + 2.24 x810 + 1.41 x811 + 4.12 x812
+ 2 x910 + 5 x911 + 8 x912
+ 3 x1011 + 6 x1012
+ 3 x1112;

/* Variable bounds */
//V1
x12+x13+x14+x16+x18+x19+x110+x111+x112 = 1;
//V2
x12+x23+x24+x26+x28+x29+x210+x211+x212 = 1;
//V3
x13+x23+x34+x36+x38+x39+x310+x311+x312 = 1;
//V4
x14+x24+x34+x46+x48+x49+x410+x411+x412 = 1;
//V6
x16+x26+x36+x46+x68+x69+x610+x611+x612 = 1;
//V8
x18+x28+x38+x48+x68+x89+x810+x811+x812 = 1;
//V9
x19+x29+x39+x49+x69+x89+x910+x911+x912 = 1;
//V10
x110+x210+x310+x410+x610+x810+x910+x1011+x1012 = 1;
//V11
x111+x211+x311+x411+x611+x811+x911+x1011+x1112 = 1;
//V12
x112+x212+x312+x412+x612+x812+x912+x1012+x1112 = 1;

//bin
bin x12 x13 x14 x16 x18 x19 x110 x111 x112
x23 x24 x26 x28 x29 x210 x211 x212
x34 x36 x38 x39 x310 x311 x312
x46 x48 x49 x410 x411 x412 x68 x69 x610 x611 x612 x89 x810 x811 x812 x910 x911 x912 x1011 x1012 x1112;
```

Fig 2 - Ficheiro de Input

8 Ficheiro Output

A solução ótima para o problema foi encontrada após executar o input. Sendo assim possível encontrar as ligações adicionais com menor distância euclidiana, convertendo todos os vértices ímpares em vértices pares.

```
Select LPSolve IDE

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

SUBMITTED
Model size:     10 constraints,     153 variables,        90 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          92.385 after       10 iter is B&B base.

Feasible solution          93.24 after       11 iter,       1 nodes (gap 0.9%)

Optimal solution           93.24 after       12 iter,       2 nodes (gap 0.9%).

Excellent numeric accuracy ||*|| = 0

MEMO: lp_solve version 5.5.2.0 for 32 bit OS, with 64 bit REAL variables.
    In the total iteration count 12, 0 (0.0%) were bound flips.
    There were 1 refactorizations, 0 triggered by time and 0 by density.
     ... on average 12.0 major pivots per refactorization.
    The largest [LUSOL v2.2.1.0] fact(B) had 33 NZ entries, 1.1x largest basis.
    The maximum B&B level was 2, 0.0x MIP order, 2 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 0.009 seconds,
     ... 0.016 seconds in simplex solver, in total 0.028 seconds.
```

Source | Matrix | Options | Result

Objective | Constraints | Sensitivity

Variables	MILP ...	re...
	93.24	93.24
x12	1	1
x34	1	1
x69	1	1
x810	1	1
x1112	1	1
x13	0	0
x14	0	0
x15	0	0
x16	0	0
x17	0	0
x18	0	0
x19	0	0
x110	0	0
x111	0	0
x112	0	0
x113	0	0
x21	0	0
x23	0	0
x24	0	0
x25	0	0
x26	0	0
x27	0	0
x28	0	0
x29	0	0
x210	0	0
x211	0	0
x212	0	0
x213	0	0

Fig 3 - Terminal de Output

Fig 4 - Resultados

9 Solução Ótima, Percurso e Distância

A partir da imagem gerada pelo LPSolve conseguimos descobrir as ligações que devem ser adicionadas, uma vez que, obtivemos o valor 1 para as arestas do caminho ótimo. Ligações a adicionar (sem sentido associado): - (vértice 1 - vértice 2) distancia euclidiana: 3,00 - (vértice 3 - vértice 4) distancia euclidiana: 2,00 - (vértice 6 - vértice 9) distancia euclidiana: 4,00 - (vértice 8 - vértice 10) distancia euclidiana: 2,24 - (vértice 11 - vértice 12) distancia euclidiana: 3,00

A solução ótima, que esta indicada na figura a cima, tem o valor 96,24. Este é alcançado a partir da soma das arestas existentes inicialmente com as ligações adicionadas.

Distância total percorrida = 79 + 3 + 2 + 4 + 2,24 + 3 = 93,24

Assim, obtivemos o seguinte percurso:

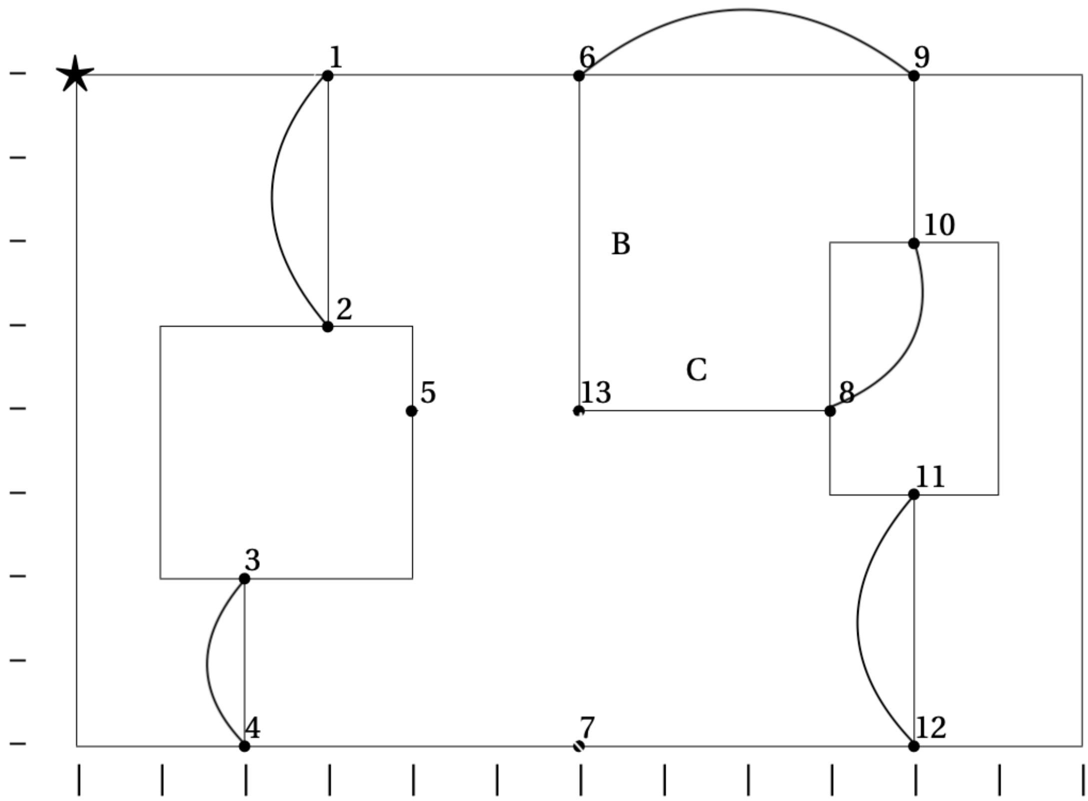

Fig 5 - Solução ótima

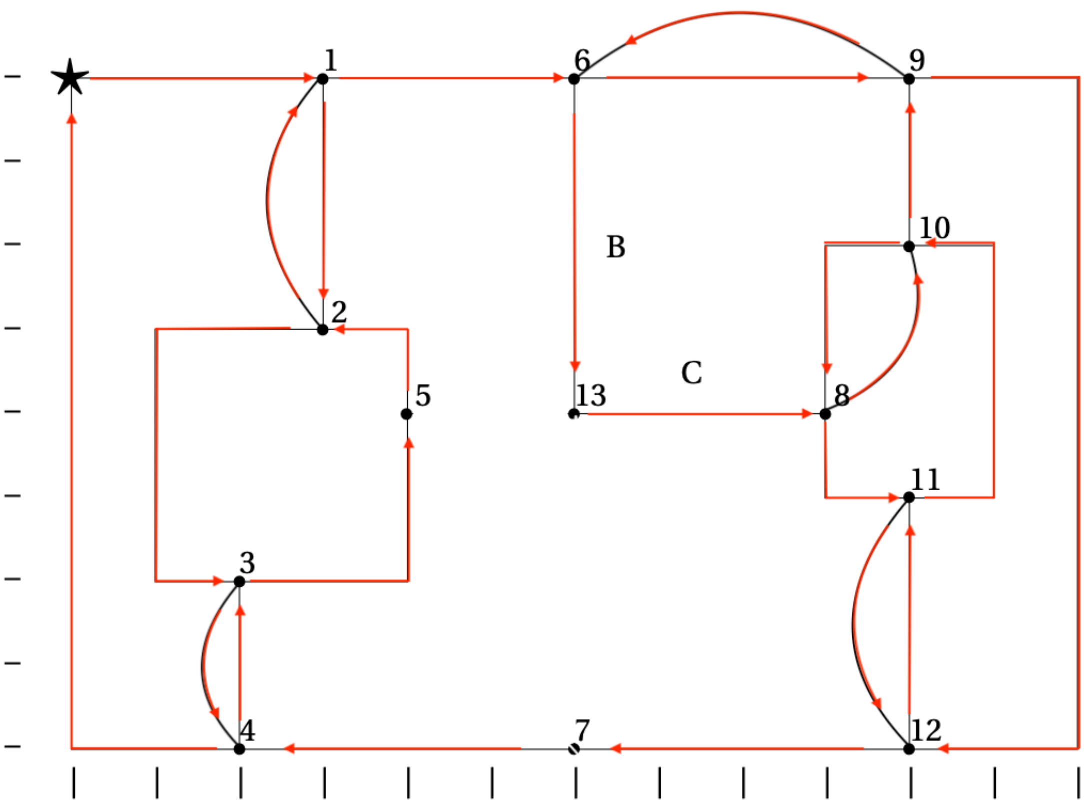

Fig 6 - Exemplo de caminho

Um exemplo de percurso é: estrela-1, 1-2, 2-1, 1-6, 6-9, 9-6, 6-13, 13-8, 8-10, 10-8, 8-11, 11-10, 10-9, 9-12, 12-11, 11-12, 12-7, 7-4, 4-3, 3-5, 5-2, 2-3, 3-4, 4-estrela.

10 Validação do Modelo

Vértices são pares: O resultado obtido no LPSolve indicou quais as açigações a adicionar aos vértices de graus ímpar, ou seja, as restrições garantem que todos os vértices ficam com o número de arestas incidentes par. Assim, no percurso obtido todas as arestas são percorridas.

Todas as arestas são percorridas: Através da solução ótima conseguimos obter a distancia total percorrida (96,24), valor calculado a partir da soma de todas as arestas (82) mais a que foram adicionadas (14,24), o que garante que todas as arestas estão incluídas no circuito do drone.

Início e fim do percurso: O início do pecurso do drone é o ponto assinalado com uma estrela (correspondente ao depósito), o fim do percurso é o ponto inicial, uma vez que, todas as arestas são percorridas são somadas (a razão pela qual somamos 82 à função objetivo) e é sempre possível o drone acabar na estrela.