

Universidade do Minho
Departamento de Informática

Trabalho Prático 1

Investigação Operacional

24 de março, 2022



Beatriz
Rodrigues
(*a.93230*)



Francisco Neves
(*a.93202*)



Gabriela Prata
(*a.93288*)



Guilherme
Fernandes
(*a.93216*)

Índice

1	Formulação do Problema	3
2	Modelo	5
2.1	Variáveis de decisão	5
2.2	Parâmetros	5
2.3	Função objetivo	5
2.4	Restrições	6
2.5	Ficheiro de Input	6
3	Solução Ótima	7
4	Validação do Modelo	9

1. Formulação do Problema

Neste problema, estamos perante um veículo não tripulado (DRONE) que tem de inspecionar linhas de transporte de energia elétrica em alta tensão tendo por objetivo verificar se há vegetação a interferir com as linhas.

De forma a resolver este problema, pretendemos que o drone percorra a menor distância possível percorrendo todas as linhas pelo menos, uma vez.

É ainda de notar que as linhas de alta tensão podem ser percorridas em qualquer sentido e que para reposicionar o drone para recomençar a inspeção de uma nova linha não é necessário seguir as linhas de alta tensão, podendo, ao invés disso, fazer o percurso mais curto através do ar.

Tendo em conta as instruções fornecidas no enunciado do trabalho prático e o maior número mecanográfico do nosso grupo, 93288, foram removidas as arestas **C**, **D** e **E**, conforme representado na figura abaixo:

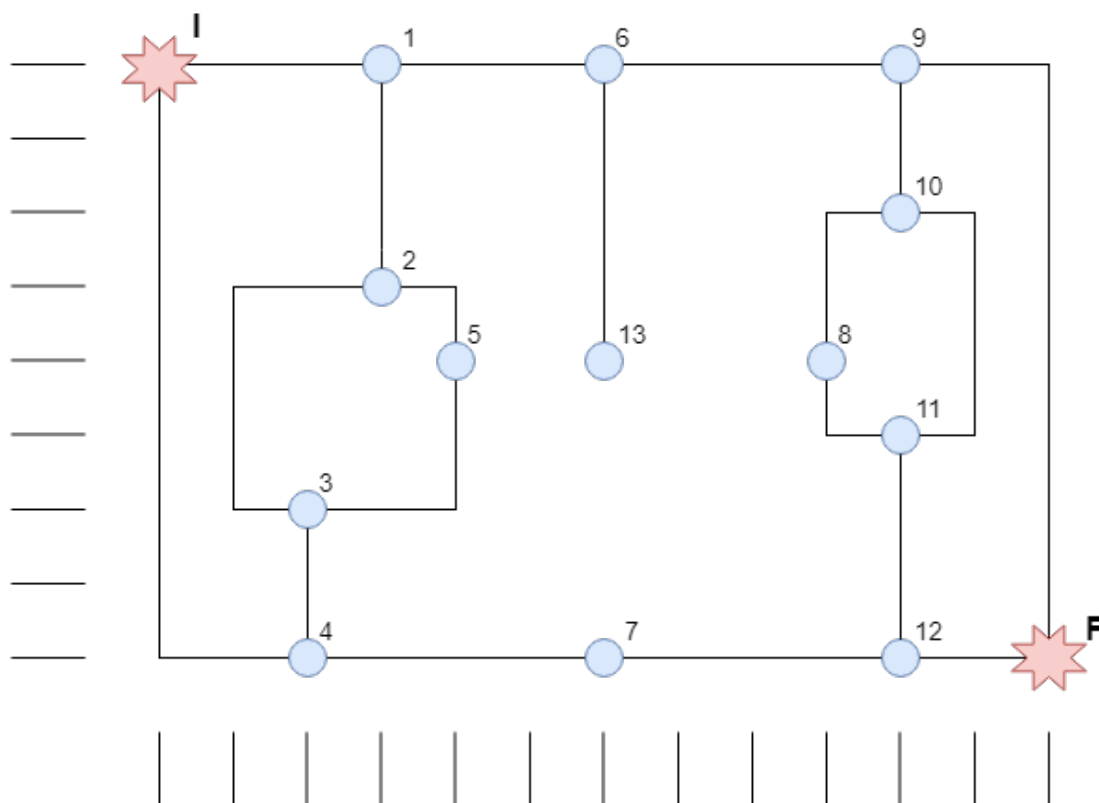


Figura 1.1: Grafo representativo do problema

Após analisarmos o problema e o grafo representativo das linhas de tensão, podemos compreender que estamos perante uma situação em que o nosso drone deverá percorrer um caminho Euleriano, isto é, um caminho no grafo em que todas as arestas são percorridas uma única vez e no qual está definido um ponto inicial e final do percurso diferentes. No entanto, para sermos capazes disso, tal como é referido pelo Teorema de Euler, teremos de trabalhar sobre um grafo em que todos os seus vértices são pares, ou seja, o número de arestas a que os vértices se encontram ligados é um número par ou um grafo em que existem no máximo dois vértices de grau ímpar (que será o nosso caso pretendido). De forma a cumprir este teorema, deveremos adicionar novas arestas ao mapa.

Temos então o nosso problema de otimização, ou seja, devemos ser capazes de responder à pergunta de quais serão as arestas passíveis de adição ao nosso grafo, de forma a tornar possível a construção de um caminho Euleriano para que, em seguida, seja minimizada a distância total percorrida. Deveremos ainda restringir a solução de forma a que não se adicione mais do que uma aresta a cada vértice.

A distância que pretendemos minimizar é dada pelo comprimento das arestas que iremos adicionar ao nosso grafo, pois, as arestas pré-existentes terão de ser obrigatoriamente percorridas de forma a darmos a resposta pretendida ao problema, pelo que, esse valor será obrigatoriamente constante.

Em seguida, apresenta-se os vértices ímpares, aqueles em que o número de arestas a que os vértices se encontram ligados é um número ímpar, coloridos a vermelho, do nosso grafo:

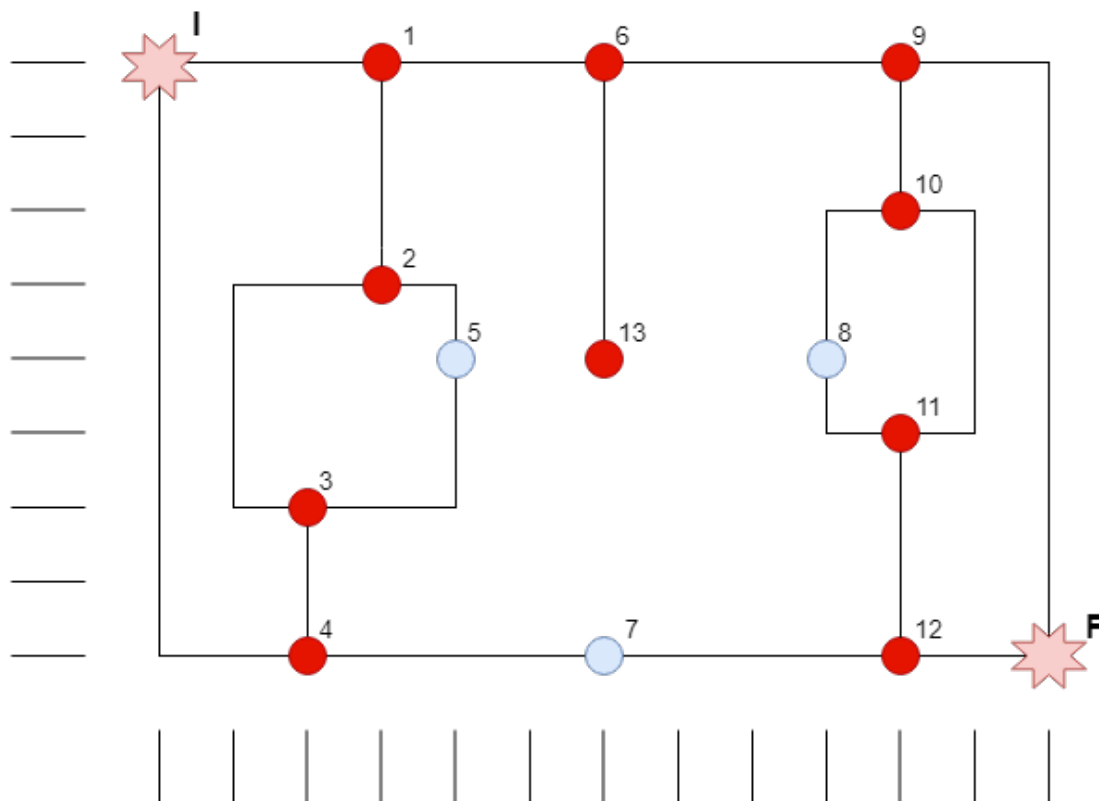


Figura 1.2: Vértices ímpares

2. Modelo

Seja V o conjunto de vértices aos quais pertencem todos os vértices de grau ímpar e os vértices inicial e final.

2.1 Variáveis de decisão

$x_{ij}; i, j \in V$: existe, ou não, uma aresta a unir os vértices i e j
 $x_{ij} \in \{0, 1\}$

Como se trata de um grafo não orientado, x_{ij} e x_{ji} representam a mesma aresta, logo decidimos representar as variáveis de decisão do modo $x_{ij} : i < j$

2.2 Parâmetros

d_{ij} : distância Euclidiana entre os vértices i e j que pode, ou não, coincidir com uma linha de alta tensão.

Estas distâncias estão descritas na seguinte tabela:

		x	0	3	3	2	2	4	6	6	9	10	10	10	6	12	
		y	8	8	5	2	0	4	8	0	4	8	6	3	0	4	
x	y		I	1	2	3	4	5	6	7	8	9	10	11	12	13	F
0	8	I	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 2.1: Tabela de distâncias Euclidianas

2.3 Função objetivo

O objetivo é minimizar o comprimento das arestas a adicionar.

$$\sum d_{ij} \times x_{ij}; i, j \in V, i < j$$

2.4 Restrições

Só deve ser adicionada uma aresta por vértice.

$$\forall i \in V : \sum x_{ij} = 1; j \in V, i < j$$

2.5 Ficheiro de Input

Tudo o referido previamente, resulta no seguinte ficheiro de input.

```
Source  Matix  Options  Result
1 /* Objective function */
2 min: 3.00 * xI_1 + 4.24 * xI_2 + 6.32 * xI_3 + 8.25 * xI_4 + 6.00 * xI_6 + 10.00 * xI_9 + 10.20 * xI_10 + 11.18 * xI_11 + 12.81 * xI_12 + 7.21 * xI_13 + 14.42 * xI_F
3       + 3.00 * xI_2 + 6.08 * xI_3 + 8.06 * xI_4 + 3.00 * xI_6 + 7.00 * xI_9 + 7.28 * xI_10 + 8.60 * xI_11 + 10.63 * xI_12 + 5.00 * xI_13 + 12.04 * xI_F
4       + 3.16 * xI_3 + 5.10 * xI_4 + 4.24 * xI_6 + 7.62 * xI_9 + 7.07 * xI_10 + 7.28 * xI_11 + 8.60 * xI_12 + 3.16 * xI_13 + 10.30 * xI_F
5       + 2.00 * xI_4 + 7.21 * xI_6 + 10.00 * xI_9 + 8.94 * xI_10 + 8.06 * xI_11 + 8.25 * xI_12 + 4.47 * xI_13 + 10.20 * xI_F
6       + 8.94 * xI_6 + 11.31 * xI_9 + 10.00 * xI_10 + 8.54 * xI_11 + 8.00 * xI_12 + 5.66 * xI_13 + 10.00 * xI_F
7       + 4.00 * xI_9 + 4.47 * xI_10 + 6.40 * xI_11 + 8.94 * xI_12 + 4.00 * xI_13 + 10.00 * xI_F
8       + 2.00 * xI_10 + 5.00 * xI_11 + 8.00 * xI_12 + 5.66 * xI_13 + 8.25 * xI_F
9       + 3.00 * xI_11 + 6.00 * xI_12 + 4.47 * xI_13 + 6.32 * xI_F
10
11
12
13 /* Variable bounds */
14 xI_1 + xI_2 + xI_3 + xI_4 + xI_6 + xI_9 + xI_10 + xI_11 + xI_12 + xI_13 + xI_F = 1;
15 xI_1 + xI_2 + xI_3 + xI_4 + xI_6 + xI_9 + xI_10 + xI_11 + xI_12 + xI_13 + xI_F = 1;
16 xI_2 + xI_2 + xI_2 + xI_2 + xI_2 + xI_2 + xI_2 + xI_2 + xI_2 + xI_2 + xI_2 = 1;
17 xI_3 + xI_3 + xI_3 + xI_3 + xI_3 + xI_3 + xI_3 + xI_3 + xI_3 + xI_3 + xI_3 = 1;
18 xI_4 + xI_4 + xI_4 + xI_4 + xI_4 + xI_4 + xI_4 + xI_4 + xI_4 + xI_4 + xI_4 = 1;
19 xI_6 + xI_6 + xI_6 + xI_6 + xI_6 + xI_6 + xI_6 + xI_6 + xI_6 + xI_6 + xI_6 = 1;
20 xI_9 + xI_9 + xI_9 + xI_9 + xI_9 + xI_9 + xI_9 + xI_9 + xI_9 + xI_9 + xI_9 = 1;
21 xI_10 + xI_10 + xI_10 + xI_10 + xI_10 + xI_10 + xI_10 + xI_10 + xI_10 + xI_10 + xI_10 = 1;
22 xI_11 + xI_11 + xI_11 + xI_11 + xI_11 + xI_11 + xI_11 + xI_11 + xI_11 + xI_11 + xI_11 = 1;
23 xI_12 + xI_12 + xI_12 + xI_12 + xI_12 + xI_12 + xI_12 + xI_12 + xI_12 + xI_12 + xI_12 = 1;
24 xI_13 + xI_13 + xI_13 + xI_13 + xI_13 + xI_13 + xI_13 + xI_13 + xI_13 + xI_13 + xI_13 = 1;
25 xI_F + xI_F + xI_F + xI_F + xI_F + xI_F + xI_F + xI_F + xI_F + xI_F + xI_F = 1;
26
27 bin xI_1, xI_2, xI_3, xI_4, xI_6, xI_9, xI_10, xI_11, xI_12, xI_13, xI_F,
28 xI_2, xI_3, xI_4, xI_6, xI_9, xI_10, xI_11, xI_12, xI_13, xI_F,
29 xI_2, xI_2, xI_2, xI_2, xI_2, xI_2, xI_2, xI_2, xI_2, xI_2, xI_2,
30 xI_3, xI_3, xI_3, xI_3, xI_3, xI_3, xI_3, xI_3, xI_3, xI_3, xI_3,
31 xI_4, xI_4, xI_4, xI_4, xI_4, xI_4, xI_4, xI_4, xI_4, xI_4, xI_4,
32 xI_6, xI_6, xI_6, xI_6, xI_6, xI_6, xI_6, xI_6, xI_6, xI_6, xI_6,
33 xI_9, xI_9, xI_9, xI_9, xI_9, xI_9, xI_9, xI_9, xI_9, xI_9, xI_9,
34 xI_10, xI_10, xI_10, xI_10, xI_10, xI_10, xI_10, xI_10, xI_10, xI_10, xI_10,
35 xI_11, xI_11, xI_11, xI_11, xI_11, xI_11, xI_11, xI_11, xI_11, xI_11, xI_11,
36 xI_12, xI_12, xI_12, xI_12, xI_12, xI_12, xI_12, xI_12, xI_12, xI_12, xI_12,
37 xI_13, xI_13, xI_13, xI_13, xI_13, xI_13, xI_13, xI_13, xI_13, xI_13, xI_13,
```

Figura 2.2: Ficheiro de Input

3. Solução Ótima

Utilizando a ferramenta *LPSolve*, obtivemos o seguinte resultado.

Variables	M...	result
	17,16	17,16
xI_1	1	1
x3_4	1	1
x12_F	1	1
x2_13	1	1
x6_9	1	1
x10_11	1	1

Figura 3.1: Output

Esta simboliza a adição de arestas ao problema, de forma a tornar todos os vértices ímpares em vértices pares e o vértice inicial e final em vértices ímpares (o que concretiza a definição de caminho euleriano). Para isso, foram adicionadas as arestas, coloridas a vermelho, conforme se pode observar na seguinte figura.

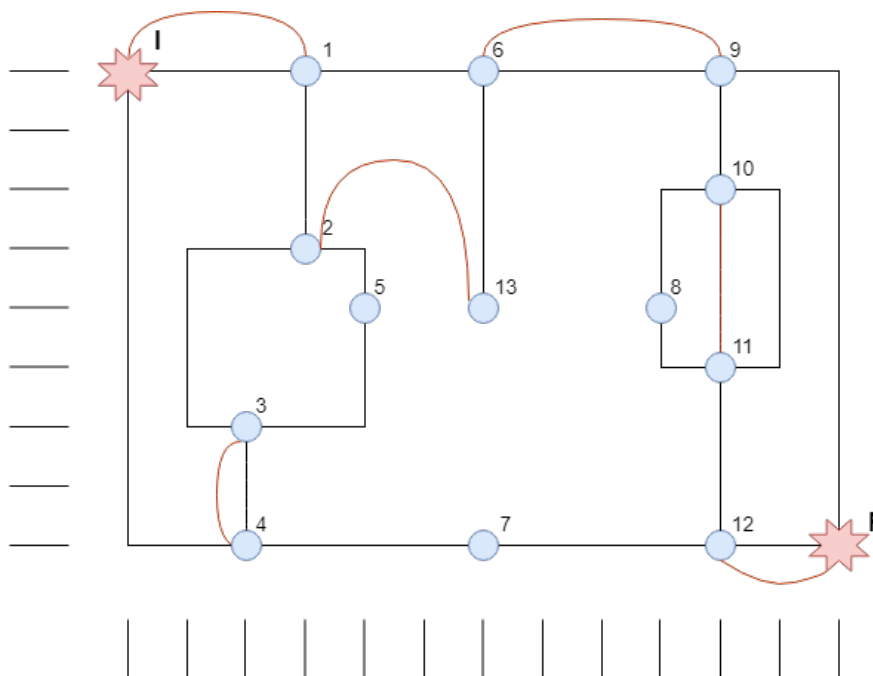


Figura 3.2: Novas arestas

Desta forma, podemos interpretar que algumas arestas serão percorridas mais que uma vez (por exemplo, a aresta I-1), mas também perceber que, por exemplo, na segunda vez que pretendermos transitar entre os vértices 10 e 11 poderemos fazê-lo por um caminho mais rápido que o caminho de alta tensão.

A partir do estabelecido previamente, podemos agora selecionar um caminho que se assume como sendo uma solução ótima para o problema. Qualquer caminho que selecionemos terá distância de $17.16 + 76 = 93.16$, visto que se trata da soma entre a distância das novas arestas que adicionamos com as distâncias de todas as linhas de alta tensão que devem ser percorridas. Um exemplo seria o representado na figura seguinte. Este corresponde à seguinte sequência de cores:

Verde → Roxo → Amarelo → Azul → Vermelho

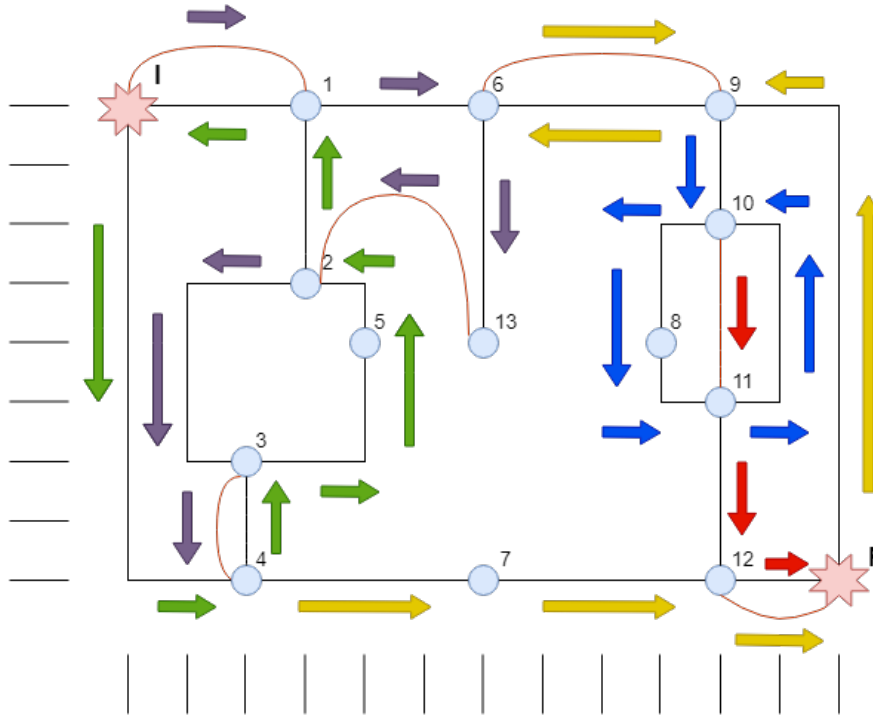


Figura 3.3: Caminho Ótimo

4. Validação do Modelo

Tendo calculado o resultado ótimo recorrendo ao *LP Solve*, é ainda necessário validar o mesmo, verificando se este satisfaz todas as condições inicialmente identificadas e, ainda, se este se adequa ao sistema real que pretendemos modelar.

Desta forma, podemos efetuar as seguintes verificações de forma a validar o nosso modelo.

1. É possível efetuar o caminho desde o nó **I** até ao nó **F** percorrendo todas as linhas de tensão;
2. Todos os vértices no grafo têm grau par, exceto dois (o inicial e o final), o que é consistente com a definição de caminho Euleriano;
3. Só foi adicionada uma aresta para cada vértice ímpar (e para os vértices iniciais e finais);
4. As distâncias percorridas foram sempre positivas;
5. Tendo em conta as restrições, o valor da distância total percorrida é o mínimo possível;
6. O valor da distância total percorrida pelo drone é admissível, uma vez que esta teria, obrigatoriamente, de ser dada pela soma entre a distância das arestas adicionadas com as distâncias de todas as arestas de alta tensão que devem ser percorridas;

De forma a verificarmos se seria possível obter uma solução melhor que a anteriormente apresentada, poderíamos ainda alterar as nossas restrições, passando, por exemplo, a verificar que mesmo que adicionássemos mais do que uma aresta por vértice, a solução ótima obtida seria igual. Para testarmos esta alteração, podemos voltar a correr novamente o modelo no *LP Solve*, trocando os '=' por '>='.

```

1 /* Objective function */
2 min: 3.00 * xI_1 + 4.24 * xI_2 + 6.32 * xI_3 + 8.25 * xI_4 + 6.00 * xI_6 + 10.00 * xI_9 + 10.20 * xI_10 + 11.18 * xI_11 + 12.81 * xI_12 + 7.21 * xI_13 + 14.42 * xI_F
3       + 3.00 * xI_2 + 6.08 * xI_3 + 8.06 * xI_4 + 3.00 * xI_6 + 7.00 * xI_9 + 7.28 * xI_10 + 8.60 * xI_11 + 10.63 * xI_12 + 5.00 * xI_13 + 12.04 * xI_F
4       + 3.16 * xI_3 + 5.10 * xI_4 + 4.24 * xI_6 + 7.62 * xI_9 + 7.07 * xI_10 + 7.28 * xI_11 + 8.60 * xI_12 + 3.16 * xI_13 + 10.30 * xI_F
5       + 2.00 * xI_4 + 7.21 * xI_6 + 10.00 * xI_9 + 8.94 * xI_10 + 8.06 * xI_11 + 8.25 * xI_12 + 4.47 * xI_13 + 10.20 * xI_F
6       + 8.94 * xI_6 + 11.31 * xI_9 + 10.00 * xI_10 + 8.54 * xI_11 + 8.00 * xI_12 + 5.66 * xI_13 + 10.00 * xI_F
7       + 4.00 * xI_9 + 4.47 * xI_10 + 6.40 * xI_11 + 8.94 * xI_12 + 4.00 * xI_13 + 10.00 * xI_F
8       + 2.00 * xI_10 + 5.00 * xI_11 + 8.00 * xI_12 + 5.66 * xI_13 + 8.25 * xI_F
9       + 3.00 * xI_11 + 6.00 * xI_12 + 4.47 * xI_13 + 6.32 * xI_F
10      + 3.00 * xI_12 + 4.12 * xI_13 + 3.61 * xI_F
11      + 5.66 * xI_13 + 2.00 * xI_F
12      + 7.21 * xI_F;
13 /* Variable bounds */
14 xI_1 + xI_2 + xI_3 + xI_4 + xI_6 + xI_9 + xI_10 + xI_11 + xI_12 + xI_13 + xI_F >= 1;
15 xI_1 + xI_2 + xI_3 + xI_4 + xI_6 + xI_9 + xI_10 + xI_11 + xI_12 + xI_13 + xI_F >= 1;
16 xI_2 + xI_2 + xI_3 + xI_4 + xI_6 + xI_9 + xI_10 + xI_11 + xI_12 + xI_13 + xI_F >= 1;
17 xI_3 + xI_3 + xI_4 + xI_6 + xI_9 + xI_10 + xI_11 + xI_12 + xI_13 + xI_F >= 1;
18 xI_4 + xI_4 + xI_6 + xI_9 + xI_10 + xI_11 + xI_12 + xI_13 + xI_F >= 1;
19 xI_6 + xI_6 + xI_9 + xI_10 + xI_11 + xI_12 + xI_13 + xI_F >= 1;
20 xI_9 + xI_9 + xI_10 + xI_11 + xI_12 + xI_13 + xI_F >= 1;
21 xI_10 + xI_10 + xI_11 + xI_12 + xI_13 + xI_F >= 1;
22 xI_11 + xI_11 + xI_12 + xI_13 + xI_F >= 1;
23 xI_12 + xI_12 + xI_13 + xI_F >= 1;
24 xI_13 + xI_13 + xI_F >= 1;
25 xI_F + xI_F >= 1;
26 bin xI_1, xI_2, xI_3, xI_4, xI_6, xI_9, xI_10, xI_11, xI_12, xI_13, xI_F;
27 xI_2, xI_3, xI_4, xI_6, xI_9, xI_10, xI_11, xI_12, xI_13, xI_F;
28 xI_3, xI_4, xI_6, xI_9, xI_10, xI_11, xI_12, xI_13, xI_F;
29 xI_4, xI_6, xI_9, xI_10, xI_11, xI_12, xI_13, xI_F;
30 xI_6, xI_9, xI_10, xI_11, xI_12, xI_13, xI_F;
31 xI_9, xI_10, xI_11, xI_12, xI_13, xI_F;
32 xI_10, xI_11, xI_12, xI_13, xI_F;
33 xI_11, xI_12, xI_13, xI_F;
34 xI_12, xI_13, xI_F;
35 xI_13, xI_F;
36 xI_F;
37

```

Figura 4.1: Alternativa

Variables	M...	result
	17,16	17,16
xI_1	1	1
x3_4	1	1
xI2_F	1	1
x2_13	1	1
x6_9	1	1
x10_11	1	1

Figura 4.2: Resultado Alternativo

Verificamos assim que, apesar disto não garantir que todos os vértices possuam grau par, o resultado do modelo é exatamente igual ao obtido anteriormente. Logo, podemos concluir que mesmo que fosse possível adicionar mais que uma aresta a cada vértice, a solução ideal continuaria a ser aquela em que cada vértice tem associado a si uma única aresta, ou seja, a restrição do modelo revela-se como sendo válida.