



Universidade do Minho  
Departamento de Informática

MDIO – Trabalho 1

Carlos Preto (a89587)

Maria João Moreira (a89540)

Pedro Veloso (a89557)

Rui Fernandes (a89138)

13 de Novembro de 2020

## Índice

1. Introdução .....	3
2. Problema .....	4
2.1. Definição do problema .....	4
2.2. Análise do problema .....	5
3. Formulação do problema .....	6
3.1. Variáveis de decisão .....	6
3.2. Restrições .....	6
3.3. Função Objetivo .....	7
4. Texto Input .....	8
5. Ficheiro Output .....	9
6. Interpretação do resultado .....	10
7. Resultado do Problema .....	11
8. Validação do Modelo .....	12
8.1. Tipo de Variáveis .....	12
8.2. Função Objetivo .....	12
8.3. Restrições .....	12
9. Conclusão .....	14

## Índice de Figuras

Figura 1: Representação do problema .....	4
Figura 2: Vértices de grau ímpar no problema .....	5
Figura 3: Script que transcreve o modelo desenvolvido .....	8
Figura 4: Output produzido pelo lp_solve .....	9
Figura 5: Grafo Euleriano obtido .....	10
Figura 6: Representação da solução do problema .....	11

## Índice de Tabelas

Tabela 1: Distâncias associadas a cada aresta .....	7
---	---

## Índice de Equações

Equação 1: Variáveis de decisão .....	6
Equação 2: Função restrição .....	6
Equação 3: Função objetivo .....	7

## 1. Introdução

O trabalho prático consiste na minimização do percurso de um drone durante a atividade de inspeção de linhas de transporte de energia elétrica em alta tensão para verificar se há vegetação a interferir com as linhas, conhecendo a disposição das linhas de alta tensão bem como o seu comprimento.

Quando transferido para um grafo orientado, este problema resume-se a encontrar o caminho mais curto efetuado pelo drone que visite todos os nodos do grafo.

O objetivo deste trabalho prático é resolver este problema utilizando programação linear.

## 2. Problema

Em Investigação Operacional podem ser expressos vários problemas práticos como problemas de programação linear, tal como acontece no problema descrito em seguida. Sendo assim, optou-se por sintetizar o problema nos próximos pontos.

### 2.1. Definição do problema

O problema apresentado é um problema comum onde é necessário determinar o percurso em que todas as arestas de um grafo são percorridas pelo menos, uma vez, minimizando a distância total percorrida.

Neste problema existe um veículo não tripulado (drone) que tem de inspecionar linhas de transporte de energia elétrica em alta tensão para verificar se há vegetação a fazer interferência. O drone deverá passar em todas as linhas pelo menos uma vez. No caso de ser necessário passar por uma linha já inspecionada, o drone pode assumir o trajeto mais curto, mesmo que isso implique não ter de percorrer os caminhos definidos pelas linhas.

O drone poderá também ir do ponto em que se encontra para qualquer outro realizando a viagem pelo ar, sem passar em qualquer linha.

Observando os números de inscrição dos membros do grupo, constatamos que o maior número de inscrição era 89587. Fazendo *pattern matching* para o padrão ABCDE e seguindo as regras definidas no enunciado, concluímos quais as arestas que deviam ou não ser removidas: neste caso, a aresta D.

Assim, conclui-se que o problema a resolver é representado pela seguinte imagem:

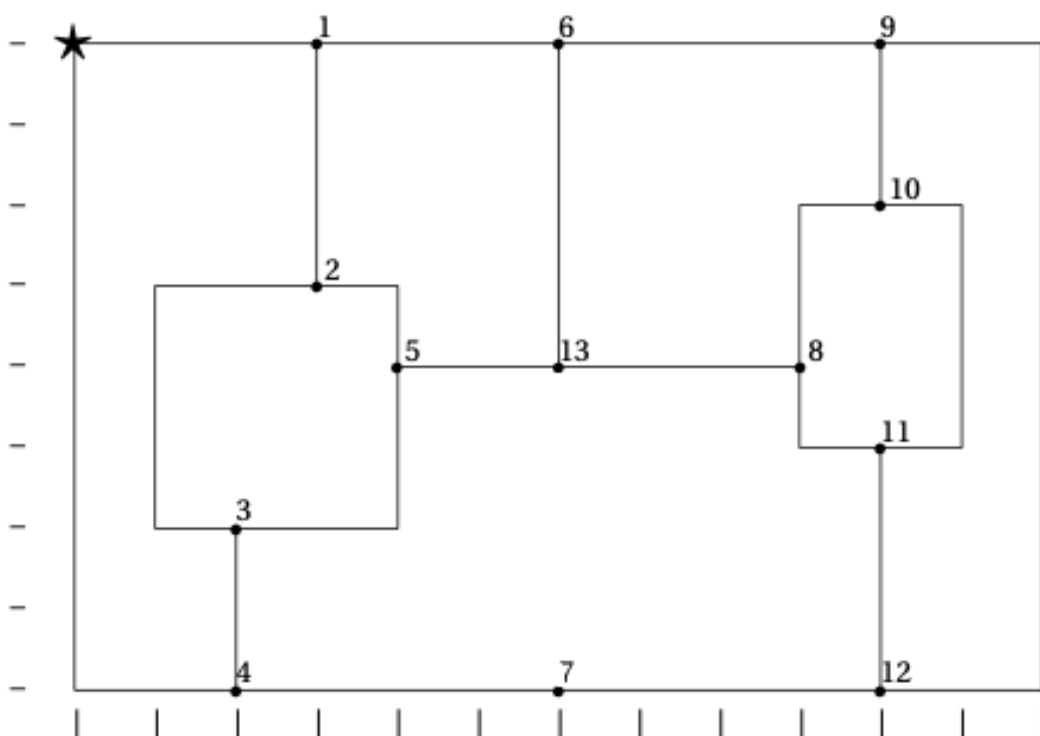


Figura 1: Representação do problema

## 2.2. Análise do problema

Ao analisar as linhas de transporte de energia elétrica em alta tensão apresentadas na Figura 1, percebe-se que é possível tratá-las como um grafo. Assim sendo, para resolver o problema é necessário ter em atenção os seguintes conceitos:

- **Caminho Euleriano:** um grafo tem um caminho Euleriano quando é possível selecionar um vértice de partida e percorrer todas as suas arestas uma e uma só vez;
- **Circuito Euleriano:** um grafo tem um circuito Euleriano, caso exista um caminho Euleriano que comece e termine no mesmo ponto.

Tendo estes conceitos interiorizados, percebe-se que o objetivo do problema é encontrar o circuito Euleriano ótimo, que é obtido de forma gratuita através de um grafo Euleriano. Assim, se o grafo obtido fosse um grafo Euleriano, ou seja, um grafo onde todos os vértices têm grau par, era garantido encontrar um circuito Euleriano, considerado uma solução suficiente, pois atravessava todas as arestas. Seria também a solução ótima, uma vez que é necessário percorrer todas as arestas.

Contudo, ao analisar o grafo observa-se que existem vértices de grau ímpar, identificados na Figura 2, o que torna impossível encontrar um circuito Euleriano sem percorrer algumas arestas mais de uma vez. Deste modo, torna-se claro a necessidade de duplicar arestas, contudo essa duplicação tem de ser feita tendo em atenção os custos associados a cada aresta.

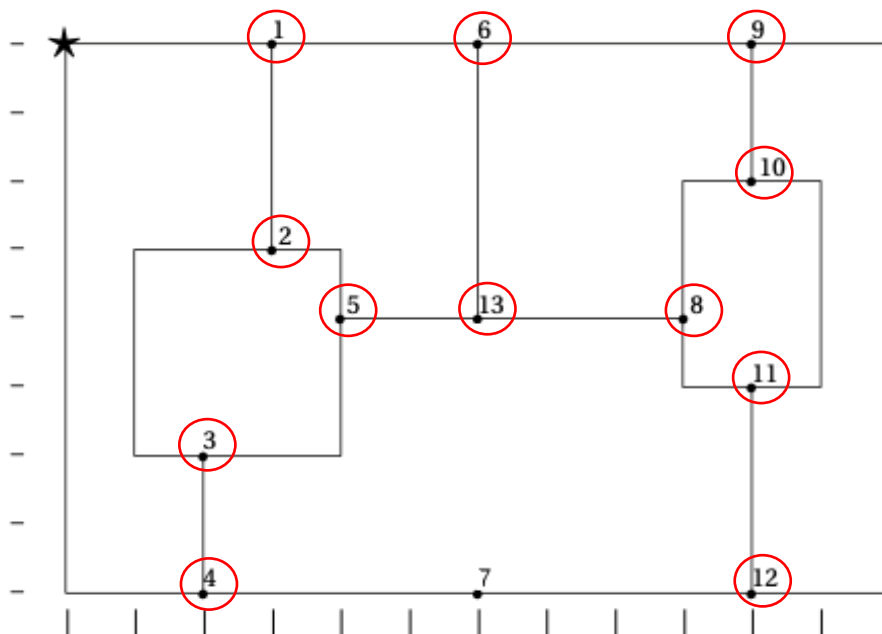


Figura 2: Vértices de grau ímpar no problema

Assim a solução para este problema passa por encontrar as arestas certas a acrescentar, de modo a todos os vértices com grau ímpar se transformem em vértices de grau par.

### 3. Formulação do problema

#### 3.1. Variáveis de decisão

As variáveis de decisão representam uma aresta entre dois pontos, o ponto  $i$  e o ponto  $j$ . Por isso, é necessário que exista uma variável de decisão por cada caminho possível. A nomenclatura adaptada pelo grupo seguiu as seguintes regras:

- As arestas têm o nome  $x_{ij}$  em que o  $i$  é o ponto onde essa aresta começa e  $j$  é o vértice onde essa aresta acaba;
- Cada variável pode ter apenas o valor 1 ou 0, ou seja, se essa aresta deve ser acrescentada ao grafo ou não.

$$\begin{cases} x_{ij} = 0, \text{ se a aresta } (i,j) \text{ não for acrescentada ao grafo} \\ x_{ij} = 1, \text{ se a aresta } (i,j) \text{ for acrescentada ao grafo} \end{cases}$$

*Equação 1: Variáveis de decisão*

Assim, por exemplo, se a aresta  $x_{1,2}$  for igual a zero, significa que a aresta que vai de 1 para 2 não deve ser duplicada no grafo existente.

#### 3.2. Restrições

Pretende-se que cada vértice do grafo com grau ímpar seja emparelhado com um e apenas um outro vértice de grau ímpar. Logo obtemos 12 restrições do tipo:

$$\text{Restrição } i : \left( \sum_{\text{vértices que chegam a } i} \right) = 1 \quad i \in \{1,2,3,4,5,6,8,9,10,11,12,13\}$$

*Equação 2: Função restrição*

Note-se que o vértice 7 não entra nas restrições uma vez que já é um vértice de grau par. Assim, por exemplo, para o nodo 1, temos que:

$$\textbf{Restrição1: } x_{2,1} + x_{3,1} + x_{4,1} + x_{5,1} + x_{6,1} + x_{8,1} + x_{9,1} + x_{10,1} + x_{11,1} + x_{12,1} + x_{13,1} = 1$$

Além da restrição anterior, é necessário ter em conta que as variáveis devem ser declaradas como binárias.

### 3.3. Função Objetivo

O objetivo é encontrar as arestas que ao inserir no grafo, torna todos os vértices com grau par.

Desta forma a função objetivo – função que permite determinar qual a melhor solução de um conjunto de soluções válidas – vai depender então da aresta e do custo dessas mesmas arestas. Para além disso, é necessário minimizar esta função de forma a obter a menor distância possível.

Podemos então definir a função objetivo como:

$$\min \sum x_{i,j} c_{i,j}, \quad (i,j) \in A$$

*Equação 3: Função objetivo*

- $x_{i,j}$  : valor – 0 ou 1 – associado a percorrer a aresta que parte de i e chega a j.
- $c_{i,j}$  : custo associado em percorrer a aresta  $x_{i,j}$ .
- A : conjunto de todas as arestas do grafo, com exceção das arestas que chegam ao vértice 7, que mais uma vez não entra no modelo por já se encontrar com grau par.

As distâncias consideradas em  $c_{i,j}$  são dadas na tabela seguinte.

		x	3	3	2	2	4	6	6	9	10	10	10	10	6
		y	8	5	2	0	4	8	0	4	8	6	3	0	4
			1	2	3	4	5	6	7	8	9	10	11	12	13
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
3	5	2	3,00	0,00	3,16	5,10	1,41	4,24	5,83	6,08	7,62	7,07	7,28	8,60	3,16
2	2	3	6,08	3,16	0,00	2,00	2,83	7,21	4,47	7,28	10,00	8,94	8,06	8,25	4,47
2	0	4	8,06	5,10	2,00	0,00	4,47	8,94	4,00	8,06	11,31	10,00	8,54	8,00	5,66
4	4	5	4,12	1,41	2,83	4,47	0,00	4,47	4,47	5,00	7,21	6,32	6,08	7,21	2,00
6	8	6	3,00	4,24	7,21	8,94	4,47	0,00	8,00	5,00	4,00	4,47	6,40	8,94	4,00
6	0	7	8,54	5,83	4,47	4,00	4,47	8,00	0,00	5,00	8,94	7,21	5,00	4,00	4,00
9	4	8	7,21	6,08	7,28	8,06	5,00	5,00	5,00	0,00	4,12	2,24	1,41	4,12	3,00
10	8	9	7,00	7,62	10,00	11,31	7,21	4,00	8,94	4,12	0,00	2,00	5,00	8,00	5,66
10	6	10	7,28	7,07	8,94	10,00	6,32	4,47	7,21	2,24	2,00	0,00	3,00	6,00	4,47
10	3	11	8,60	7,28	8,06	8,54	6,08	6,40	5,00	1,41	5,00	3,00	0,00	3,00	4,12
10	0	12	10,63	8,60	8,25	8,00	7,21	8,94	4,00	4,12	8,00	6,00	3,00	0,00	5,66
6	4	13	5,00	3,16	4,47	5,66	2,00	4,00	4,00	3,00	5,66	4,47	4,12	5,66	0,00

*Tabela 1: Distâncias associadas a cada aresta*

## 4. Texto Input

De modo a conseguir resolver o problema formulado anteriormente, é necessário transcrever para uma linguagem reconhecida pelo lp\_solve o modelo matemático formulado. De seguida, é apresentado o ficheiro de input definido de acordo com o modelo desenvolvido, obtido através do programa lp\_solve.

```
/* Função Objetivo */

min:
3.00 x2_1 + 6.08 x3_1 + 3.16 x3_2 +
8.06 x4_1 + 5.10 x4_2 + 2.00 x4_3 +
4.12 x5_1 + 1.41 x5_2 + 2.83 x5_3 + 4.47 x5_4 +
3.00 x6_1 + 4.24 x6_2 + 7.21 x6_3 + 8.94 x6_4 +
4.47 x6_5 + 7.21 x8_1 + 6.08 x8_2 + 7.28 x8_3 +
8.06 x8_4 + 5.00 x8_5 + 5.00 x8_6 + 7.00 x9_1 +
7.62 x9_2 + 10.00 x9_3 + 11.31 x9_4 + 7.21 x9_5 +
4.00 x9_6 + 4.12 x9_8 + 7.28 x10_1 + 7.07 x10_2 +
8.94 x10_3 + 10.00 x10_4 + 6.32 x10_5 + 4.47 x10_6 +
2.24 x10_8 + 2.00 x10_9 + 8.60 x11_1 + 7.28 x11_2 +
8.06 x11_3 + 8.54 x11_4 + 6.08 x11_5 + 6.40 x11_6 +
1.41 x11_8 + 5.00 x11_9 + 3.00 x11_10 + 10.63 x12_1 +
8.60 x12_2 + 8.25 x12_3 + 8.00 x12_4 + 7.21 x12_5 +
8.94 x12_6 + 4.12 x12_8 + 8.00 x12_9 + 6.00 x12_10 +
3.00 x12_11 + 5.00 x13_1 + 3.16 x13_2 + 4.47 x13_3 +
5.66 x13_4 + 2.00 x13_5 + 4.00 x13_6 + 3.00 x13_8 +
5.66 x13_9 + 4.47 x13_10 + 4.12 x13_11 + 5.66 x13_12;

/* Restrições */

v1 : x2_1 + x3_1 + x4_1 + x5_1 + x6_1 + x8_1 + x9_1 + x10_1 + x11_1 + x12_1 + x13_1 = 1;
v2 : x2_1 + x3_2 + x4_2 + x5_2 + x6_2 + x8_2 + x9_2 + x10_2 + x11_2 + x12_2 + x13_2 = 1;
v3 : x3_1 + x3_2 + x4_3 + x5_3 + x6_3 + x8_3 + x9_3 + x10_3 + x11_3 + x12_3 + x13_3 = 1;
v4 : x4_2 + x4_3 + x4_1 + x5_4 + x6_4 + x8_4 + x9_4 + x10_4 + x11_4 + x12_4 + x13_4 = 1;
v5 : x5_1 + x5_2 + x5_3 + x5_4 + x6_5 + x8_5 + x9_5 + x10_5 + x11_5 + x12_5 + x13_5 = 1;
v6 : x6_1 + x6_2 + x6_3 + x6_4 + x6_5 + x8_6 + x9_6 + x10_6 + x11_6 + x12_6 + x13_6 = 1;
v8 : x8_1 + x8_2 + x8_3 + x8_4 + x8_5 + x8_6 + x9_8 + x10_8 + x11_8 + x12_8 + x13_8 = 1;
v9 : x9_1 + x9_2 + x9_3 + x9_4 + x9_5 + x9_6 + x9_8 + x10_9 + x11_9 + x12_9 + x13_9 = 1;
v10 : x10_1 + x10_2 + x10_3 + x10_4 + x10_5 + x10_6 + x10_8 + x10_9 + x11_10 + x12_10 + x13_10 = 1;
v11 : x11_1 + x11_2 + x11_3 + x11_4 + x11_5 + x11_6 + x11_8 + x11_9 + x11_10 + x12_11 + x13_11 = 1;
v12 : x12_1 + x12_2 + x12_3 + x12_4 + x12_5 + x12_6 + x12_8 + x12_9 + x12_10 + x12_11 + x13_12 = 1;
v13 : x13_1 + x13_2 + x13_3 + x13_4 + x13_5 + x13_6 + x13_8 + x13_9 + x13_10 + x13_11 + x13_12 = 1;

/* Defenição das variaveis */

bin x2_1 x3_1 x3_2 x4_1 x4_2 x4_3 x5_1 x5_2 x5_3 x5_4 x6_1 x6_2 x6_3 x6_4 x6_5 x8_1
x8_2 x8_3 x8_4 x8_5 x8_6 x9_1 x9_2 x9_3 x9_4 x9_5 x9_6 x9_8 x10_1
x10_2 x10_3 x10_4 x10_5 x10_6 x10_8 x10_9 x11_1 x11_2 x11_3 x11_4
x11_5 x11_6 x11_8 x11_9 x11_10 x12_1 x12_2 x12_3 x12_4 x12_5 x12_6 x12_8
x12_9 x12_10 x12_11 x13_1 x13_2 x13_3 x13_4 x13_5 x13_6 x13_8 x13_9 x13_10 x13_11 x13_12;
```

Figura 3: Script que transcreve o modelo desenvolvido



## 5. Ficheiro Output

O lp\_solve, através do método Simplex, calculou o valor das variáveis de decisão que minimizam o custo das arestas a acrescentar no grafo. De seguida, apresenta-se o conjunto de soluções válidas.

Variables	MILP Feasible	result
	14,41	14,41
x4_3	1	1
x5_2	1	1
x6_1	1	1
x10_9	1	1
x12_11	1	1
x13_8	1	1
x2_1	0	0
x3_1	0	0
x3_2	0	0
x4_1	0	0
x4_2	0	0
x5_1	0	0
x5_3	0	0
x5_4	0	0
x6_2	0	0
x6_3	0	0
x6_4	0	0
x6_5	0	0
x8_1	0	0
x8_2	0	0
x8_3	0	0
x8_4	0	0
x8_5	0	0
x8_6	0	0
x9_1	0	0
x9_2	0	0
x9_3	0	0
x9_4	0	0
x9_5	0	0
x9_6	0	0
x9_8	0	0
x10_1	0	0
x10_2	0	0
x10_3	0	0
x10_4	0	0
x10_5	0	0
x10_6	0	0
x10_8	0	0
x11_1	0	0
x11_2	0	0
x11_3	0	0
x11_4	0	0
x11_5	0	0
x11_6	0	0
x11_8	0	0
x11_9	0	0
x11_10	0	0
x12_1	0	0
x12_2	0	0
x12_3	0	0
x12_4	0	0
x12_5	0	0
x12_6	0	0
x12_8	0	0
x12_9	0	0
x12_10	0	0
x13_1	0	0
x13_2	0	0
x13_3	0	0
x13_4	0	0
x13_5	0	0
x13_6	0	0
x13_9	0	0
x13_10	0	0
x13_11	0	0
x13_12	0	0

Figura 4: Output produzido pelo lp\_solve



## 7. Resultado do Problema

Uma vez que o grafo obtido depois da duplicação das arestas resultantes do modelo é um grafo Euleriano, percebe-se que existe mais do que um circuito Euleriano, sendo qualquer um deles um circuito válido e ótimo.

Um dos circuitos válidos e ótimos –  $x^*_1 \rightarrow x1\_2 \rightarrow x2\_3 \rightarrow x3\_4 \rightarrow x4\_3 \rightarrow x3\_5 \rightarrow x5\_2 \rightarrow x2\_5 \rightarrow x5\_13 \rightarrow x13\_6 \rightarrow x6\_1 \rightarrow x1\_6 \rightarrow x6\_9 \rightarrow x9\_10 \rightarrow x10\_8 \rightarrow x8\_13 \rightarrow x13\_8 \rightarrow x8\_11 \rightarrow x11\_12 \rightarrow x12\_11 \rightarrow x11\_10 \rightarrow x10\_9 \rightarrow x9\_12 \rightarrow x12\_7 \rightarrow x7\_4 \rightarrow x4\_*$  – está representado na Figura 6, onde o drone posicionado na estrela segue a seta número 1 e depois toma o rumo da seta número 2 e assim sucessivamente até finalmente retornar à estrela pela seta número 26.

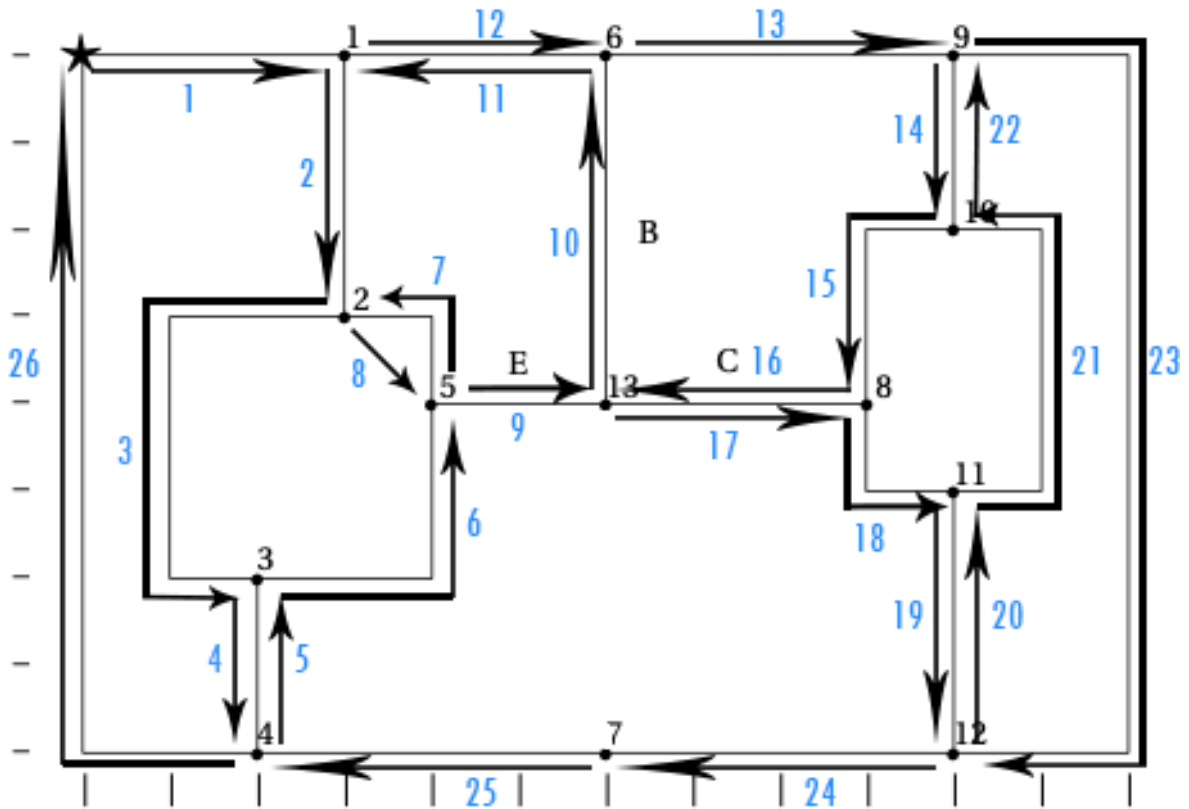


Figura 6: Representação da solução do problema

## 8. Validação do Modelo

Com o intuito de validar o modelo, primeiramente verificou-se se realmente o resultado obtido estava de acordo com o objetivo. Desta forma, ao analisar a Figura 5, que contém um esquema gráfico com o resultado obtido, percebe-se que todos os vértices têm grau par e desta forma obteve-se, como esperado, um grafo Euleriano. Faltava então perceber se os valores obtidos respeitavam as restrições, o tipo de variáveis e se o valor da função objetivo está correto.

### 8.1. Tipo de Variáveis

As variáveis têm de ser todas do tipo binário, 0 ou 1, pois o seu valor representa se uma aresta é adicionada ou não. De facto, após analisar os resultados obtidos, todos os resultados respeitavam esse requisito.

### 8.2. Função Objetivo

O resultado da função objetivo utilizada no modelo quando o valor das variáveis é manualmente substituindo pela solução ótima tem de coincidir com o resultado obtido no `lp_solve` – 14,41 – de modo a garantir que não houve nenhum tipo de erro ao definir o valor das distâncias no modelo. Assim:

$$\begin{aligned} & 3.00*0 + 6.08*0 + 3.16*0 + 8.06*0 + 5.10*0 + 2.00*1 + 4.12*0 + 1.41*1 + 2.83*0 + 4.47*0 + \\ & 3.00*1 + 4.24*0 + 7.21*0 + 8.94*0 + 4.47*0 + 7.21*0 + 6.08*0 + 7.28*0 + 8.06*0 + 5.00*0 + \\ & 5.00*0 + 7.00*0 + 7.62*0 + 10.00*0 + 11.31*0 + 7.21*0 + 4.00*0 + 4.12*0 + 7.28*0 + 7.07*0 + \\ & 8.94*0 + 10.00*0 + 6.32*0 + 4.47*0 + 2.24*0 + 2.00*1 + 8.60*0 + 7.28*0 + 8.06*0 + 8.54*0 + \\ & 6.08*0 + 6.40*0 + 1.41*0 + 5.00*0 + 3.00*0 + 10.63*0 + 8.60*0 + 8.25*0 + 8.00*0 + 7.21*0 + \\ & 8.94*0 + 4.12*0 + 8.00*0 + 6.00*0 + 3.00*1 + 5.00*0 + 3.16*0 + 4.47*0 + 5.66*0 + 2.00*0 + \\ & 4.00*0 + 3.00*1 + 5.66*0 + 4.47*0 + 4.12*0 + 5.66*0 = 14,41 \end{aligned}$$

### 8.3. Restrições

De modo a verificar que as restrições do problema foram cumpridas, substituiu-se os valores das variáveis obtidas e verificou-se que garantiam as igualdades lógicas, como se pode comprovar:

$$\text{Restrição 1: } x_{21} + x_{31} + x_{41} + x_{51} + x_{61} + x_{81} + x_{91} + x_{101} + x_{111} + x_{121} + x_{131} = 1$$

$$\Rightarrow 0 + 0 + 0 + 0 + 1 + 0 + 0 + 0 + 0 + 0 + 0 = 1$$

$$\Rightarrow 1 = 1$$

$$\text{Restrição 2: } x_{21} + x_{32} + x_{42} + x_{52} + x_{62} + x_{82} + x_{92} + x_{102} + x_{112} + x_{122} + x_{132} = 1$$

$$\Rightarrow 0 + 0 + 0 + 1 + 0 + 0 + 0 + 0 + 0 + 0 + 0 = 1$$

$$\Rightarrow 1 = 1$$

$$\text{Restrição 3: } x_{31} + x_{32} + x_{43} + x_{53} + x_{63} + x_{83} + x_{93} + x_{103} + x_{113} + x_{123} + x_{133} = 1$$

$$\Rightarrow 0 + 0 + 1 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 = 1$$

$$\Rightarrow 1 = 1$$

$$\text{Restrição 4: } x_{42} + x_{43} + x_{41} + x_{54} + x_{64} + x_{84} + x_{94} + x_{104} + x_{114} + x_{124} + x_{134} = 1$$

$$\Rightarrow 0 + 1 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 = 1$$

$$\Rightarrow 1 = 1$$

$$\text{Restrição 5: } x_{51} + x_{52} + x_{53} + x_{54} + x_{65} + x_{85} + x_{95} + x_{105} + x_{115} + x_{125} + x_{135} = 1$$

$$\Rightarrow 0 + 1 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 = 1$$

$$\Rightarrow 1 = 1$$

$$\text{Restrição 6: } x_{61} + x_{62} + x_{63} + x_{64} + x_{65} + x_{86} + x_{96} + x_{106} + x_{116} + x_{126} + x_{136} = 1$$

$$\Rightarrow 1 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 = 1$$

$$\Rightarrow 1 = 1$$

$$\text{Restrição 8: } x_{81} + x_{82} + x_{83} + x_{84} + x_{85} + x_{86} + x_{98} + x_{108} + x_{118} + x_{128} + x_{138} = 1$$

$$\Rightarrow 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 1 = 1$$

$$\Rightarrow 1 = 1$$

$$\text{Restrição 9: } x_{91} + x_{92} + x_{93} + x_{94} + x_{95} + x_{96} + x_{98} + x_{109} + x_{119} + x_{129} + x_{139} = 1$$

$$\Rightarrow 0 + 0 + 0 + 0 + 0 + 0 + 0 + 1 + 0 + 0 + 0 = 1$$

$$\Rightarrow 1 = 1$$

$$\text{Restrição 10: } x_{101} + x_{102} + x_{103} + x_{104} + x_{105} + x_{106} + x_{108} + x_{109} + x_{1110} + x_{1210} + x_{1310} = 1$$

$$\Rightarrow 0 + 0 + 0 + 0 + 0 + 0 + 0 + 1 + 0 + 0 + 0 = 1$$

$$\Rightarrow 1 = 1$$

$$\text{Restrição 11: } x_{111} + x_{112} + x_{113} + x_{114} + x_{115} + x_{116} + x_{118} + x_{119} + x_{110} + x_{1211} + x_{1311} = 1$$

$$\Rightarrow 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 1 + 0 = 1$$

$$\Rightarrow 1 = 1$$

$$\text{Restrição 12: } x_{121} + x_{122} + x_{123} + x_{124} + x_{125} + x_{126} + x_{128} + x_{129} + x_{1210} + x_{1211} + x_{1312} = 1$$

$$\Rightarrow 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 1 + 0 = 1$$

$$\Rightarrow 1 = 1$$

$$\text{Restrição 13: } x_{131} + x_{132} + x_{133} + x_{134} + x_{135} + x_{136} + x_{138} + x_{139} + x_{1310} + x_{1311} + x_{1312} = 1$$

$$\Rightarrow 0 + 0 + 0 + 0 + 0 + 0 + 1 + 0 + 0 + 0 + 0 = 1$$

$$\Rightarrow 1 = 1$$

## 9. Conclusão

Neste trabalho foi implementada uma solução para minimizar o percurso que um drone tem de realizar durante a inspeção de linhas de alta tensão, utilizando programação linear.

Através da utilização do método simplex, do `lp_solve`, obteve-se uma solução ótima onde o drone percorrerá uma distância de 95,41. As distâncias previamente conhecidas foram acrescidas de 14,41, correspondentes às arestas que se repetem.

Sendo a solução um grafo Euleriano, existem vários circuitos válidos e ótimos, como é o caso do percurso seguinte:

$x^*_1 \rightarrow x1\_2 \rightarrow x2\_3 \rightarrow x3\_4 \rightarrow x4\_3 \rightarrow x3\_5 \rightarrow x5\_2 \rightarrow x2\_5 \rightarrow x5\_13 \rightarrow x13\_6 \rightarrow x6\_1 \rightarrow x1\_6$   
 $\rightarrow x6\_9 \rightarrow x9\_10 \rightarrow x10\_8 \rightarrow x8\_13 \rightarrow x13\_8 \rightarrow x8\_11 \rightarrow x11\_12 \rightarrow x12\_11 \rightarrow x11\_10 \rightarrow x10\_9$   
 $\rightarrow x9\_12 \rightarrow x12\_7 \rightarrow x7\_4 \rightarrow x4\_*$