

UNIVERSIDADE DO MINHO

MESTRADO INTEGRADO EM ENGENHARIA INFORMÁTICA

MODELOS DETERMINÍSTICOS DE INVESTIGAÇÃO OPERACIONAL

Fluxo de Redes

Etienne Costa (a76089)
João Coutinho (a86272)
Margarida Faria (a71924)
Rui Azevedo (a80789)

11 de Outubro de 2019

Conteúdo

| | | |
|----------|-------------------------------------|-----------|
| 1 | Introdução | 3 |
| 2 | Desenvolvimento do Modelo | 4 |
| 2.1 | Definição do problema | 4 |
| 2.2 | Análise do problema | 5 |
| 2.3 | Construção do modelo | 6 |
| 2.4 | Solução do modelo | 6 |
| 2.4.1 | Modelo matemático | 7 |
| 2.4.2 | Variáveis de decisão | 7 |
| 2.4.3 | Função objectivo | 7 |
| 2.4.4 | Restrições | 7 |
| 2.5 | Modelo final | 8 |
| 2.5.1 | Ficheiro de input | 8 |
| 2.5.2 | Ficheiro de <i>output</i> | 10 |
| 2.6 | Validação do modelo | 10 |
| 3 | Discussão | 14 |
| 4 | Conclusão | 15 |

Lista de Figuras

| | | |
|---|---|----|
| 1 | Mapa da cidade | 4 |
| 2 | Sub-Mapa da cidade | 5 |
| 3 | Mapa com o conjunto de soluções | 10 |
| 4 | Circuito 1 e Circuito 2 | 11 |
| 5 | Circuito 3 | 12 |
| 6 | Circuito 4 e Circuito 5 | 13 |
| 7 | Sub-Mapa da cidade | 14 |

Lista de Tabelas

| | | |
|---|----------------------------|----|
| 1 | Tabela de custos | 13 |
|---|----------------------------|----|

1 Introdução

A realização deste trabalho prático ocorre no âmbito na unidade curricular Modelos Determinísticos de Investigação Operacional e visa desenvolver a capacidade de analisar sistemas complexos, de criar modelos para os descrever, de obter soluções para esses modelos utilizando programas computacionais adequados validando e interpretando as soluções obtidas de modo a elaborar recomendações para o sistema em análise. Neste caso em concreto, pretende-se modelar um conjunto de circuitos em que todos os arcos de um grafo são percorridos, pelo menos uma vez, minimizando a distância total percorrida.

2 Desenvolvimento do Modelo

Vários problemas práticos em Investigação Operacional podem ser expressos como problemas de programação linear e este problema não foge à regra. Sendo assim, optou-se por sintetizar o problema em cinco pontos distintos.

2.1 Definição do problema

Um problema comum no âmbito da Investigação Operacional é o fluxo de redes. É um cenário corrente em empresas de distribuição que pretendem entregar produtos aos seus clientes de modo a minimizar os custos da viagem, garantindo que todos os seus clientes recebem as suas encomendas.

O trabalho a desenvolver trata-se de um problema de fluxo de redes onde existe um camião do lixo que parte de um ponto inicial e tem que percorrer todas as ruas da cidade de modo a recolher todo o lixo das ruas acabando o trajecto no ponto de partida. É de ressaltar que o camião deverá passar em todas as ruas pelo menos uma vez e que estas têm um comprimento inteiro, proporcional à dimensão do traço em centímetros.

A baixo é apresentado o mapa da cidade a estudar. Note-se que todas as ruas são de sentido único, sendo muito provável que o camião tenha que passar mais do que uma vez na mesma rua, de maneira a conseguir voltar sempre ao ponto inicial (vértice marcado com a estrela).

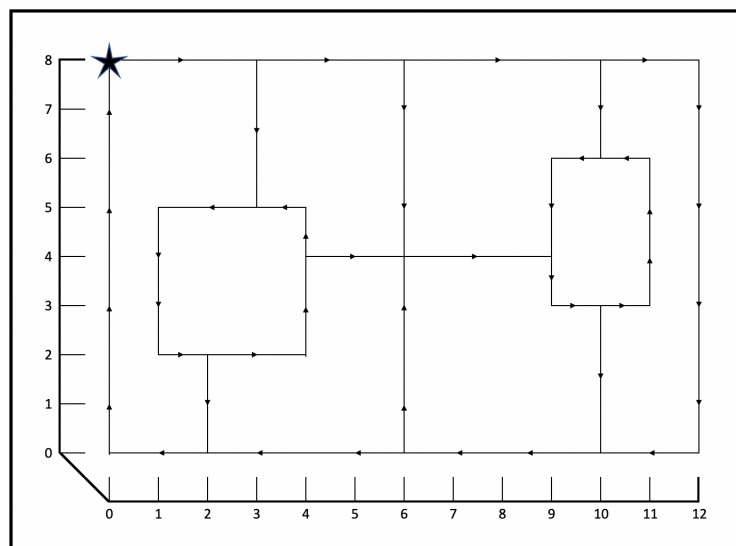


Figura 1: Mapa da cidade

2.2 Análise do problema

A solução para este problema passa por descobrir o circuito ou conjunto de circuitos possíveis que minimizam o custo do percurso. Naturalmente, a solução preferível seria passar uma única vez em cada rua começando e acabando no mesmo ponto.

Após uma breve análise ao mapa da cidade, é visível que existem ruas que vão ter que ser percorridas mais do que uma vez. Na *figura 2*, tem-se um exemplo em que a *rua 3* vai ter que ser percorrida, no mínimo, duas vezes, uma vez que a *rua 1* e a *rua 2* têm ambas que ser percorridas e a única rua que permite a saída destas mesmas é a *rua 3*.

Ao longo do mapa encontram-se várias situações semelhantes a esta, daí poder-se concluir que a solução preferível, que seria percorrer todas as ruas uma única vez, é impossível, devido à topologia das ruas e à restrição de que todas as ruas têm que ser visitadas. Da mesma maneira, é possível identificar ruas em que é suficiente visitá-las uma única vez.

Será também desejável abstrair o problema e representar o mapa da cidade como um grafo orientado (digrafo)[2]:

$$G = (V, A, v_i)$$

onde V representa o conjunto de todos os vértices, ou seja, os locais onde se troca de rua, A é o conjunto ordenado das arestas, que, neste caso representará o conjunto das ruas da cidade e $v_i \in V$ representa o ponto de partida. Para o problema ter solução, é imperativo que o grafo seja fortemente ligado, isto é, a partir de cada vértice existir pelo menos um caminho para qualquer outro vértice. Como a topologia da rede é pequena, verifica-se facilmente que tal regra é obedecida. Numa topologia de rede mais complexa, não era tão facilmente visível esta propriedade do grafo, sendo que seria necessário usar um algoritmo para provar o fecho transitivo de cada vértice (algoritmo de *Floyd-Warshall*, por exemplo).

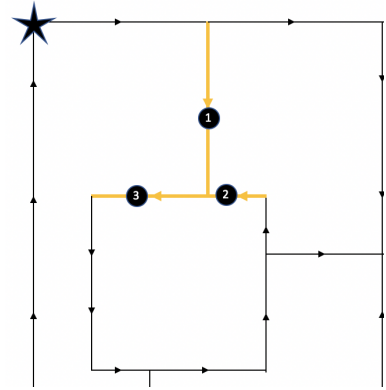


Figura 2: Sub-Mapa da cidade

Nesta primeira fase de análise, pode-se retirar no mínimo três observações importantes que ajudarão a modular o problema em questão:

- É impossível, devido à topologia da rede, passar uma única vez em todas as ruas.
- A solução do problema irá ser um conjunto de sub-circuitos, isto é, vários percursos que saem do ponto inicial e voltam ao mesmo, passando num sub-conjunto das arestas do grafo.
- O número de vezes que se passa no vértice inicial determina o número de sub-circuitos necessários para resolver o problema.

2.3 Construção do modelo

A construção do modelo consiste na especificação de expressões quantitativas para o objectivo e **restrições** do problema, em função das suas **variáveis de decisão**. Dentro da construção do modelo podem distinguir-se três componentes básicas:

1. Variáveis de decisão e Parâmetros

- (a) Variáveis de decisão: incógnitas do problema. Representam os níveis das actividades[1].
- (b) Parâmetros: variáveis controladas do sistema, representando, por exemplo, os níveis de recursos disponíveis[1].

2. Condições, Constrangimentos ou Restrições

- (a) Restrições: explicitam as regras do funcionamento do sistema. Uma solução que obedeça ao conjunto de condições designa-se por **solução válida**[1].

3. Função Objectivo

- (a) Define a medida de eficiência do sistema em função das variáveis de decisão. Permite determinar qual a melhor solução de um conjunto de soluções válidas[1].

2.4 Solução do modelo

A solução do modelo consiste na aplicação de algoritmos existentes ou no desenvolvimento de novos algoritmos para obtenção da solução ótima do modelo.

Visto que este problema consiste em minimizar a distância total percorrida atravessando todas as arestas, optou-se por utilizar a conservação de fluxo e as restrições de não negatividade de modo a obter o conjunto de soluções válidas. A maneira mais simples de tratar problemas do género é através da Programação Linear.

A Programação Linear é um método para atingir o melhor resultado num modelo matemático (quer ele seja de maximização ou minimização) cujas restrições são funções lineares.

Após a modulação do trabalho, irá ser usada a ferramenta *lp_solve* para calcular a solução ótima do problema. Esta ferramenta baseia-se no algoritmo *Simplex*, desenvolvido por *George Dantzig*, para obter a solução ótima de um modelo de programação linear.

2.4.1 Modelo matemático

Após uma análise detalhada do problema e uma vez definidas as ferramentas a usar, bem como a metodologia a aplicar, irá-se definir o modelo matemático que modela o grafo em questão.

2.4.2 Variáveis de decisão

A resolução do problema parte por descobrir qual o caminho com menos custo percorrendo todas as arestas do grafo. Desta maneira, o custo do percurso vai depender do número de vezes que se passa em cada aresta.

Dado isto, este problema tem apenas uma variável de decisão, $x_{i,j}$, que representa o número de vezes que se passa na aresta que liga o vértice i ao vértice j .

2.4.3 Função objectivo

A função objectivo vai depender então do número de vezes que se passa em cada aresta e do custo dessas mesmas arestas. Para além disso, é necessário minimizar esta função de maneira a ter o menor custo possível.

Podemos então definir a função objectivo como:

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

- $x_{i,j}$: número de vezes que se passa na aresta que liga os vértices i e j
- $c_{i,j}$: custo da aresta, em centímetros, que liga os vértices i e j
- A : conjunto de todas as arestas da rede

2.4.4 Restrições

As restrições implementadas ditam as regras para o correto funcionamento do sistema. Com base nisso, dividiu-se as restrições em três grupos:

- **Restrições de conservação de fluxo:** são as restrições que permitem formar caminhos. Para definir um circuito ou conjunto de sub-circuitos temos que garantir que todos os vértices são balanceados, isto é, o número de vezes que se entra num vértice tem que ser igual à soma do número de vezes que se sai desse mesmo vértice.

$$\sum x_{k,i} = \sum x_{i,j}, \quad \forall i \in V$$

$$\{k \mid (k,i) \in A\},$$

$$\{j \mid (i,j) \in A\}$$

- **Restrições de não negatividade:** são as restrições que garantem que o conjunto de soluções válidas sejam positivas e, neste caso, maiores que 0, de maneira a garantir que todas as arestas são visitadas pelo menos uma vez.

$$x_{i,j} \geq 1$$

- **Restrições de integralidade:** são as restrições que definem o domínio do conjunto de soluções válidas. Neste caso, o valor de cada aresta tem que ser um valor inteiro positivo

$$x_{i,j} \in \mathbb{N}^+$$

2.5 Modelo final

De seguida, é apresentado o modelo matemático final.

$$\begin{aligned} \min \quad & \sum x_{i,j} c_{i,j}, & (i,j) \in A \\ \text{s.a} \quad & x_{i,j} \geq 1, & x_{i,j} \in \mathbb{N}^+ \\ & \sum x_{k,i} = \sum x_{i,j}, & \forall i \in V, \\ & & \{k \mid (k,i) \in A\}, \\ & & \{j \mid (i,j) \in A\} \end{aligned}$$

2.5.1 Ficheiro de input

É necessário agora transcrever o modelo matemático para a linguagem que o *lp_solve* reconhece. De seguida, é apresentado o ficheiro de *input* definido de acordo com o modelo desenvolvido.

```

1
2 /* Funcao Objectivo */
3
4 min: 3*x0_1    + 3*x1_2    + 3*x1_10   + 4*x2_3    + 4*x2_24   + 2*x3_4    +
5        2*x3_17   + 8*x4_5    + 2*x5_6    + 4*x6_7    + 4*x7_24   + 4*x7_8    +
6        2*x8_9    + 8*x9_0    + 2*x10_11  + 3*x11_12  + 1*x12_13  + 2*x13_8    +
7        2*x13_14  + 2*x14_15  + 1*x15_16  + 1*x16_10  + 2*x15_24  + 3*x24_19  +
8        1*x19_20  + 1*x20_21  + 3*x21_6   + 1*x21_22  + 3*x22_23  + 1*x23_17  +
9        1*x17_18  + 2*x18_19 ;
10
11 /* Restricoes de controlo de fluxo */
12
13 x0_1 = x1_10 + x1_2;          x1_2 = x2_24 + x2_3;
14 x2_3 = x3_17 + x3_4;          x3_4 = x4_5;
15 x4_5 = x5_6;                  x5_6 + x21_6 = x6_7;
16 x6_7 = x7_24 + x7_8;          x7_8 + x13_8 = x8_9;
17 x8_9 = x9_0;                  x1_10 + x16_10 = x10_11;
18 x10_11 = x11_12;              x11_12 = x12_13;
19 x12_13 = x13_14 + x13_8;       x13_14 = x14_15;
20 x0_1 = x9_0;                  x14_15 = x15_24 + x15_16;
21 x15_16 = x16_10;              x3_17 + x23_17 = x17_18;
22 x17_18 = x18_19;              x18_19 + x24_19 = x19_20;
23 x19_20 = x20_21;              x20_21 = x21_22 + x21_6;
24 x21_22 = x22_23;              x22_23 = x23_17;
25 x2_24 + x15_24 + x7_24 = x24_19;
26
27 /* Restricoes de nao negatividade */
28 x0_1 >= 1;          x1_2 >= 1;          x2_3 >= 1;
29 x3_4 >= 1;          x4_5 >= 1;          x5_6 >= 1;
30 x6_7 >= 1;          x7_8 >= 1;          x8_9 >= 1;
31 x9_0 >= 1;          x1_10 >= 1;         x10_11 >= 1;
32 x11_12 >= 1;         x12_13 >= 1;        x13_14 >= 1;
33 x14_15 >= 1;         x15_16 >= 1;        x16_10 >= 1;
34 x13_8 >= 1;          x3_17 >= 1;         x17_18 >= 1;
35 x18_19 >= 1;         x19_20 >= 1;        x20_21 >= 1;
36 x21_22 >= 1;         x22_23 >= 1;        x23_17 >= 1;
37 x21_6 >= 1;          x15_24 >= 1;        x24_19 >= 1;
38 x2_24 >= 1;          x7_24 >= 1;
39
40 /* Restricoes de integralidade */
41 int  x0_1    , x1_2.    , x1_10    , x2_3    , x2_24    , x3_4    , x3_17    ,
42      x5_6    , x6_7    , x7_24    , x7_8.    , x8_9    , x9_0    , x10_11   ,
43      x12_13   , x13_8    , x13_14   , x14_15   , x15_16   , x16_10   , x15_24   ,
44      x24_19   , x19_20   , x20_21   , x21_6    , x21_22   , x22_23   , x23_17   ,
45      x17_18   , x18_19   , x4_5     , x11_12;

```

Listing 1: *Script* que transcreve o modelo desenvolvido

2.5.2 Ficheiro de *output*

Tendo como base o método *Simplex*, o *lp_solve* calculou o valor das variáveis de decisão que minimizam o custo do percurso no grafo. De seguida, apresenta-se o conjunto de soluções válidas.

```
1 Valor da funcao objectivo: 220.00
2
3 Valor atual das variaveis:
4 x0_1   = 5      x1_2   = 3      x1_10  = 2
5 x2_3   = 2      x2_24  = 1      x3_4   = 1
6 x3_17  = 1      x4_5   = 1      x5_6   = 1
7 x6_7   = 5      x7_24  = 1      x7_8   = 4
8 x8_9   = 5      x9_0   = 5      x10_11 = 3
9 x11_12 = 3      x12_13 = 3      x13_8   = 1
10 x13_14 = 2      x14_15 = 2      x15_16 = 1
11 x16_10 = 1      x15_24 = 1      x24_19 = 3
12 x19_20 = 5      x20_21 = 5      x21_6   = 4
13 x21_22 = 1      x22_23 = 1      x23_17 = 1
14 x17_18 = 2      x18_19 = 2
```

Listing 2: *Output* produzido pelo *lp_solve*

2.6 Validação do modelo

Uma vez calculada a solução do problema, pode-se representar o grafo com os valores obtidos pelo *lp_solve*.

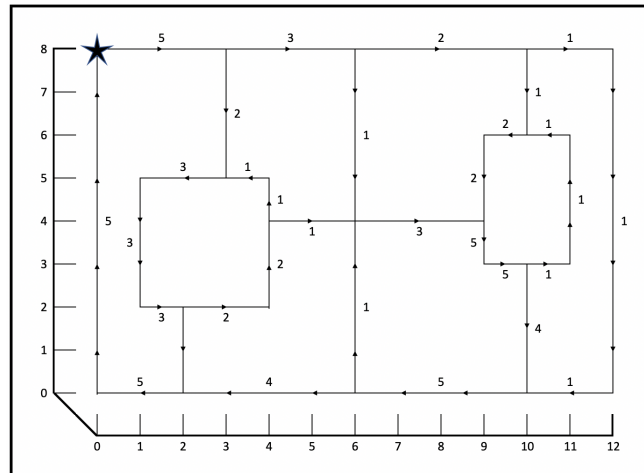


Figura 3: Mapa com o conjunto de soluções

É necessário agora interpretar os resultados e definir o conjunto dos sub-circuitos que minimizam o custo total do trajecto. Este processo passa por ir traçando arestas, decrementando o seu valor, até atingir o valor de 0. Quando é atingido este limite, a aresta em questão já não necessita de ser visitada.

De seguida, é apresentado o conjunto de sub-circuitos obtidos, uma vez feita a análise dos resultados. Note-se que a solução apresentada não é única, isto é, é possível apresentar um conjunto diferente de soluções do que aquele apresentado a baixo obtendo, mesmo assim, o custo mínimo possível.

As arestas coloridas representam o percurso a percorrer. Existem três cores possíveis para estas arestas e têm o seguinte significado:

- Laranja : Arestas percorridas apenas uma vez
- Azul : Arestas percorridas duas vezes
- Vermelho : Arestas percorridas três vezes

Os circuitos 1, 2 e 3 representados nas *figuras 4 e 5* são circuitos acíclicos, isto é, é possível traçar um percurso que parta e regresse ao vértice inicial passando apenas uma vez nas arestas desse mesmo percurso.

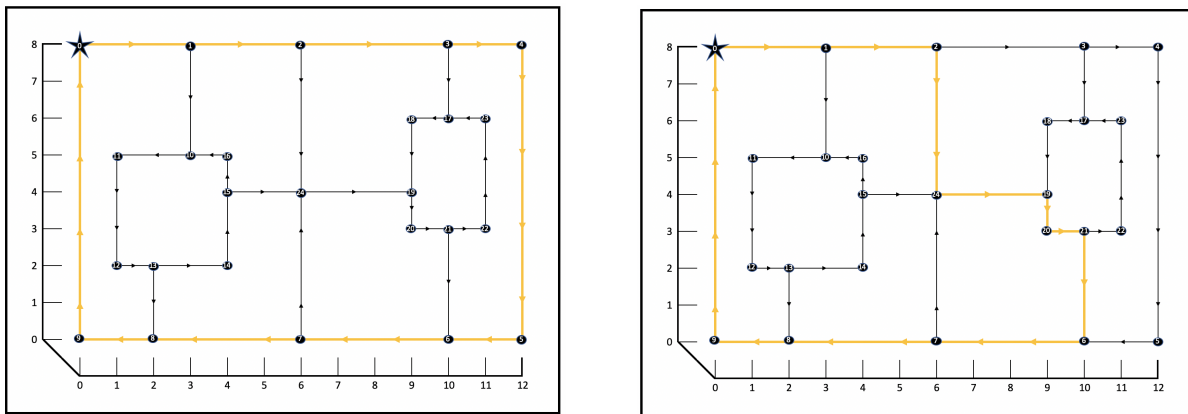


Figura 4: Circuito 1 e Circuito 2

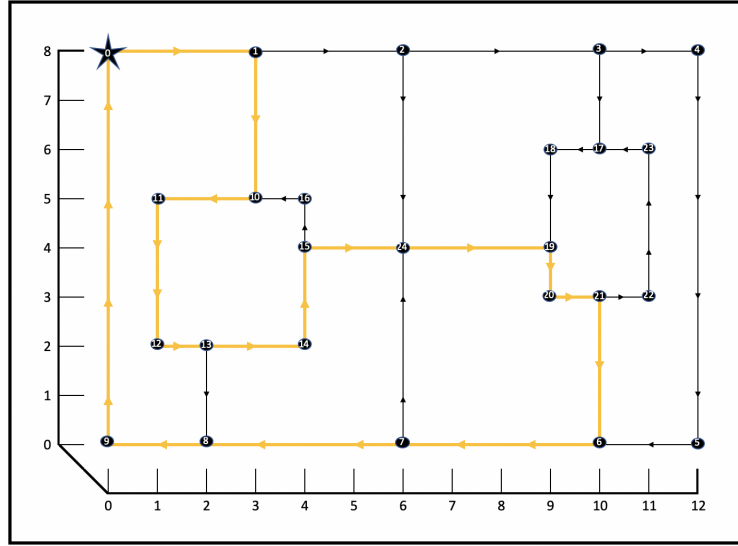


Figura 5: Circuito 3

Na *figura 6* estão representados circuitos cíclicos, isto é, existem arestas que têm que ser percorridas mais do que uma vez de modo a conseguir-se voltar ao vértice inicial.

No *Circuito 4* existem duas maneiras distintas de percorrer o grafo, e são as seguintes:

- **Hipótese 1**

$$\begin{aligned}
 &x_{0,1} \Rightarrow x_{1,2} \Rightarrow x_{2,3} \Rightarrow x_{3,17} \Rightarrow x_{17,18} \Rightarrow x_{18,19} \Rightarrow x_{19,20} \Rightarrow x_{20,21} \Rightarrow \\
 &x_{21,22} \Rightarrow x_{22,23} \Rightarrow x_{23,17} \Rightarrow x_{17,18} \Rightarrow x_{18,19} \Rightarrow x_{19,20} \Rightarrow x_{20,21} \Rightarrow \\
 &x_{21,6} \Rightarrow x_{6,7} \Rightarrow x_{7,24} \Rightarrow x_{24,19} \Rightarrow x_{19,20} \Rightarrow x_{20,21} \Rightarrow x_{21,6} \Rightarrow x_{20,21} \Rightarrow \\
 &x_{6,7} \Rightarrow x_{7,8} \Rightarrow x_{8,9} \Rightarrow x_{9,0}
 \end{aligned}$$

- **Hipótese 2**

$$\begin{aligned}
 &x_{0,1} \Rightarrow x_{1,2} \Rightarrow x_{2,3} \Rightarrow x_{3,17} \Rightarrow x_{17,18} \Rightarrow x_{18,19} \Rightarrow x_{19,20}x_{20,21} \Rightarrow \\
 &x_{21,6} \Rightarrow x_{6,7} \Rightarrow x_{7,24} \Rightarrow x_{24,19} \Rightarrow x_{19,20} \Rightarrow x_{20,21} \Rightarrow x_{20,21} \Rightarrow x_{21,22} \Rightarrow \\
 &x_{22,23} \Rightarrow x_{23,17} \Rightarrow x_{17,18} \Rightarrow x_{18,19} \Rightarrow x_{19,20} \Rightarrow x_{20,21} \Rightarrow x_{21,6} \Rightarrow \\
 &x_{6,7} \Rightarrow x_{7,8} \Rightarrow x_{8,9} \Rightarrow x_{9,0}
 \end{aligned}$$

O *Circuito 5*, embora tenha três arestas que têm que ser percorridas mais do que uma vez, só existe um trajecto admissível:

$$\begin{aligned}
 &x_{0,1} \Rightarrow x_{1,10} \Rightarrow x_{10,11} \Rightarrow x_{11,12} \Rightarrow x_{12,13} \Rightarrow x_{13,14} \Rightarrow x_{14,15} \Rightarrow \\
 &x_{15,16} \Rightarrow x_{16,10} \Rightarrow x_{10,11} \Rightarrow x_{11,12} \Rightarrow x_{12,13} \Rightarrow x_{13,8} \Rightarrow x_{8,9} \Rightarrow x_{9,0}
 \end{aligned}$$

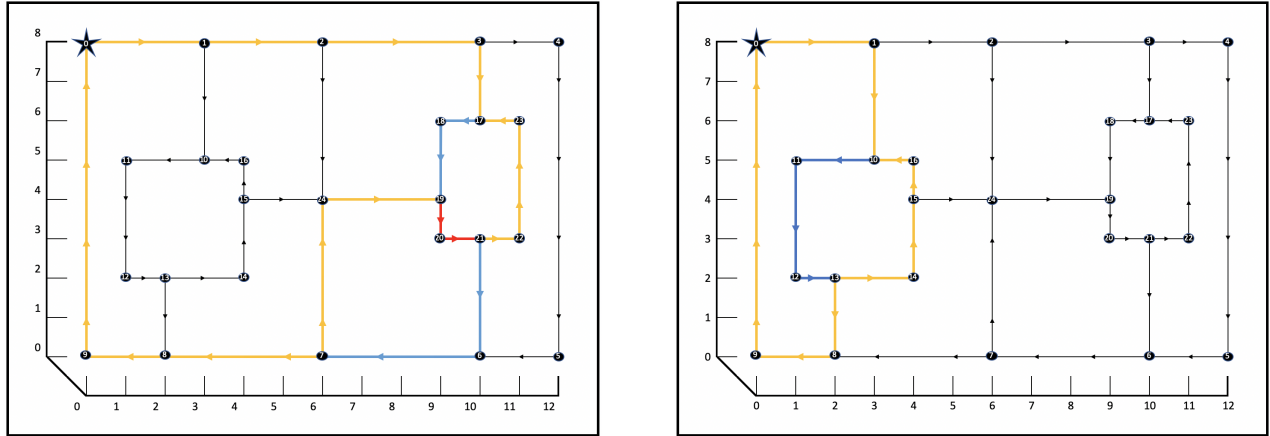


Figura 6: Circuito 4 e Circuito 5

Substituindo agora as variáveis de decisão na função objectivo, obtém-se o custo mínimo do percurso. O custo mínimo total é igual à soma do custo dos sub-circuitos.

Sendo c_x o custo do sub-circuito x , pode-se formular que:

$$\begin{aligned} c_{Total} &= c_{c1} + c_{c2} + c_{c3} + c_{c4} + c_{c5} \\ &= 40 + 36 + 44 + 64 + 36 \\ &= 220cm \end{aligned}$$

Tabela 1: Tabela de custos

| Circuitos | Custo(cm) |
|-----------|-----------|
| 1 | 40 |
| 2 | 36 |
| 3 | 44 |
| 4 | 64 |
| 5 | 36 |
| Total | 220 |

Pode-se concluir que a solução obtida para o problema é uma solução admissível pois obedece a todas as restrições definidas. Na *figura 3* observa-se que, em cada vértice, o valor de entrada é igual à soma dos valores de saída, garantindo, desta forma, a conservação do fluxo da rede. Para além disso, sendo todos os valores obtidos positivos e inteiros, pode-se afirmar que todas as arestas são percorridas pelo menos uma vez. A solução obtida é também óptima para o modelo definido pois foi usado o método *Simplex*, através da ferramenta *lp_solve*, para a calcular.

Como foi mencionado anteriormente, a solução para este problema não é única, sendo possível traçar trajectos diferentes mas que no fim resultam no mesmo valor na função objectivo.

Equiparando com a situação do mundo real dos camiões do lixo, o condutor do camião poderia escolher a ordem pela qual os sub-circuitos são percorridos, uma vez que a distância final percorrida seria a mesma.

3 Discussão

Nesta secção vão-se apresentar as dificuldades encontradas no desenvolvimento do trabalho, assim como uma abordagem diferente para a modulação do grafo.

Inicialmente, acreditou-se que a solução passaria por modular uma árvore de custo mínimo (*MST - Minimum Span Tree*), no entanto, a propriedade que define que uma árvore é um sub-grafo acíclico não permite, para este problema em específico, garantir que todas as arestas são visitadas pelo menos uma vez.

A abordagem seguinte foi pensar em definir a variável de decisão como uma variável binária, isto é, o valor de 0 representava uma aresta não ter sido percorrida e o valor de 1 o contrário. O resultado desta abordagem vem com um problema muito grande, pois, com isto, uma vez que todas as arestas têm que ser percorridas pelo menos uma vez, o valor de todas as variáveis de decisão seria 1, ou seja, só se saberia que todas as arestas tinham sido percorridas, não conseguindo definir os caminhos que foram usados.

Por fim, acabou-se por definir a variável de decisão como o número de vezes que se passa numa determinada aresta, conseguindo assim modular corretamente o problema.

Era possível otimizar o modelo desenvolvido para este caso em específico. Uma vez que existem casos em que só existe um caminho possível entre dois vértices não adjacentes, podia-se assumir que o conjunto de arestas que percorrem esses mesmos vértices formam apenas uma aresta. Por exemplo, na *figura 7*, encontram-se duas situações representativas do que foi dito, podemos sintetizar o conjunto de vértices $\{v_{3,4}, v_{4,5}, v_{5,6}\}$ em apenas $v_{3,6}$, assim como o conjunto $\{v_{21,22}, v_{22,23}, v_{23,17}\}$ em $v_{21,17}$. Esta abordagem tem as suas vantagens e desvantagens. Por um lado, conseguia-se reduzir o número de restrições para este problema, tornando o processo de cálculo da solução mais eficiente. Por outro lado, perdia-se o facto do modelo matemático desenvolvido não ser adaptável a uma rede diferente. Note-se que o modelo desenvolvido é genérico para qualquer grafo orientado, dado que se trabalha com o balanço de cada vértice do grafo, não estando dependente da sua topologia.

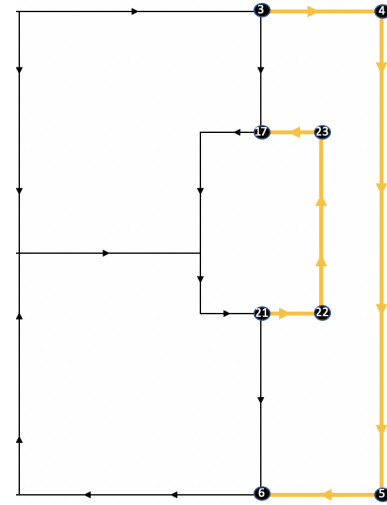


Figura 7: Sub-Mapa da cidade

4 Conclusão

Num sistema de logística de distribuição, a redução do custo do caminho a percorrer é de grande importância, havendo já um conjunto bem definido de algoritmos que permitem calcular a solução ótima para um problema deste género. Os modelos de **Programação Linear** fazem com que seja fácil a modulação de problemas de fluxo de redes, conseguindo atingir um valor ótimo para a função objectivo do modelo.

O trabalho desenvolvido é um variante de um problema de distribuição em que se pretende planear um conjunto de percursos de modo a percorrer todas as arestas de um grafo orientado.

Numa primeira fase, foi feita a análise do problema de modo a encontrar algumas propriedades que facilitassem a melhor percepção do problema em causa.

Após uma análise detalhada, passou-se para a definição do modelo matemático que modela o grafo bem como todos os seus condicionantes. Para isto, foi usado um modelo de **Programação Linear**.

Por fim, foi apresentada a transcrição do modelo para um *script* na linguagem definida pelo *lp_solve* obtendo-se assim a solução ótima para o modelo definido.

O modelo matemático desenvolvido neste trabalho modela qualquer tipo de rede, seja ela mais simples ou complexa, desde que se obedeça à restrição de que o grafo seja orientado. Para além disso, aplica-se também a diferentes contextos do mundo real, podendo modular um sistema de recolha do lixo de uma cidade, bem como um sistema de distribuição de encomendas de uma empresa.

Referências

- [1] A. J. M. Guimarães Rodrigues. *Investigação Operacional, Vol.1* Universidade do Minho, 1993.
- [2] Jon Kleinberg & Éva Tardos, *Algorithm Design, 1st ed.*, Cornell University