



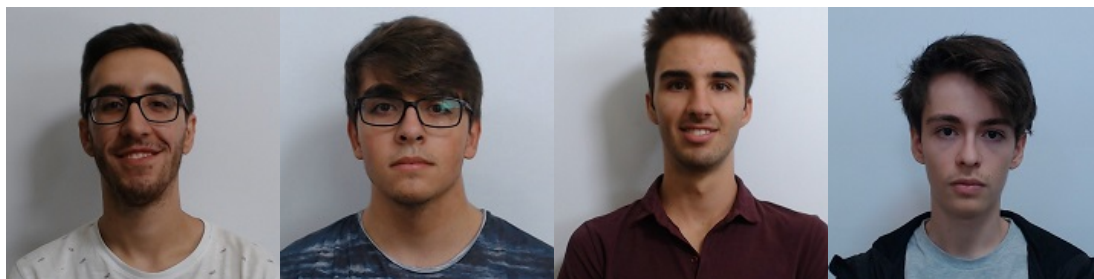
Universidade do Minho

13 de novembro de 2020

Relatório do trabalho prático 1

Métodos Determinísticos de Investigação Operacional

Mestrado Integrado em Engenharia Informática - 3º ano



Adriano Novo
Soto Maior
A89483

Bruno Pinto
Jácome
A89515

José Pedro
Ribeiro Peixoto
A89602

Afonso Trindade
Araújo de Pascoal Faria
A83920

Índice

1	Introdução	3
2	Resolução das questões	4
2.1	Indique o valor de ABCDE, e apresente a rede após a devida remoção das arestas	4
2.1.1	Valor de ABCDE	4
2.1.2	Remoção das arestas	4
2.1.3	Mapa de linhas resultante	4
2.2	Apresente a formulação deste problema	5
2.2.1	Contextualização	5
2.2.2	Regras	5
2.2.3	Considerações:	5
2.3	Apresente o modelo de programação linear	6
2.3.1	Variáveis de decisão	6
2.3.2	Parâmetros	6
2.3.3	Função Objetivo	7
2.3.4	Restrições	8
2.4	Apresente o ficheiro de input	9
2.4.1	droneMDIO.lp	9
2.5	Apresente o ficheiro de output produzido pelo programa	10
2.5.1	solutionMDIO	10
2.6	Interprete a solução ótima, apresente o percurso que o drone deve efectuar, e calcule a distância total percorrida	11
2.6.1	Solução ótima	11
2.6.2	Distância total percorrida	11
2.7	Descreva os procedimentos usados para validar o modelo	12
2.7.1	É solução admissível?	12
2.7.2	Exemplo de caminho ótimo	12
3	Anexos	13
3.1	Tabela das distâncias euclidianas	13

*

1 | Introdução

Este relatório foi realizado no âmbito da Unidade Curricular de Métodos Determinísticos em Investigação Operacional, envolvendo o conceito de otimização linear, utilizando *lpsolve*, um *software* para resolver, neste caso, problemas lineares. Tem como objetivos principais, abordar o raciocínio e mostrar uma solução para um problema de otimização sobre o percurso mínimo de um drone para a inspeção de linhas de alta tensão.

Este relatório divide-se essencialmente em 4 partes, todas relacionadas com a resolução do enunciado do trabalho prático:

Na primeira parte, identifica-se o mapa obtido pelo grupo de acordo com os requisitos do enunciado.

Já na segunda parte, formula-se o problema, identificando-se as variáveis de decisão, função objetivo e restrições.

Depois, analisa-se os resultados obtidos pela aplicação do *lpsolve* e apresenta-se a solução ótima.

Em último lugar, valida-se o modelo, exemplificando com um dos possíveis caminhos ótimos.

2 | Resolução das questões

2.1 Indique o valor de ABCDE, e apresente a rede após a devida remoção das arestas

2.1.1 Valor de ABCDE

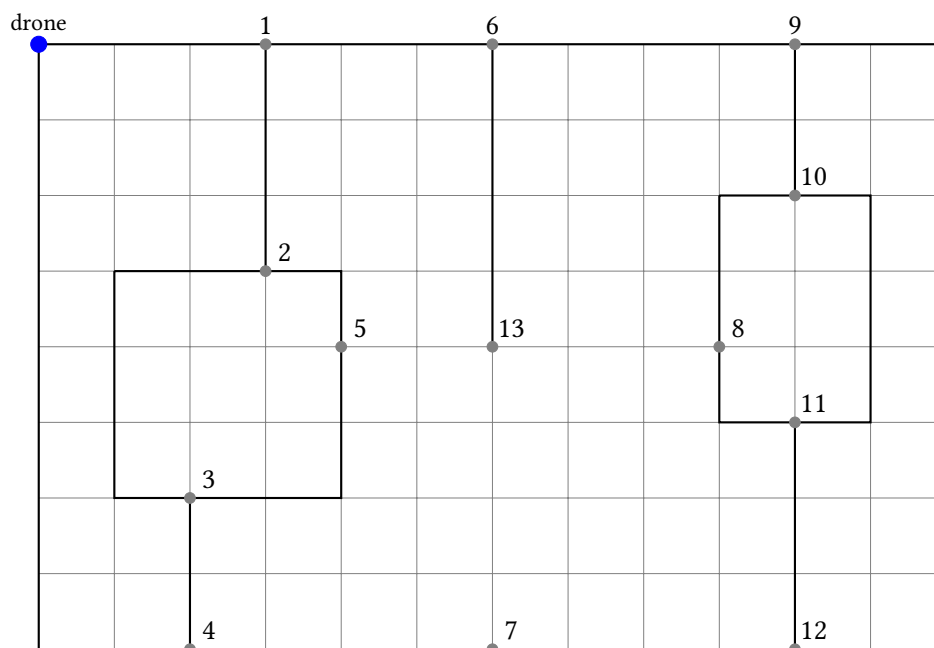
Comparando todos os números de aluno do grupo de trabalho (89483, 89515, 89602, 83920), verifica-se que o número maior é o 89602. Portanto, **A = 8, B = 9, C = 6, D = 0 e E = 2.**

2.1.2 Remoção das arestas

Seguindo as restrições indicadas no enunciado do trabalho, conclui-se que:

1. Como B é ímpar, **não se remove a aresta B.**
2. Como C é par, **remove-se a aresta C.**
3. Como D é par, **remove-se a aresta D.**
4. Como E é par, **remove-se a aresta E.**

2.1.3 Mapa de linhas resultante



2.2 Apresente a formulação deste problema

2.2.1 Contextualização

O problema de otimização consiste em minimizar a distância total percorrida por um **drone** com a função de inspecionar e verificar se há vegetação em linhas de alta tensão, representadas pelo mapa anterior.

Para isso, o drone **terá de percorrer todas arestas do mapa uma vez**. Além disso, poderá a qualquer momento percorrer um arco aéreo. Por exemplo, o arco $2 \longleftrightarrow 13$ representa um passagem aérea do nodo 2 ao nodo 13 (ou vice-versa). Estes arcos aéreos servirão para não só permitir ao drone viajar livremente entre nodos, como para que o caminho percorrido pelo drone seja um circuito (dado que este terá de terminar no local de onde partiu (0,0)). Uma vez que todas as arestas do mapa terão de ser percorridas uma vez e o drone terá de terminar onde começou, este é um circuito euleriano e, por isso, todos os nodos têm grau par.

2.2.2 Regras

1. Todas as arestas do mapa têm de ser percorridas uma vez.
2. O drone poderá passar num arco entre qualquer dois nodos, de forma aérea.
3. Todos os vértices têm grau par.
4. No fim, o drone terá de se encontrar no ponto de partida (0,0).

2.2.3 Considerações:

Arestas

Explicou-se, anteriormente, que o drone terá de percorrer todas as arestas uma vez e, sendo assim, não fará sentido que se fale do número de passagens por uma dada aresta. Desta forma, estas não são consideradas como variáveis de decisão.

Nota: a soma do comprimento de todas as arestas é **76 u.m.**.

Arcos aéreos

Percebe-se, seguindo o raciocínio anterior, que varia é, na verdade, os **trajetos extra** que o drone terá de fazer para que possa cumprir as regras em 2.2.2.

O número de passagens pelos arcos adicionais são, por isso, variáveis de decisão.

Distância adicional percorrida: soma do comprimento de cada arco aéreo percorrido num caminho.

Heurística: Minimizar a distância adicional percorrida.

Para isto, utilizou-se a ferramenta de heurística *lpsolve* e enunciou-se o problema no ficheiro **droneMDIO.lp**, gerando-se o ficheiro **output solutionMDIO**.

2.3 Apresente o modelo de programação linear

2.3.1 Variáveis de decisão

Considerações:

De forma a minimizar o número caracteres no nome das variáveis, utilizou-se base 16 (hexadecimal) para representar os nodos. Portanto, os nodos **10**, **11**, **12** e **13** são, respetivamente, **a**, **b**, **c** e **d**.

Número de passagens pelo arco aéreo entre os nodos i e j

$$v_{ij} \in \mathbb{N}_0, (i, j) \in A$$

$$A = \{(1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (1, 7), (1, 8), (1, 9), (1, a), (1, b), (1, c), (1, d), \\ (2, 3), (2, 4), (2, 5), (2, 6), (2, 7), (2, 8), (2, 9), (2, a), (2, b), (2, c), (2, d), \\ (3, 4), (3, 5), (3, 6), (3, 7), (3, 8), (3, 9), (3, a), (3, b), (3, c), (3, d), \\ (4, 5), (4, 6), (4, 7), (4, 8), (4, 9), (4, a), (4, b), (4, c), (4, d), \\ (5, 6), (5, 9), (5, a), (5, b), (5, c), (5, d), \\ (6, 7), (6, 8), (6, 9), (6, a), (6, b), (6, c), (6, d), \\ (7, 9), (7, a), (7, b), (7, c), (7, d), \\ (8, 9), (8, a), (8, b), (8, c), (8, d), \\ (9, a), (9, b), (9, c), (9, d), \\ (a, b), (a, c), (a, d), \\ (b, c), (b, d), \\ (c, d)\}$$

2.3.2 Parâmetros

Circuitos Eulerianos

Um Circuito Euleriano é um caminho num grafo que visita todas as arestas exatamente uma vez e que começa e termina no mesmo vértice.

Uma das principais condições para um grafo ser Euleriano é que todos os vértices precisam ser de grau par. Notemos que, qualquer grafo admissível deste problema é exatamente um circuito euleriano e, por isso, todos os nodos têm grau par.

Paridade dos nodos

Considerando que:

1. O drone começa e termina no mesmo vértice.
2. O drone tem de passar pelo menos uma vez em todos os nodos.

Então, todos os vértices do circuito do drone terão de ter grau par.

Dizer que um número x é par, é o mesmo que dizer $\exists k \in \mathbb{N}_0 : x = 2 \times k$.

Logo, como o grau de um vértice é dado pelo número de arcos a que está ligado, de modo a garantir a paridade do grau de cada um dos nodos (1..13), então cada nodo (1..13) associar-se-á a:

$$k_i \in \mathbb{N}_0, i = 1..13 \text{ (hexadecimal)}$$

Como apresentar-se-á nas Restrições, estes parâmetros são aplicados nas restrições **R1** a **Rd**.

2.3.3 Função Objetivo

Sejam:

- v_{ij} , o número de arcos (aéreos) entres os nodos i e j .
- d_{ij} , a distância euclidiana entre dois nodos i e j .

Então, a distância total percorrida num dado caminho é dada por:

$$\sum_{ij} v_{ij} d_{ij}, (i, j) \in A$$

Portanto, minimizar a distância total percorrida é:

$$\min : \sum_{ij} v_{ij} d_{ij}, (i, j) \in A$$

A partir da tabela de distâncias euclidianas nos anexos, enunciou-se a seguinte função objetivo:

```
/* Objective function */
min: 3*v12 + 6.08*v13 + 8.06*v14 + 4.12*v15 + 3*v16 + 8.54*v17 + 7.21*v18 + 7*v19
      + 7.28*v1a + 8.6*v1b + 10.63*v1c + 5*v1d +
      3.16*v23 + 5.10*v24 + 1.41*v25 + 4.24*v26 + 5.83*v27 + 6.08*v28 + 7.62*v29 +
      7.07*v2a + 7.28*v2b + 8.6*v2c + 3.16*v2d +
      2*v34 + 2.83*v35 + 7.21*v36 + 4.47*v37 + 7.28*v38 + 10*v39 + 8.94*v3a + 8.06*
      v3b + 8.25*v3c + 4.47*v3d
      4.47*v45 + 8.94*v46 + 4*v47 + 8.06*v48 + 11.31*v49 + 10*v4a + 8.54*v4b + 8*
      v4c + 5.66*v4d +
      4.47*v56 + 7.21*v59 + 6.32*v5a + 6.08*v5b + 7.21*v5c + 2*v5d
      8*v67 + 5*v68 + 4*v69 + 4.47*v6a + 6.40*v6b + 8.94*v6c + 4*v6d +
      8.94*v79 + 7.21*v7a + 5*v7b + 4*v7c + 4*v7d +
      4.12*v89 + 2.24*v8a + 1.41*v8b + 4.12*v8c + 3*v8d +
      2*v9a + 5*v9b + 8*v9c + 5.66*v9d +
      3*vab + 6*vac + 4.47*vad +
      3*vbc + 4.12*vbd +
      5.66*vcd;
```

2.3.4 Restrições

Paridade dos nodos

Como explicado em 2.3.2, todos os nodos num grafo admissível terão de ter grau par. Contudo, não esqueçamos que, apesar das arestas não serem contabilizadas na heurística, estas continuam a influenciar o grau de um nodo.

Por exemplo, em relação ao nodo 1: se apenas contabilizarmos as arestas do mapa, este nodo tem grau 3. Se agora, contabilizarmos tanto as arestas como os arcos aéreos, então o grau deste nodo passa a ser:

$$3 + \sum_i v_{1i}, i = 1..13 \text{ (hexadecimal)}$$

Além disso, é essencial referir que o drone **nunca passará de um nodo com grau par para outro nodo (com grau par)**. Porquê? Caso isso acontecesse esses dois nodos passariam a ter grau ímpar e o drone teria de, em ambos os nodos, arranjar um arco extra para outro nodo (de modo a tornar o grau desse nodo, novamente par). É óbvio que isso acrescentaria custo adicional, o que é indesejável.

Analisando o mapa, conclui-se que apenas os nodos 5, 7 e 8 são nodos com grau par e, portanto, **não foram admitidos os arcos compostos por cada par destes nodos**.

Restrições

De seguida, mostra-se as restrições escritas no *lpsolve*. A restrição R_i é a restrição sobre a paridade associada ao nodo $i, i = 1..13$ (hexadecimal).

```
//Todos os vértices têm grau par
R1: 3 + v12 + v13 + v14 + v15 + v16 + v17 + v18 + v19 + v1a + v1b + v1c + v1d =
    2*k1;
R2: 3 + v12 + v23 + v24 + v25 + v26 + v27 + v28 + v29 + v2a + v2b + v2c + v2d =
    2*k2;
R3: 3 + v31 + v32 + v34 + v35 + v36 + v37 + v38 + v39 + v3a + v3b + v3c + v3d =
    2*k3;
R4: 3 + v14 + v24 + v34 + v45 + v46 + v47 + v48 + v49 + v4a + v4b + v4c + v4d =
    2*k4;
R5: v15 + v25 + v35 + v45 + v56 + v59 + v5a + v5b + v5c + v5d = 2 * k5;
R6: 3 + v16 + v26 + v36 + v46 + v56 + v67 + v68 + v69 + v6a + v6b + v6c + v6d =
    2*k6;
R7: v17 + v27 + v37 + v47 + v67 + v79 + v7a + v7b + v7c + v7d = 2 * k7;
R8: v18 + v28 + v38 + v48 + v68 + v89 + v8a + v8b + v8c + v8d = 2 * k8;
R9: 3 + v19 + v29 + v39 + v49 + v59 + v69 + v79 + v89 + v9a + v9b + v9c + v9d =
    2*k9;
Ra: 3 + v1a + v2a + v3a + v4a + v5a + v6a + v7a + v8a + v9a + v9b + v9c + vad =
    2*k10;
Rb: 3 + v1b + v2b + v3b + v4b + v5b + v6b + v7b + v8b + v9b + vab + vbc + vbd =
    2*k11;
Rc: 3 + v1c + v2c + v3c + v4c + v5c + v6c + v7c + v8c + v9c + vac + vbc + vcd =
    2*k12;
Rd: 1 + v1d + v2d + v3d + v4d + v5d + v6d + v7d + v8d + v9d + vad + vbd + vcd =
    2*k13;
```


2.4 Apresente o ficheiro de input

2.4.1 droneMDIO.lp

```
/* Objective function */
min: 3*v12 + 6.08*v13 + 8.06*v14 + 4.12*v15 + 3*v16 + 8.54*v17 + 7.21*v18 +
7*v19 + 7.28*v1a + 8.6*v1b + 10.63*v1c + 5*v1d +
3.16*v23 + 5.10*v24 + 1.41*v25 + 4.24*v26 + 5.83*v27 + 6.08*v28 + 7.62*
v29 + 7.07*v2a + 7.28*v2b + 8.6*v2c + 3.16*v2d +
2*v34 + 2.83*v35 + 7.21*v36 + 4.47*v37 + 7.28*v38 + 10*v39 + 8.94*v3a +
8.06*v3b + 8.25*v3c + 4.47*v3d
4.47*v45 + 8.94*v46 + 4*v47 + 8.06*v48 + 11.31*v49 + 10*v4a + 8.54*v4b +
8*v4c + 5.66*v4d +
4.47*v56 + 7.21*v59 + 6.32*v5a + 6.08*v5b + 7.21*v5c + 2*v5d
8*v67 + 5*v68 + 4*v69 + 4.47*v6a + 6.40*v6b + 8.94*v6c + 4*v6d +
8.94*v79 + 7.21*v7a + 5*v7b + 4*v7c + 4*v7d +
4.12*v89 + 2.24*v8a + 1.41*v8b + 4.12*v8c + 3*v8d +
2*v9a + 5*v9b + 8*v9c + 5.66*v9d +
3*vab + 6*vac + 4.47*vad +
3*vbc + 4.12*vbd +
5.66*vcd;

//Todos os vértices têm grau par
R1: 3 + v12 + v13 + v14 + v15 + v16 + v17 + v18 + v19 + v1a + v1b + v1c +
v1d = 2*k1;
R2: 3 + v12 + v23 + v24 + v25 + v26 + v27 + v28 + v29 + v2a + v2b + v2c +
v2d = 2*k2;
R3: 3 + v31 + v32 + v34 + v35 + v36 + v37 + v38 + v39 + v3a + v3b + v3c +
v3d = 2*k3;
R4: 3 + v14 + v24 + v34 + v45 + v46 + v47 + v48 + v49 + v4a + v4b + v4c +
v4d = 2*k4;
R5: v15 + v25 + v35 + v45 + v56 + v59 + v5a + v5b + v5c + v5d = 2 * k5;
R6: 3 + v16 + v26 + v36 + v46 + v56 + v67 + v68 + v69 + v6a + v6b + v6c +
v6d = 2*k6;
R7: v17 + v27 + v37 + v47 + v67 + v79 + v7a + v7b + v7c + v7d = 2 * k7;
R8: v18 + v28 + v38 + v48 + v68 + v89 + v8a + v8b + v8c + v8d = 2 * k8;
R9: 3 + v19 + v29 + v39 + v49 + v59 + v69 + v79 + v89 + v9a + v9b + v9c +
v9d = 2*k9;
Ra: 3 + v1a + v2a + v3a + v4a + v5a + v6a + v7a + v8a + v9a + v9b + v9c +
vad = 2*k10;
Rb: 3 + v1b + v2b + v3b + v4b + v5b + v6b + v7b + v8b + v9b + vab + vbc +
vbd = 2*k11;
Rc: 3 + v1c + v2c + v3c + v4c + v5c + v6c + v7c + v8c + v9c + vac + vbc +
vcd = 2*k12;
Rd: 1 + v1d + v2d + v3d + v4d + v5d + v6d + v7d + v8d + v9d + vad + vbd +
vcd = 2*k13;

int k1,k2,k3,k4,k5,k6,k7,k8,k9,k10,k11,k12,k13;
int v12,v13,v14,v15,v16,v17,v18,v19,v1a,v1b,v1c,v1d,v23,v24,v25,v26,v27,v28,
v29,v2a,v2b,v2c,v2d,v34,v35,v36,v37,v38,v39,v3a,v3b,v3c,v3d,v45,v46,v48,
v49,v4a,v4b,v4c,v4d,v56,v67,v68,v69,v6a,v6b,v6c,v6d,v59,v79,v89,v9a,v9b,
v9c,v9d,v5a,v7a,v8a,vab,vac,vad,v5b,v7b,v8b,vbc,vbd,v5c,v7c,v8c,vcd,v5d,
v7d,v8d;
```

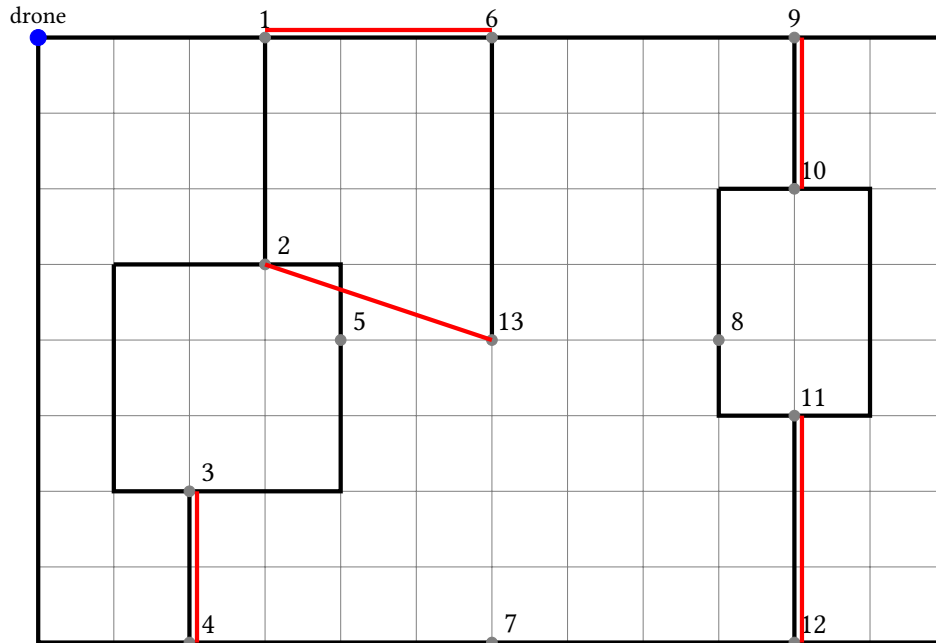
2.5 Apresente o ficheiro de output produzido pelo programa

2.5.1 solutionMDIO

Value of objective function:		v59	0
13.16000000		v5a	0
		v5b	0
Actual values of the variables:		v5c	0
v12	0	v5d	0
v13	0	v67	0
v14	0	v68	0
v15	0	v69	0
v16	1	v6a	0
v17	0	v6b	0
v18	0	v6c	0
v19	0	v6d	0
v1a	0	v79	0
v1b	0	v7a	0
v1c	0	v7b	0
v1d	0	v7c	0
v23	0	v7d	0
v24	0	v89	0
v25	0	v8a	0
v26	0	v8b	0
v27	0	v8c	0
v28	0	v8d	0
v29	0	v9a	1
v2a	0	v9b	0
v2b	0	v9c	0
v2c	0	v9d	0
v2d	1	vab	0
v34	1	vac	0
v35	0	vad	0
v36	0	vbc	1
v37	0	vbd	0
v38	0	vcd	0
v39	0	k1	2
v3a	0	k2	2
v3b	0	v31	0
v3c	0	v32	0
v3d	0	k3	2
v45	0	k4	2
v46	0	k5	0
v47	0	k6	2
v48	0	k7	0
v49	0	k8	0
v4a	0	k9	2
v4b	0	k10	2
v4c	0	k11	2
v4d	0	k12	2
v56	0	k13	1

2.6 Interprete a solução óptima, apresente o percurso que o drone deve efectuar, e calcule a distância total percorrida

2.6.1 Solução óptima



Representação do percurso óptimo

Uma vez que o drone passa uma vez em cada uma das arestas (cabos de alta tensão), as arestas ficaram representadas da mesma forma que em 2.1.3.

Representou-se a vermelho os arcos adicionais, mais concretamente: (1,6), (3,4), (9,10), (11,12) e (2,d) (i.e. (2,13)).

2.6.2 Distância total percorrida

Ao analisar o ficheiro *output* do *lpsolve*, verifica-se que a distância adicional percorrida ótima é **13.16 u.m.**. Portanto, a **distância total percorrida** é $76 + 13.16 = 89.16$ u.m..

Value of objective function: 13.16000000

2.7 Descreva os procedimentos usados para validar o modelo

2.7.1 É solução admissível?

$$\mathbf{R1:} \ 3 + 0 + 0 + 0 + 0 + 0 + 1 + 0 + 0 + 0 + 0 + 0 + 0 + 0 = 2 * 2 \checkmark$$

$$\mathbf{R2:} \ 3 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 1 = 2 * 2 \checkmark$$

$$\mathbf{R3:} \ 3 + 0 + 0 + 1 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 = 2 * 2 \checkmark$$

$$\mathbf{R4:} \ 3 + 0 + 1 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 = 2 * 2 \checkmark$$

$$\mathbf{R5:} \ 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 = 2 * 0 \checkmark$$

$$\mathbf{R6:} \ 3 + 1 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 = 2 * 2 \checkmark$$

$$\mathbf{R7:} \ 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 = 2 * 0 \checkmark$$

$$\mathbf{R8:} \ 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 = 2 * 0 \checkmark$$

$$\mathbf{R9:} \ 3 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 1 + 0 + 0 + 0 = 2 * 2 \checkmark$$

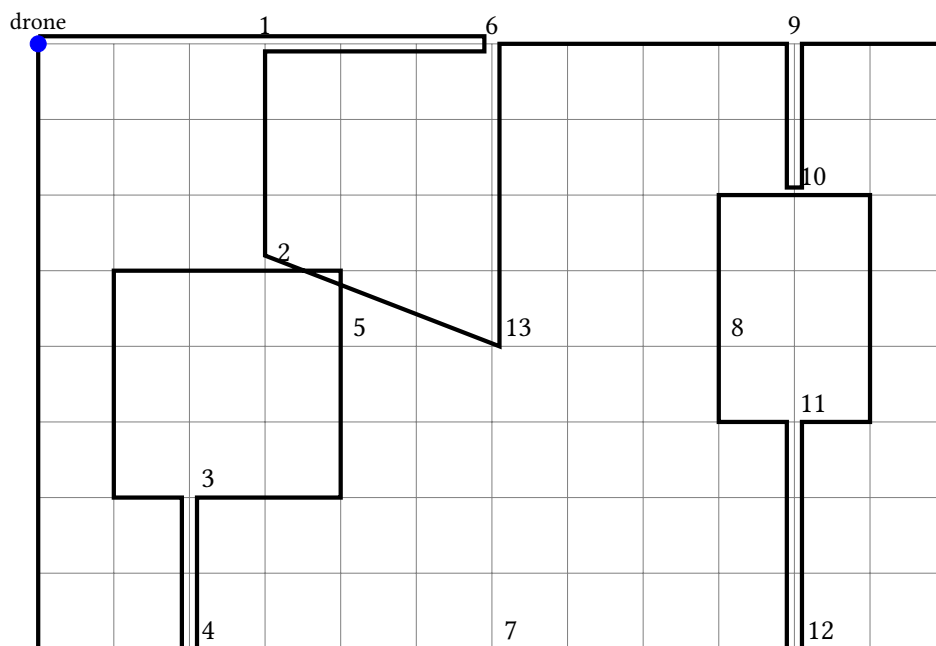
$$\mathbf{Ra:} \ 3 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 1 + 0 + 0 + 0 = 2 * 2 \checkmark$$

$$\mathbf{Rb:} \ 3 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 1 + 0 = 2 * 2 \checkmark$$

$$\mathbf{Rc:} \ 3 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 1 = 2 * 2 \checkmark$$

$$\mathbf{Rd:} \ 1 + 0 + 1 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 = 2 * 1 \checkmark$$

2.7.2 Exemplo de caminho ótimo



3 | Anexos

3.1 Tabela das distâncias euclidianas

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
		Nodo	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