

UNIVERSIDADE DO MINHO

DEPARTAMENTO DE PRODUÇÃO E SISTEMAS

**Modelos Determinísticos de Investigação
Operacional**
Trabalho 1

Benjamim Coelho, Henrique Neto, Leonardo Marreiros e Paulo Pereira

e-mail: {a89616,a89618,a89537,a86475}@alunos.uminho.pt

9 de outubro 2020

Introdução

O problema apresentado ocorre quando um veículo não tripulado tem de inspeccionar linhas de transporte de energia eléctrica em alta tensão para verificar se há vegetação a interferir com as linhas. Em termos concretos, trata-se de um problema do tipo "Carteiro Chinês".

O problema do Carteiro Chinês caracteriza-se pelo atravessamento de arcos e tem como objetivo a cobertura de arcos de um grafo, criando uma rota que passe pelo menos uma vez em cada um destes arcos. Além disso pretende-se o caminho mais curto e, neste caso, as arestas podem ser percorridas em qualquer sentido, e pode ser necessário atravessar a mesma aresta mais de uma vez.

Sabendo que a cada aresta está associado uma distância, o problema consiste em determinar o caminho que o drone deve percorrer para minimizar a distância total utilizando programação linear.

Problema

De forma a diferenciar as soluções dos diferentes grupos, foi implementado um método para remoção de arestas do grafo. Seja ABCDE o número de inscrição do estudante do grupo com maior número de inscrição:

1. se B par, remover a aresta B;
2. se C par, remover a aresta C;
3. se D par, remover a aresta D;
4. se E par, remover a aresta E;

O maior número de inscrição entre os elementos do grupo é 89618, pelo que removemos as arestas C e E, o que resultou no seguinte grafo:

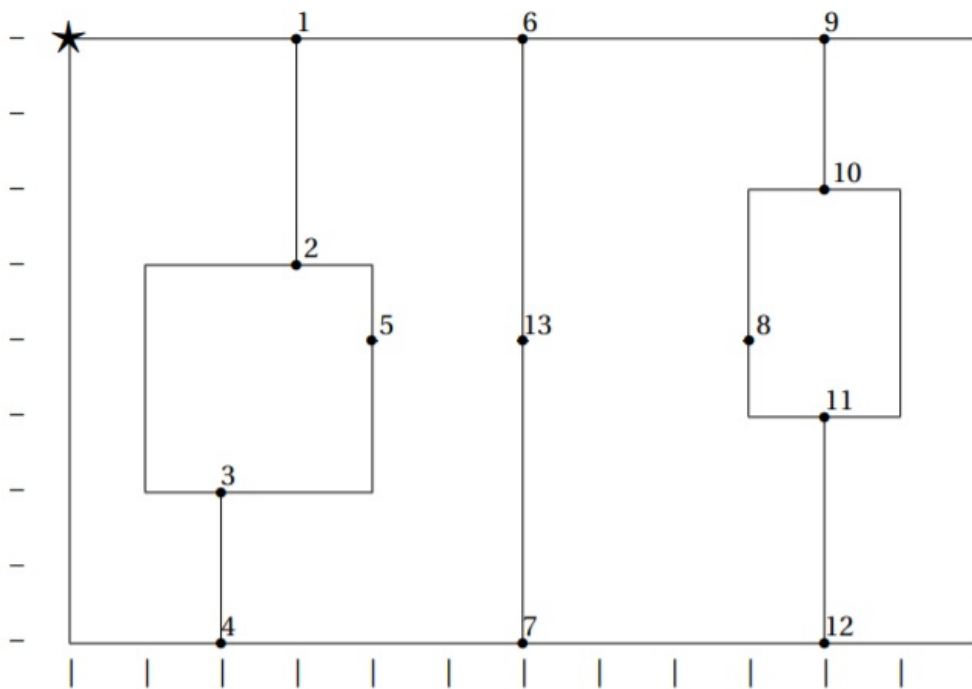


Figura 2.1: Grafo do problema

Formulação do problema

Tal como dito anteriormente, o problema consiste no cálculo do caminho mais curto de um drone que inspecciona linhas de transporte de energia eléctrica em alta tensão. Pretende-se que o drone verifique todas as linhas (arestas). No entanto, não está restringido a sobrevoar apenas estas linhas, podendo reposicionar-se pelo ar e em qualquer sentido. Além disso temos como dados as distâncias entre cada um dos vértices. É de notar que um grafo é Euleriano se e só se o grafo for ligado e o grau de todos os vértices for par, o que não acontece neste caso. Isto levou-nos a criar uma função objetivo cuja finalidade é encontrar os emparelhamentos em falta, isto é, selecção das arestas adicionais a percorrer atendendo ao seu custo. O resultado disto foi um grafo com todos os vértices de grau par.

3.1 Variáveis de decisão

As variáveis de decisão representam as arestas adicionais que irão ser percorridas pelo ar. Por isso, é necessário que exista uma variável de decisão por cada ligação possível entre vértices. Deste modo, as variáveis têm a forma x_{ij} onde i e j representam as extremidades da aresta representada. Naturalmente, a aresta x_{ij} corresponde à aresta x_{ji} . Assim, adiciona-se apenas uma das opções para evitar redundâncias.

Note-se que as arestas percorridas sobre os fios não são contabilizadas visto que, no contexto deste problema, estas serão sempre percorridas independentemente da solução admissível permanecendo sempre como constantes no modelo.

3.2 Nós

O circuito Euleriano, para cada aresta de entrada no vértice, deve ter uma aresta de saída de maneira a que todas as arestas possam ser percorridas sem repetições.

3.2.1 Nós Pares

Nos nós de grau par, o número de arestas de entrada é igual ao número de arestas de saída pelo que estes vértices foram ignorados no modelo de programação uma vez que não constituem um problema na decisão do circuito Euleriano de menor custo. O conjunto de vértices de grau par no nosso problema é $V_p = [v_5, v_8, v_{13}]$.

3.2.2 Nós Ímpares

Em relação aos nós de grau ímpar a précondição já não se mantém. Sendo assim, o objetivo foi encontrar os emparelhamentos ótimos tendo em consideração as distâncias de forma a tornar todos os vértices de grau par.

Modelo de Programação Linear

4.1 Função Objetivo

Tendo em conta que se pretende percorrer a menor distancia possível, logicamente a função objetivo consistirá na minimização dos custos dos emparelhamentos adicionais. A função objetivo é do tipo $z = c_1X_1 + c_2X_2 + \dots + c_nX_n$ onde os coeficientes c_1, c_2, \dots, c_n são os coeficientes de custos/distâncias (que são dados conhecidos) e X_1, X_2, \dots, X_n representam as variáveis de decisão que serão determinadas. Neste caso, representam os pares de vértices que constituem uma possível aresta, sem colocar em causa a paridade dos vértices já pares. Por exemplo, a aresta x_{1_13} não é considerada pois, caso fosse feita esta ligação, o vértice 13 passaria a ser de grau ímpar, o que é exatamente o contrário daquilo que pretendemos.

4.2 Restrições

Com base no problema formulado, concluímos que cada restrição representaria um vértice de grau ímpar e desta forma asseguraria a paridade do mesmo no modelo final. Posteriormente estas restrições teriam de assegurar que uma e uma só aresta se formaria entre dois vértices e que cada vértice teria apenas uma aresta nova. Sendo assim, com base nestas necessidades, introduzimos para cada vértice de grau ímpar i , uma restrição do tipo:

$$\sum x_{ji} + \sum x_{ij} = 1 \quad (4.1)$$

onde no primeiro somatório j percorre todos os vértices de grau ímpar menores que i e no segundo somatório j percorre todos os vértices de grau ímpar maiores que i . Por exemplo a restrição do vértice $i = 7$ seria:

$$[x_{1_7} + x_{2_7} + x_{3_7} + x_{4_7} + x_{6_7}] + [x_{7_9} + x_{7_10} + x_{7_11} + x_{7_12}] = 1 \quad (4.2)$$

Modelação no LPSolve

5.1 Ficheiro de Input

```
/* Função Objetivo */
min: 3.00 x1_2 + 6.08 x1_3 + 8.06 x1_4 + 3.00 x1_6 + 8.54 x1_7 + 7.00 x1_9 +
      7.28 x1_10 + 8.60 x1_11 + 10.63 x1_12 + 3.16 x2_3 + 5.10 x2_4 + 4.24 x2_6 +
      5.83 x2_7 + 7.62 x2_9 + 7.07 x2_10 + 7.28 x2_11 + 8.60 x2_12 + 2.00 x3_4 +
      7.21 x3_6 + 4.47 x3_7 + 10.00 x3_9 + 8.94 x3_10 + 8.06 x3_11 + 8.25 x3_12 +
      8.94 x4_6 + 4.00 x4_7 + 11.31 x4_9 + 10.00 x4_10 + 8.54 x4_11 + 8.00 x4_12 +
      8.00 x6_7 + 4.00 x6_9 + 4.47 x6_10 + 6.40 x6_11 + 8.94 x6_12 + 8.94 x7_9 +
      7.21 x7_10 + 5.00 x7_11 + 4.00 x7_12 + 2.00 x9_10 + 5.00 x9_11 + 8.00 x9_12 +
      3.00 x10_11 + 6.00 x10_12 + 3.00 x11_12;

/* Restrições */
vertice1: x1_2 + x1_3 + x1_4 + x1_6 + x1_7 + x1_9 + x1_10 + x1_11 + x1_12 = 1;
vertice2: x1_2 + x2_3 + x2_4 + x2_6 + x2_7 + x2_9 + x2_10 + x2_11 + x2_12 = 1;
vertice3: x1_3 + x2_3 + x3_4 + x3_6 + x3_7 + x3_9 + x3_10 + x3_11 + x3_12 = 1;
vertice4: x1_4 + x2_4 + x3_4 + x4_6 + x4_7 + x4_9 + x4_10 + x4_11 + x4_12 = 1;
/* Vertice 5 tem grau par */
vertice6: x1_6 + x2_6 + x3_6 + x4_6 + x6_7 + x6_9 + x6_10 + x6_11 + x6_12 = 1;
vertice7: x1_7 + x2_7 + x3_7 + x4_7 + x6_7 + x7_9 + x7_10 + x7_11 + x7_12 = 1;
/* Vertice 8 tem grau par */
vertice9: x1_9 + x2_9 + x3_9 + x4_9 + x6_9 + x7_9 + x9_10 + x9_11 + x9_12 = 1;
vertice10: x1_10 + x2_10 + x3_10 + x4_10 + x6_10 + x7_10 + x9_10 + x10_11 + x10_12 = 1;
vertice11: x1_11 + x2_11 + x3_11 + x4_11 + x6_11 + x7_11 + x9_11 + x10_11 + x11_12 = 1;
vertice12: x1_12 + x2_12 + x3_12 + x4_12 + x6_12 + x7_12 + x9_12 + x10_12 + x11_12 = 1;
/* Vertice 13 tem grau par */

bin x1_2, x1_3, x1_4, x1_6, x1_7, x1_9, x1_10, x1_11, x1_12, x2_3, x2_4, x2_6, x2_7,
    x2_9, x2_10, x2_11, x2_12, x3_4, x3_6, x3_7, x3_9, x3_10, x3_11, x3_12, x4_6, x4_7,
    x4_9, x4_10, x4_11, x4_12, x6_7, x6_9, x6_10, x6_11, x6_12, x7_9, x7_10, x7_11,
    x7_12, x9_10, x9_11, x9_12, x10_11, x10_12, x11_12;
```

5.2 Ficheiro de Output

5.2.1 Registo

```
Model name: 'LPSolver' - run #1
Objective: Minimize(R0)
SUBMITTED
Model size:      10 constraints,      45 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      15.12 after      10 iter is B&B base.
Feasible solution     16 after       11 iter,      1 nodes (gap 5.5%)
```

Improved solution	15.16 after	12 iter,	2 nodes (gap 0.2%)
Optimal solution	15.16 after	12 iter,	2 nodes (gap 0.2%).

Relative numeric accuracy ||*|| = 1.11022e-016

MEMO: lp_solve version 5.5.2.5 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 0.002 seconds, presolve used 0.005 seconds,
 ... 0.016 seconds in simplex solver, in total 0.023 seconds.

5.2.2 Resultado

Variables	MILP Feasible	MILP Better	Result
Objective function	16	15,16	15,16
x1_2	1	0	0
x1_3	0	0	0
x1_4	0	0	0
x1_6	0	1	1
x1_7	0	0	0
x1_9	0	0	0
x1_10	0	0	0
x1_11	0	0	0
x1_12	0	0	0
x2_3	0	1	1
x2_4	0	0	0
x2_6	0	0	0
x2_7	0	0	0
x2_9	0	0	0
x2_10	0	0	0
x2_11	0	0	0
x2_12	0	0	0
x3_4	1	0	0
x3_6	0	0	0
x3_7	0	0	0
x3_9	0	0	0
x3_10	0	0	0
x3_11	0	0	0
x3_12	0	0	0

Variables	MILP Feasible	MILP Better	Result
x4_6	0	0	0
x4_7	0	1	1
x4_9	0	0	0
x4_10	0	0	0
x4_11	0	0	0
x4_12	0	0	0
x6_7	0	0	0
x6_9	1	0	0
x6_10	0	0	0
x6_11	0	0	0
x6_12	0	0	0
x7_9	0	0	0
x7_10	0	0	0
x7_11	0	0	0
x7_12	1	0	0
x9_10	0	1	1
x9_11	0	0	0
x9_12	0	0	0
x10_11	1	0	0
x10_12	0	0	0
x11_12	0	1	1

Interpretação do Resultado

Como podemos observar, as variáveis binárias x_{1_6} , x_{2_3} , x_{4_7} , x_{9_10} e x_{11_12} obtiveram resultado 1, o que significa que entre estes vértices vamos ter mais uma aresta que representa um caminho que o drone pode percorrer pelo ar.

O valor da função objetivo é 15,16. Este representa o valor mínimo da soma das distancias adicionais, provenientes das arestas cuja variável tem valor 1. Na figura 6.1 podemos observar o grafo ótimo com as novas arestas (representadas a verde) .

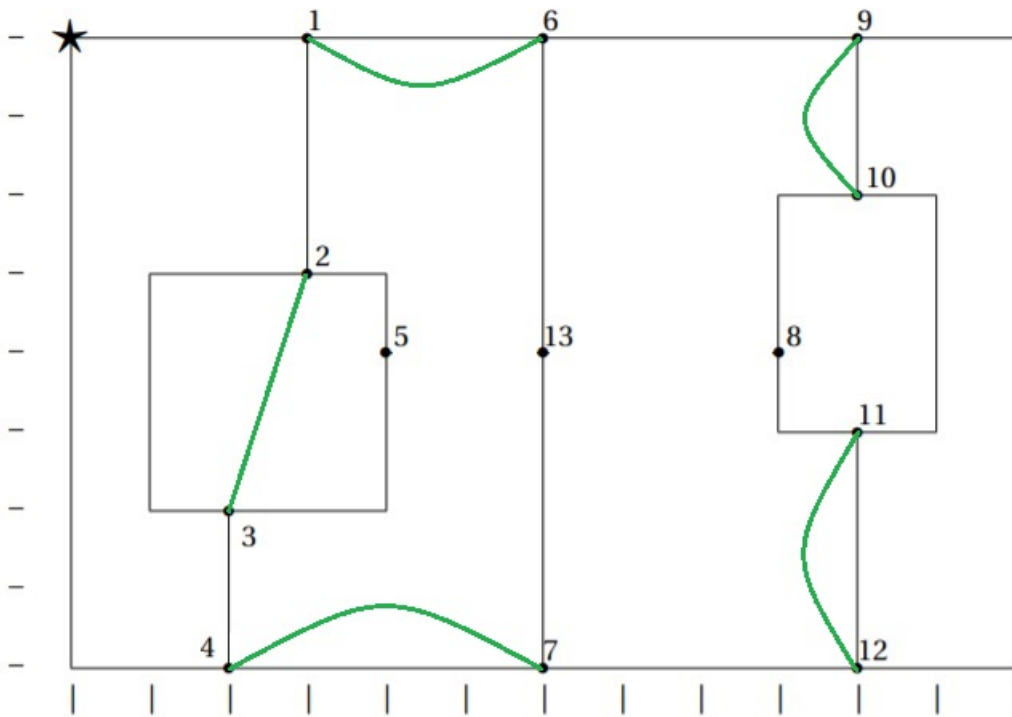


Figura 6.1: Grafo Ótimo

Ora, uma vez que agora que todos os vértices têm grau par, passamos a estar perante um grafo Euleriano, o que significa que existe pelo menos um caminho em que o drone percorre todas as arestas do grafo se repetir. Cada um destes caminhos Eulerianos representa uma solução ótima para este problema e, devido a sua natureza, cada um possuirá o mesma distância total.

Na figura 6.2 estão exemplificadas 3 soluções ótimas completamente diferentes ou, se considerarmos os caminhos inversos, estão presentes 6 soluções ótimas.

Validação do modelo

7.1 Função objetivo

Substituindo na função objetivo os valores das variáveis pelos resultados obtidos, a solução deve coincidir com a solução ótima obtida.

$$\begin{aligned} & 3.00 \times x1_2 + 6.08 \times x1_3 + 8.06 \times x1_4 + 3.00 \times x1_6 + 8.54 \times x1_7 + 7.00 \times x1_9 + 7.28 \times x1_10 \\ & + 8.60 \times x1_11 + 10.63 \times x1_12 + 3.16 \times x2_3 + 5.10 \times x2_4 + 4.24 \times x2_6 + 5.83 \times x2_7 + 7.62 \times x2_9 \\ & + 7.07 \times x2_10 + 7.28 \times x2_11 + 8.60 \times x2_12 + 2.00 \times x3_4 + 7.21 \times x3_6 + 4.47 \times x3_7 + 10.00 \times x3_9 \\ & + 8.94 \times x3_10 + 8.06 \times x3_11 + 8.25 \times x3_12 + 8.94 \times x4_6 + 4.00 \times x4_7 + 11.31 \times x4_9 + 10.00 \times x4_10 \\ & + 8.54 \times x4_11 + 8.00 \times x4_12 + 8.00 \times x6_7 + 4.00 \times x6_9 + 4.47 \times x6_10 + 6.40 \times x6_11 + 8.94 \times x6_12 \\ & + 8.94 \times x7_9 + 7.21 \times x7_10 + 5.00 \times x7_11 + 4.00 \times x7_12 + 2.00 \times x9_10 + 5.00 \times x9_11 + 8.00 \times x9_12 \\ & + 3.00 \times x10_11 + 6.00 \times x10_12 + 3.00 \times x11_12 \quad (7.1) \end{aligned}$$

Substituindo os valores das variáveis de decisão em 7.1 temos

$$\begin{aligned} & 3.00 \times 0 + 6.08 \times 0 + 8.06 \times 0 + 3.00 \times 1 + 8.54 \times 0 + 7.00 \times 0 + 7.28 \times 0 \\ & + 8.60 \times 0 + 10.63 \times 0 + 3.16 \times 1 + 5.10 \times 0 + 4.24 \times 0 + 5.83 \times 0 + 7.62 \times 0 \\ & + 7.07 \times 0 + 7.28 \times 0 + 8.60 \times 0 + 2.00 \times 0 + 7.21 \times 0 + 4.47 \times 0 + 10.00 \times 0 \\ & + 8.94 \times 0 + 8.06 \times 0 + 8.25 \times 0 + 8.94 \times 0 + 4.00 \times 1 + 11.31 \times 0 + 10.00 \times 0 \\ & + 8.54 \times 0 + 8.00 \times 0 + 8.00 \times 0 + 4.00 \times 0 + 4.47 \times 0 + 6.40 \times 0 + 8.94 \times 0 \\ & + 8.94 \times 0 + 7.21 \times 0 + 5.00 \times 0 + 4.00 \times 0 + 2.00 \times 1 + 5.00 \times 0 + 8.00 \times 0 \\ & + 3.00 \times 0 + 6.00 \times 0 + 3.00 \times 1 = 15.16 \quad (7.2) \end{aligned}$$

pelo que a solução coincide com a solução ótima obtida.

7.2 Restrições

$$\begin{aligned} V_1 : 1 &= x1_2 + x1_3 + x1_4 + x1_6 + x1_7 + x1_9 + x1_10 + x1_11 + x1_12 \\ 1 &= 0 + 0 + 0 + 1 + 0 + 0 + 0 + 0 + 0 \\ 1 &= 1 \end{aligned} \quad (7.3)$$

$$\begin{aligned} V_2 : 1 &= x1_2 + x2_3 + x2_4 + x2_6 + x2_7 + x2_9 + x2_10 + x2_11 + x2_12 \\ 1 &= 0 + 1 + 0 + 0 + 0 + 0 + 0 + 0 + 0 \\ 1 &= 1 \end{aligned} \quad (7.4)$$

$$\begin{aligned} V_3 : 1 &= x1_3 + x2_3 + x3_4 + x3_6 + x3_7 + x3_9 + x3_10 + x3_11 + x3_12 \\ 1 &= 0 + 1 + 0 + 0 + 0 + 0 + 0 + 0 + 0 \\ 1 &= 1 \end{aligned} \quad (7.5)$$

$$\begin{aligned}
\mathbf{V}_4 : 1 &= x1_4 + x2_4 + x3_4 + x4_6 + x4_7 + x4_9 + x4_10 + x4_11 + x4_12 \\
1 &= 0 + 0 + 0 + 0 + 1 + 0 + 0 + 0 + 0 \\
1 &= 1
\end{aligned} \tag{7.6}$$

$$\begin{aligned}
\mathbf{V}_6 : 1 &= x1_6 + x2_6 + x3_6 + x4_6 + x6_7 + x6_9 + x6_10 + x6_11 + x6_12 \\
1 &= 1 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 \\
1 &= 1
\end{aligned} \tag{7.7}$$

$$\begin{aligned}
\mathbf{V}_7 : 1 &= x1_7 + x2_7 + x3_7 + x4_7 + x6_7 + x7_9 + x7_10 + x7_11 + x7_12 \\
1 &= 0 + 0 + 0 + 1 + 0 + 0 + 0 + 0 + 0 \\
1 &= 1
\end{aligned} \tag{7.8}$$

$$\begin{aligned}
\mathbf{V}_9 : 1 &= x1_9 + x2_9 + x3_9 + x4_9 + x6_9 + x7_9 + x9_10 + x9_11 + x9_12 \\
1 &= 0 + 0 + 0 + 0 + 0 + 0 + 1 + 0 + 0 \\
1 &= 1
\end{aligned} \tag{7.9}$$

$$\begin{aligned}
\mathbf{V}_{10} : 1 &= x1_10 + x2_10 + x3_10 + x4_10 + x6_10 + x7_10 + x9_10 + x10_11 + x10_12 \\
1 &= 0 + 0 + 0 + 0 + 0 + 0 + 1 + 0 + 0 \\
1 &= 1
\end{aligned} \tag{7.10}$$

$$\begin{aligned}
\mathbf{V}_{11} : 1 &= x1_11 + x2_11 + x3_11 + x4_11 + x6_11 + x7_11 + x9_11 + x10_11 + x11_12 \\
1 &= 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 1 \\
1 &= 1
\end{aligned} \tag{7.11}$$

$$\begin{aligned}
\mathbf{V}_{12} : 1 &= x1_12 + x2_12 + x3_12 + x4_12 + x6_12 + x7_12 + x9_12 + x10_12 + x11_12 \\
1 &= 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 1 \\
1 &= 1
\end{aligned} \tag{7.12}$$

Conclusão

Para concluir, de forma a provar que, qualquer que seja o circuito escolhido que incorpore os novos emparelhamentos, a distância total é a mesma, vamos substituir os valores das variáveis das equações (6.1), (6.2) e (6.3) pelas distâncias correspondentes.

i. (6.1)

$$10 + 4 + 4 + 12 + 2 + 5 + 3 + \mathbf{3} + 2 + 3 + \mathbf{2} + 4 + 4 + 4 + \mathbf{4} + 2 + 4 + 2 + \mathbf{3.16} + 6 + 3 + \mathbf{3} + 3 + 3 = 95.16$$

ii. (6.2)

$$10 + 2 + 6 + \mathbf{3.16} + 4 + 2 + 3 + \mathbf{3} + 4 + 4 + 4 + \mathbf{4} + 4 + 3 + 2 + 3 + 2 + \mathbf{2} + 5 + \mathbf{3} + 12 + 10 = 95.16$$

iii. (6.3)

$$3 + 3 + 2 + 4 + \mathbf{3.16} + 6 + 2 + 4 + 4 + 4 + 3 + \mathbf{3} + 4 + 2 + \mathbf{2} + 12 + 3 + 5 + 3 + 2 + \mathbf{3} + 4 + \mathbf{4} + 10 = 95.16$$

É de notar que este resultado fica em conformidade com o que foi referido anteriormente, na medida em que existe uma distância mínima constante, equivalente à soma das distâncias de todas as arestas do grafo inicial que é somada à distância das arestas adicionais. É possível verificar isto somando a distância das arestas a negrito (que correspondem aos novos emparelhamentos) cujo resultado é 15.16 em cada uma das equações, tal como tínhamos previsto.

Concluindo, com este trabalho implementamos uma solução para o problema do Carteiro Chinês fazendo uso da programação linear. A solução ótima obtida corresponde a uma distância total de 95.16 centímetros e tem como exemplo de percurso o circuito:

$$\begin{aligned} \star \rightarrow x4_7 \rightarrow x7_12 \rightarrow x12_9 \rightarrow x9_10 \rightarrow x10_11 \rightarrow x11_12 \rightarrow x12_11 \rightarrow x11_8 \rightarrow x8_10 \rightarrow \\ x10_9 \rightarrow x9_6 \rightarrow x6_3 \rightarrow x3_7 \rightarrow x7_4 \rightarrow x4_3 \rightarrow x3_5 \rightarrow x5_2 \rightarrow x2_3(e)^1 \rightarrow x3_2 \rightarrow \\ x2_1 \rightarrow x1_6 \rightarrow x6_1 \rightarrow \star \end{aligned} \tag{8.1}$$

¹Distância euclidiana.