Universidade do Minho
Departamento de Matemática
Matemática das Coisas

Projeto 3,
Grafos, Redes & Aplicações

Grupo 3:
Dong Xuyong, 92960, L.E.G.S.I
Inês José, 88690, L.E
Leandro Pereira, 90078, L.C.P
Luís Zhou, 88608, L.E.A
Ricardo Oliveira, 73055, M.I.E.E.I.C

Professora:
Ana Jacinta Soares

31 de Maio de 2022

Conteúdo

Lista de Figuras

Lista de Tabelas

1 Introdução

1.1 Objetivos de aprendizagem

Este relatório realiza-se no âmbito de unidade curricular Matemática das coisas sobre matéria redes e grafos que contêm três problema: Grafo de Euler e o problema das pontes de K¨onigsberg, Grafo de Hamilton e o problema da viagem à volta do mundo e Grafo de Hamilton e o problema do caixeiro viajante.
Estas problemas é atravessar todos vértices dos caminhos fechados (como um circuito fechado) e não volta ao vértice que já passou e com menor trajetória possível, ou seja, trajetória mais económicas.

1.2 Ferramentas utilizadas

(1) *Wolfram One* [1] - é uma plataforma de computação híbrida, integrando totalmente nuvem e desktop - o ponto de partida ideal para usar todos os recursos das tecnologias Wolfram.
Da análise de dados à modelagem (com nossos ou os seus dados selecionados), da publicação de uma API à uma apresentação ao vivo do seu último R&D, de notebooks instantâneos para programar rapidamente seu protótipo, Wolfram One é um produto fácil de usar da empresa de computação que é líder mundial.
De formulários web básicos à análise de dados em larga escala, a tecnologia Wolfram inclui a funcionalidade para qualquer tipo de tarefa computacional.

(2) LaTeX [2] - é um sistema de preparação de documentos para composição tipográfica de alta qualidade. É mais frequentemente usado para documentos técnicos ou científicos de médio a grande porte, mas pode ser usado para quase qualquer forma de publicação.
LaTeXnão é um processador de texto! Em vez disso, o LaTeXincentiva os autores a não se preocuparem muito com a aparência de seus documentos, mas a se concentrarem em obter o conteúdo certo.

(3) *Overleaf* [3] - é uma startup e empresa social que cria ferramentas modernas de autoria colaborativa para ajudar a tornar a ciência e a pesquisa mais rápidas, abertas e transparentes. A tecnologia de colaboração líder de mercado da Overleaf está agora em uso por mais de nove milhões de pesquisadores, estudantes e professores em instituições, laboratórios e indústrias em todo o mundo.

(4) *Maple* [4] - é o fornecedor líder de ferramentas de software de alto desempenho para engenharia, ciências e matemática. Seu conjunto de produtos reflete a filosofia de que, com ótimas ferramentas, as pessoas podem fazer grandes coisas.

(5) *Matlab* [5] - é uma plataforma de programação projetada especificamente para engenheiros e cientistas analisarem e projetarem sistemas e produtos que transformam nosso mundo. O coração do Matlab é a linguagem Matlab, uma linguagem baseada em matriz que permite a expressão mais natural da matemática computacional.

2 O método de Dijkstra

2.1 Introdução

O algoritmo de Dijkstra é um algoritmo para encontrar os caminhos mais curtos entre nós em um grafo, que pode representar, por exemplo, redes rodoviárias. Foi concebido pelo cientista da computação Edsger W. Dijkstra em 1956 e publicado três anos depois.
O algoritmo existe em muitas variantes. O algoritmo original de Dijkstra encontrou o caminho mais curto entre dois nós, mas uma variante mais comum fixa um único nó como o nó "fonte"e encontra os caminhos mais curtos da origem para todos os outros nós no grafo, produzindo uma árvore de caminho mais curto.
Para um determinado nó de origem no grafo, o algoritmo encontra o caminho mais curto entre esse nó e todos os outros. Ele também pode ser usado para encontrar os caminhos mais curtos de um único nó para um único nó de destino, parando o algoritmo assim que o caminho mais curto para o nó de destino for determinado. Por exemplo, se os nós do gráfico representam cidades e os custos de caminhos de borda representam distâncias de condução entre pares de cidades conectadas por uma estrada direta (para simplificar, ignore luzes vermelhas, sinais de parada, estradas com pedágio e outras obstruções), o algoritmo de Dijkstra pode ser usado para encontrar a rota mais curta entre uma cidade e todas as outras cidades.
Uma aplicação amplamente utilizada de algoritmos de caminho mais curto são os protocolos de roteamento de rede, mais notavelmente IS-IS (Sistema Intermediário para Sistema Intermediário) e OSPF (Open Shortest Path First). Também é empregado como uma sub-rotina em outros algoritmos, como o de Johnson.
O algoritmo de Dijkstra usa rótulos que são inteiros positivos ou números reais, que são totalmente ordenados. Pode ser generalizado para usar quaisquer rótulos parcialmente ordenados, desde que os rótulos subsequentes (um rótulo subsequente é produzido ao atravessar uma aresta) sejam monotonicamente não decrescentes. Essa generalização é chamada de algoritmo genérico de caminho mais curto de Dijkstra.

2.2 Algoritmo

Seja a distância do nó Y a distância do nó inicial a Y. O algoritmo de Dijkstra começará inicialmente com distâncias infinitas e tentará melhorá-las passo a passo.

(1) Marque todos os nós como não visitados. Crie um conjunto de todos os nós não visitados chamado de conjunto não visitado.

(2) Atribua a cada nó um valor de distância provisório: defina-o como zero para nosso nó inicial e para infinito para todos os outros nós. A distância tentativa de um nó v é o comprimento do caminho mais curto descoberto até agora entre o nó v e o nó inicial. Como inicialmente nenhum caminho é conhecido para qualquer outro vértice além da própria fonte (que é um caminho de comprimento zero), todas as outras distâncias provisórias são inicialmente definidas como infinitas. Defina o nó inicial como atual.

(3) Para o nó atual, considere todos os seus vizinhos não visitados e calcule suas distâncias provisórias através do nó atual. Compare a distância provisória recém-calculada com a distância atualmente atribuída ao vizinho e atribua a menor. Por exemplo, se o nó atual A está marcado com uma distância de 6, e a aresta que o conecta com um vizinho B tem comprimento 2, então a distância de B até A será $6 + 2 = 8$. Se B foi previamente marcado com uma distância maior que 8 então mude para 8. Caso contrário, o valor atual será mantido.

(4) Quando terminarmos de considerar todos os vizinhos não visitados do nó atual, marque o nó atual como visitado e remova-o do conjunto não visitado. Um nó visitado nunca será verificado novamente (isso é válido e ótimo em conexão com o comportamento na etapa 6.: que os próximos nós a serem visitados serão sempre na ordem de 'menor distância do nó inicial primeiro', portanto, quaisquer visitas posteriores seriam tem uma distância maior).

(5) Se o nó de destino foi marcado como visitado (ao planejar uma rota entre dois nós específicos) ou se a menor distância tentativa entre os nós do conjunto não visitado for infinita (ao planejar uma travessia completa; ocorre quando não há conexão entre o nó inicial e nós não visitados restantes), então pare. O algoritmo terminou.

(6) Caso contrário, selecione o nó não visitado que está marcado com a menor distância tentativa, defina-o como o novo nó atual e volte para a etapa 3.

Ao planear uma rota, na verdade não é necessário esperar até que o nó de destino seja "visitado"como acima: o algoritmo pode parar quando o nó de destino tiver a menor distância tentativa entre todos os nós "não visitados"(e, portanto, pode ser selecionado como o nó de destino). próxima "atual").

2.3 Enunciado do problema

Dado um grafo e um vértice fonte no grafo, encontre os caminhos mais curtos da fonte para todos os vértices no grafo dado. O algoritmo de Dijkstra é muito semelhante ao algoritmo de Prim para árvore geradora mínima.
O objetivo é determinar-mos o caminho mais curto do vértice 0 ao vértice 4.
Abaixo estão as etapas detalhadas usadas no algoritmo de Dijkstra para encontrar o caminho mais curto de um único vértice de origem para todos os outros vértices no grafo fornecido.

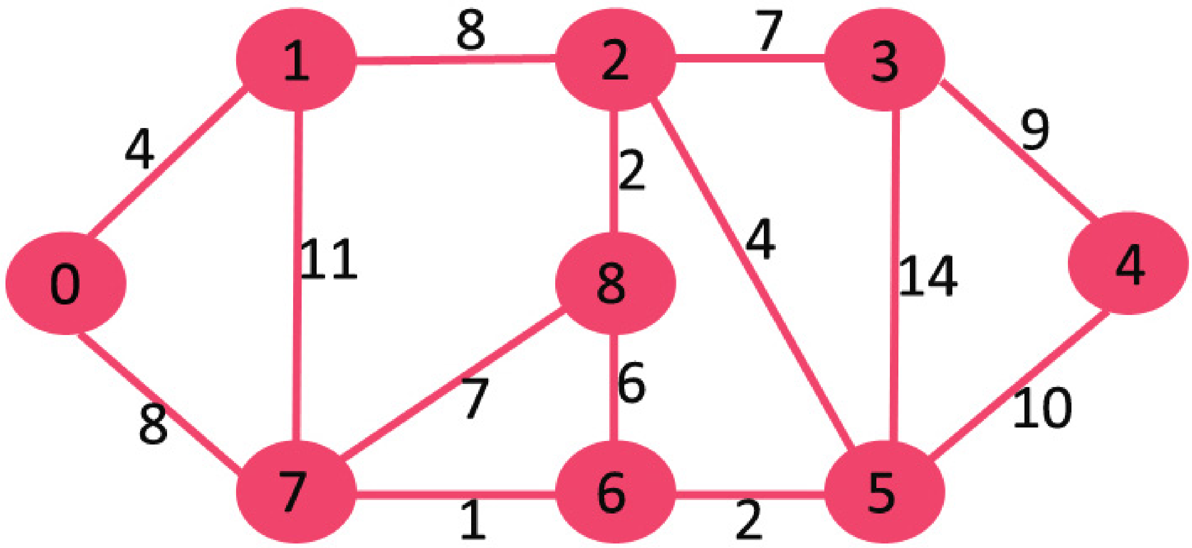

Figura 1: Grafo do problema.

2.4 Resolução do problema

Vamos agora processar em resultados númericos, para depois codificar-mos numa ferramenta computacional, para isso, obtemos a respectiva Matriz de Adjacência:

$$\begin{bmatrix} 0 & 4 & 0 & 0 & 0 & 0 & 0 & 8 & 0 \\ 4 & 0 & 8 & 0 & 0 & 0 & 0 & 11 & 0 \\ 0 & 8 & 0 & 7 & 0 & 4 & 0 & 0 & 2 \\ 0 & 0 & 7 & 0 & 9 & 14 & 0 & 0 & 0 \\ 0 & 0 & 0 & 9 & 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 4 & 14 & 10 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 1 & 6 \\ 8 & 11 & 0 & 0 & 0 & 0 & 1 & 0 & 7 \\ 0 & 0 & 2 & 0 & 0 & 0 & 6 & 7 & 0 \end{bmatrix}$$

A ferramenta que vamos usar para resolver este exercício é a ferramenta Maple.

O Maple tem vários comandos relacionados com teoria de grafos, que fazem parte da package networks.

```
with(GraphTheory)
```

A matriz pode ser escrita no Maple de várias formas, a que nos utilizamos, foi a seguinte:

```
C1 := Graph({[{0, 1}, 4], [{0, 7}, 8], [{1, 2}, 8], [{1, 7}, 11],
[{2, 3}, 7], [{2, 5}, 4], [{2, 8}, 2], [{3, 4}, 9], [{3, 5}, 14],
[{4, 5}, 10], [{5, 6}, 2], [{6, 7}, 1], [{6, 8}, 6], [{7, 8}, 7]})
```

Output:

```
C1 := ‘Graph 1: an undirected weighted graph with 9 vertices and 14 edge(s)‘
```

```
DrawGraph(C1)
```

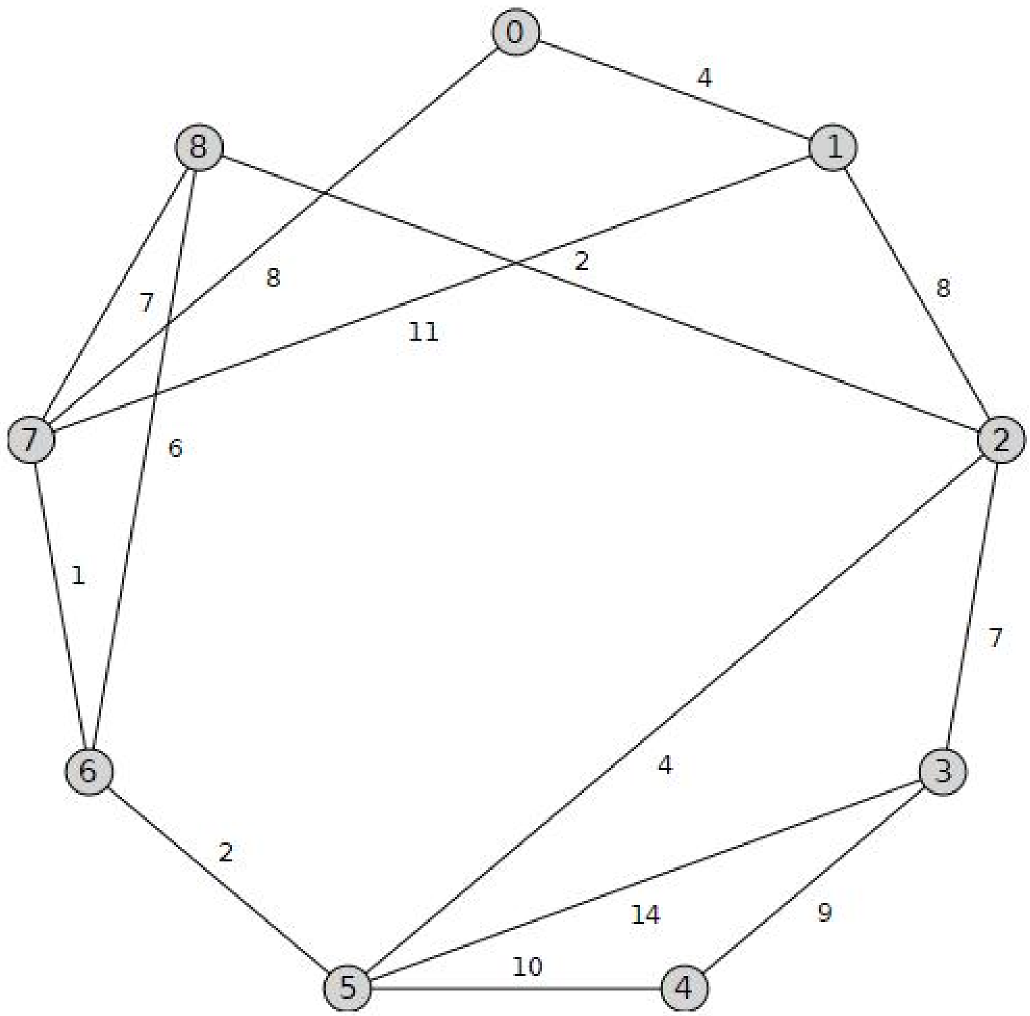

Figura 2: Grafo do problema, usando a ferramente Maple.

Calculamos todos os caminhos possíveis a partir do vértice 0:

```
DijkstrasAlgorithm(C2, 0)
Output:
[[[0], 0], [[0, 1], 4], [[0, 1, 2], 12], [[0, 1, 2, 3], 19],
[[0, 7, 6, 5, 4], 21], [[0, 7, 6, 5], 11], [[0, 7, 6], 9],
[[0, 7], 8], [[0, 1, 2, 8], 14]]
```

Calculamos todos o caminhos mais próximo do vértice 0 ão vértice 4:

```
DijkstrasAlgorithm(C1, 0, 4)
Output:
[[0, 7, 6, 5, 4], 21]
```

Depois de obtermos a solução, visualizamos o gráfico.

```
C2 := Graph({[{0, 7}, 8], [{4, 5}, 10], [{5, 6}, 2], [{6, 7}, 1]})
Output:
C2 := ‘Graph 2: an undirected weighted graph with 5 vertices and 4 edge(s)‘

DrawGraph(C2)
```

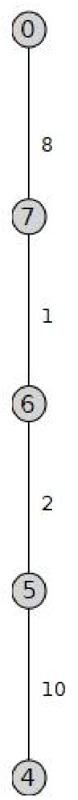

Figura 3: Grafo do caminho mais curto de Dijkstra, do ponto 0 ao 4.

3 Grafo de Euler e o problema das pontes de Konigsberg

3.1 Contexto histórico

Após sua segunda cruzada contra os prussianos, cavaleiros fundaram Königsberg em 1254 sob a liderança do rei Ottoker II. A forte economia da cidade permitiu-lhe construir sete pontes sobre o rio, a maioria das quais ligada à ilha de Kneiphof. Mesmo que ninguém pudesse conceber uma rota que lhes permitisse atravessar cada ponte apenas uma vez, eles não podiam demonstrar que isso era impossível.

3.2 Euler e o problema

Euler foi um brilhante matemático do século XVIII que contribuiu para quase todos os ramos da matemática. Ele resolveu um problema astronômico aos 28 anos que os astrônomos previram que levaria meses para ser resolvido: o quebra-cabeça de Koenigsberg é um chamado problema de rede em topologia- um ramo da matemática preocupado com as propriedades de objetos que podem resistir a alongamentos, torções, dobras e outras mudanças. Euler percebeu que um importante princípio científico estava escondido na estrutura do quebra-cabeça ao resolvê-lo. A solução de Euler para o Problema da Ponte de Königsberg foi baseada apenas na lógica e não foi descoberta usando nenhum princípio matemático (Scientific American, 1953).
O quebra-cabeça de Koenigsberg envolve traçar uma determinada figura no papel sem levantar o lápis ou refazer uma linha que são um excelente exemplo da simplicidade enganosa dos problemas de topologia: apenas um dos 8 padrões pode ser traçado em um único traço, e esse padrão é o da direita. Por lidar com relacionamentos baseados apenas em posição e ignorar magnitudes, Leibnitz cunhou o termo "geometria de posição". A questão de saber se é possível realizar uma caminhada para que cada uma das sete pontes da ilha prussiana possa ser atravessada apenas uma vez.
Esse problema, acreditava Euler, estava relacionado a um tópico que Gottfried Wilhelm Leibniz havia discutido anteriormente e com o qual desejava trabalhar, que agora é conhecido como teoria dos grafos.

3.3 A solução apresentada

Euler apresenta um artigo para solucionar o problema a 26 de agosto de 1735, publicado posteriormente como 'Solutio problematis ad geometriam situs pertinentis'. Euler explica como ele desenvolveu um sistema útil para representar uma travessia de ponte. Por exemplo, em sua Figura 4, AB denotaria uma jornada que começou na massa de terra A e terminou na massa de terra B.

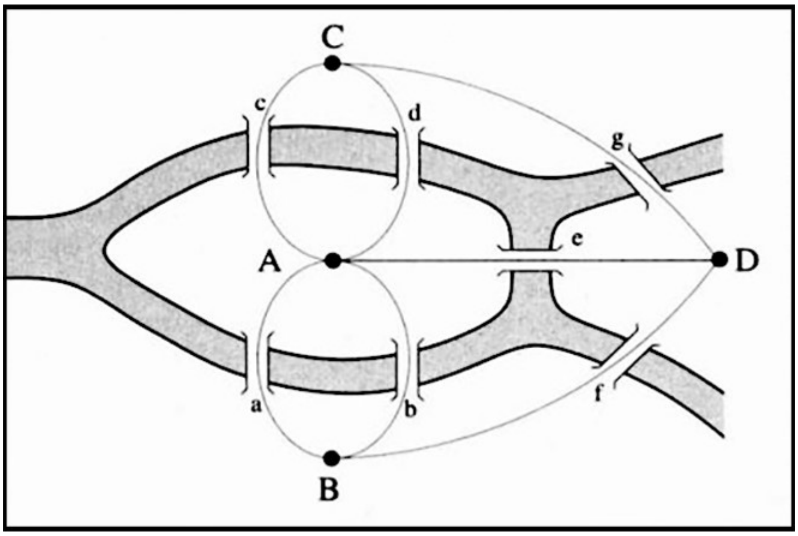

Figura 4: Math Stack Exchange.

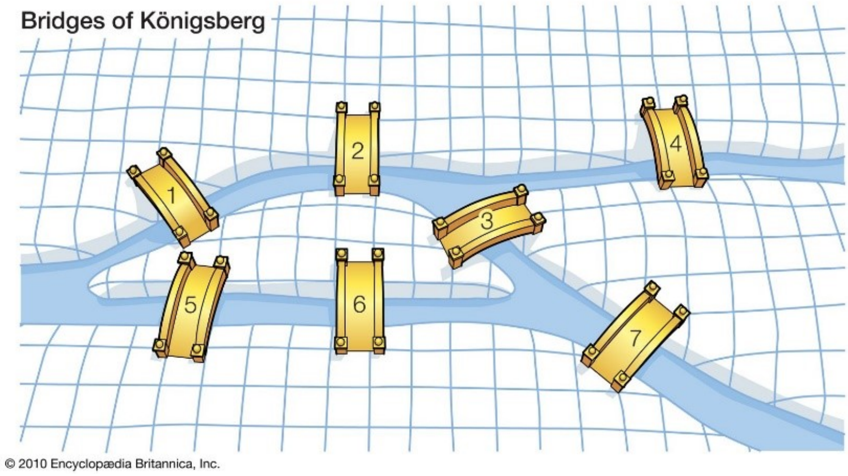

Figura 5: Britannica.

De seguida o trabalho de Euler sublinha que se houver várias pontes disponíveis para a travessia de uma massa de terra para outra, não faz diferença qual delas é usada. Usando seu diagrama original, ele explica que o problema de Königsberg requer exatamente oito letras, com os pares de A, B e C aparecendo próximos um do outro exatamente duas vezes para que exista um caminho que cruze cada ponte uma vez e apenas uma vez. Se houver um número ímpar de pontes conectando as regiões A entre si, divida por dois para encontrar o número de vezes que A deve ser usado no caminho, onde cada ponte é usada apenas uma vez. Como A está conectado por cinco pontes, ele deve aparecer três vezes, enquanto B, C e D devem aparecer duas vezes. No entanto, Euler já afirmou que as sete pontes podem ter apenas oito ocorrências, surgido uma incoerência. Deste modo, podemos concluir que é impossível atravessar todas as pontes de Königsberg uma vez e apenas uma vez.

3.4 Aplicações gerais

Euler demonstra o seguinte usando o problema de Konigsberg como seu primeiro exemplo: Número de pontes = 7, mais um, para um total de 8, que é maior que 8. Euler tenta encontrar um caminho que permite que alguém atravessar cada ponte apenas uma vez. Ele decide criar seu próprio cenário com 2 ilhas, 4 rios e 15 pontes porque este exemplo é muito simples. Euler segue o mesmo procedimento de antes, nomeando as cinco regiões com letras maiúsculas.
Como o total é 16, a jornada deve começar em D ou E. Este é um bom exemplo do método que Euler empregaria ao resolver um problema dessa natureza.
Nº pontes = 15, Nº de pontes +1 = 16

Regiões	Pontes	Vezes que pontes tem de aparecer
A	8	4
B	4	2
C	4	2
D	3	2
E	5	3
F	6	3

Tabela 1: Tabela do primeiro exemplo do problema de Konigsberg.

Além disso, $4 + 2 + 2 + 2 + 3 + 3 = 16$, que é igual ao número de pontes, mais uma, o que significa que a viagem é, de fato, possível. (Paoletti,2006)

4 O problema do carteiro chinês

Neste problema, originalmente estudado pelo matemático chinês Mei-Ku Kwan, em 1962, um carteiro tem que distribuir cartas pelas casas de um bairro, voltando depois ao ponto de partida no estacão dos correios. Qual é a menor distancia que terá o carteiro de percorrer? Evidentemente terá de percorrer cada rua pelo menos uma vez, mas deverá evitar percorrê-las mais do que uma vez. Este problema pode ser formulado em termos de grafos com pesos, onde o grafo corresponde ‘a rede de ruas e o peso de cada aresta é o comprimento da respetiva rua. Se o grafo tiver um caminho euleriano, o carteiro pode iniciar esse caminho no estacão dos correios, percorrê-lo e voltar aos correios. Nenhum outro trajeto tem comprimento menor.

Neste problema, o mesmo pretende visitar várias cidades e voltar ao ponto de partida, percorrendo a menor distancia possível. Por exemplo, se existirem 5 cidades A, B, C, D e E que apresenta vários pesos, como ilustrado na seguinte figura.

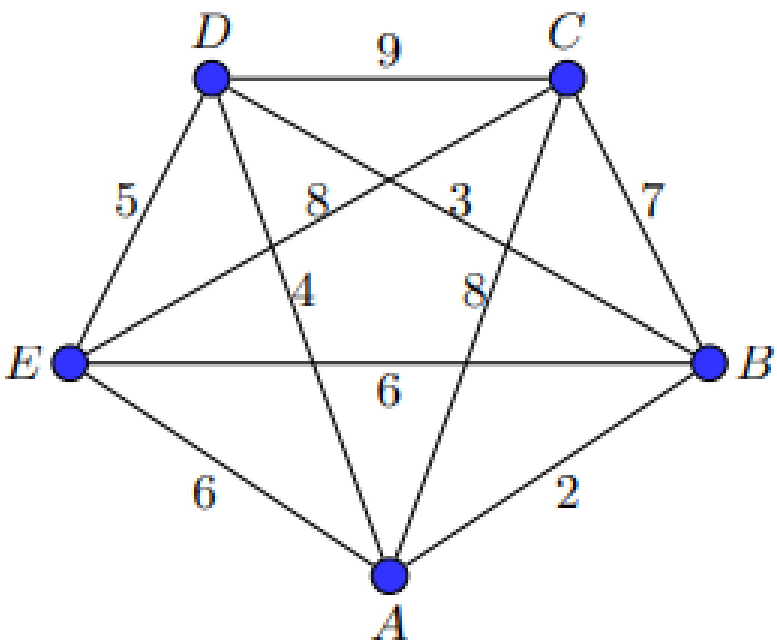

Figura 6: Figura do problema do carteiro chinês.

Neste caso o que se pretende é encontrar um caminho hamiltoniano de menor peso possível.

4.1 Resolução do problema

O problema vai ser resolvido recorrendo ao programa LPSolve, começando com a construção do modelo, seguido das suas restrições e por fim minimização do função objetivo.

4.2 Construção do modelo

A	B	C	D	E
	2	8	4	6
		7	9	6
			9	8
				5

Tabela 2: Tabela do problema do carteiro chinês.

Todas as arestas estão representadas por: xij, onde i e j pertence ao conjunto 1,2,3,4,5.
A função objetivo que minimiza os custos, construído a partir das tabelas anteriores.

1	A
2	B
3	C
4	D
5	E

min: $2x12 + 8x13 + 4x14 + 6x15 + 7x23 + 3x24 + 6x25 + 9x34 + 8x35 + 5x45$
Definição de todas as arestas por binários, no caso de uso da aresta apresenta o valor 1, caso contrário, valor 0.

```
bin x12, x13, x14, x15,
    x23, x24, x25,
    x34, x35,
    x45;
```

Construção de todas as ligações entre as arestas que pertencem ao cada um dos vértices respetivamente.

```
A = x12 + x13 + x14 + x15;
B = x12 + x23 + x24 + x25;
C = x13 + x23 + x34 + x35;
D = x14 + x24 + x34 + x45;
E = x15 + x25 + x35 + x45;
```

4.3 Restrições

Todos os vértices têm de estar associados a dois arestas, ou seja, todos os vértices estarão interligados por duas aresta formando um caminho euleriano.

```
A = 2;
B = 2;
C = 2;
D = 2;
E = 2;
```

4.4 Obtenção dos resultados

Após correr o programa, obtemos os seguintes resultados. Pode se concluir o caminho euleriano

x12	1
x13	0
x14	1
x15	0
x23	1
x24	0
x25	0
x34	0
x35	1
x45	1

será constituído pelas seguintes arestas x12, x14, x23, x35 e x45. O que um custo de 26, no total. Considerando o resultado, existe vários percursos possíveis, portanto selecionamos começar no ponto A, e obtemos o seguinte percurso.

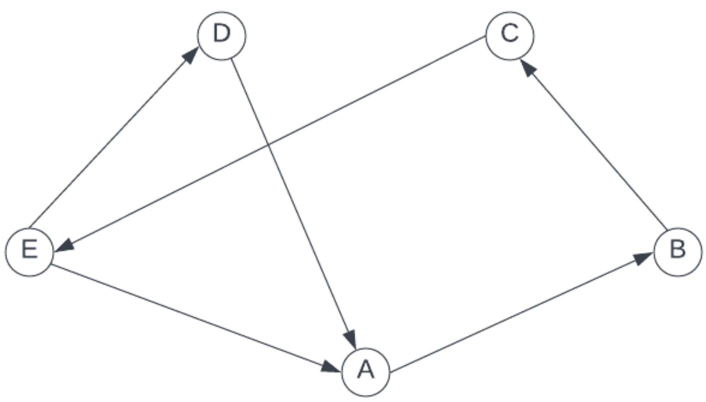

Figura 7: Resultado Final.

5 Conclusão

Referências

[1] Wolfram One. `https://www.wolfram.com/wolfram-one/`

[2] Project Latex. `https://www.latex-project.org/about/`

[3] Overleaf. `https://www.overleaf.com/about/`

[4] Maple. `https://www.maplesoft.com/`

[5] Matlab. `https://www.mathworks.com/discovery/what-is-matlab.html/`

[6] Ana Jacinta Soares, *Cálculo (A e B), MIEEIC, MIECOM, 2007/2008 : notas sobre a disciplina*, Departamento de Matemática, Universidade do Minho, 2007.

[7] Gaspar J. Machado, *Tópicos de Álgebra Linear e Geometria Analítica*, Departamento de Matemática e Aplicações, Universidade do Minho, 2014.

[8] Jorgue Figueiredo e Carolina Ribeiro, *Apontamentos de Equações Diferenciais (Complementos de Análise Matemática EE)*, Departamento de Matemática e Aplicações, Universidade do Minho, 2013.

[9] A. Ismael F. Vaz, *Métodos Numéricos - MATLAB*, Departamento de Produção e Sistemas, Escola de Engenharia, Universidade do Minho, 2016/2017.

[10] Jorge Picado, *Estruturas Discretas Texto de apoio*, Departamento de matemática, Univeridade de Coimbra.

[11] Paoletti, Teo (2006) Leonard Euler's Solution to the Konigsberg Bridge Problem. `https://www.maa.org/press/periodicals/convergence/leonard-eulers-solution-to-the-konigsberg-bridge-problem`

[12] Scientific American (1953) Leonhard Euler and the Koenigsberg bridges. `https://www.imsc.res.in/~sitabhra/teaching/sb15b/ScientificAmerican_1953_Leonhard_Euler_and_the_Koenigsberg_Bridges.pdf`