



UNICAMP

UNIVERSIDADE ESTADUAL DE CAMPINAS

Instituto de Matemática, Estatística e Computação Científica

CLÓVIS RODRIGUES VILAS-BOAS

INTRODUÇÃO AO ESTUDO DE GRAFOS:
ORIGEM E APLICAÇÕES

CAMPINAS

2016



UNIVERSIDADE ESTADUAL DE CAMPINAS
Instituto de Matemática, Estatística
e Computação Científica

CLÓVIS RODRIGUES VILAS-BOAS

INTRODUÇÃO AO ESTUDO DE GRAFOS:
ORIGEM E APLICAÇÕES

Dissertação apresentada ao Instituto de Matemática, Estatística e Computação Científica da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Mestre.

Orientador: Prof. Dr. Pedro José Catuogno

ESTE EXEMPLAR CORRESPONDE À VERSÃO FINAL DA DISSERTAÇÃO DEFENDIDA PELO ALUNO CLÓVIS RODRIGUES VILAS-BOAS, E ORIENTADA PELO PROF. DR. PEDRO JOSÉ CATUOGNO.

Assinatura do Orientador

CAMPINAS

2016

Agência(s) de fomento e nº(s) de processo(s): Não se aplica.

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Maria Fabiana Bezerra Muller - CRB 8/6162

V71i Vilas-Boas, Clóvis Rodrigues, 1978-
Introdução ao estudo de grafos : origem e aplicações / Clóvis Rodrigues
Vilas-Boas. – Campinas, SP : [s.n.], 2016.

Orientador: Pedro José Catuogno.
Dissertação (mestrado profissional) – Universidade Estadual de Campinas,
Instituto de Matemática, Estatística e Computação Científica.

1. Teoria dos grafos. 2. Problema do caixeiro viajante. 3. Euler, Teorema de.
4. Problema das quatro cores. 5. Problema de colorização de mapa. 6.
Heurística. I. Catuogno, Pedro José, 1959-. II. Universidade Estadual de
Campinas. Instituto de Matemática, Estatística e Computação Científica. III.
Título.

Informações para Biblioteca Digital

Título em outro idioma: Introduction to the study of graphs : origin and applications

Palavras-chave em inglês:

Graph theory

Traveling-salesman problem

Euler theorem

Four-color problem

Map-coloring problem

Heuristic

Área de concentração: Matemática em Rede Nacional

Titulação: Mestre

Banca examinadora:

Pedro José Catuogno [Orientador]

Roberto Andreani

Daniel Miranda Machado

Data de defesa: 19-10-2016

Programa de Pós-Graduação: Matemática em Rede Nacional

**Dissertação de Mestrado Profissional defendida em 19 de outubro de 2016
e aprovada pela Banca Examinadora composta pelos Profs. Drs.**

Prof(a). Dr(a). PEDRO JOSÉ CATUOGNO

Prof(a). Dr(a). ROBERTO ANDREANI

Prof(a). Dr(a). DANIEL MIRANDA MACHADO

A Ata da defesa com as respectivas assinaturas dos membros
encontra-se no processo de vida acadêmica do aluno.

Agradecimentos

Aos meus pais, Miguel e Maria, que, com toda paciência e amor, me educaram e ensinaram que temos a responsabilidade de nos aprimorar sempre e o dever de aplicar o conhecimento em prol do desenvolvimento da sociedade.

Ao meu tio Paulo, que foi como um pai e sempre participou da educação dos sobrinhos.

Ao meu irmão Cláudio, que muito auxiliou na minha educação e formação acadêmica.

À minha esposa Juliana e aos meus filhos, Maria Laura, Guilherme e Henrique, que sempre me incentivaram e auxiliaram nos momentos mais turbulentos.

À amiga e colega de turma Andrea de Paula Machado, pela companhia nos dois anos de viagens semanais e pelo exemplo de dedicação e responsabilidade.

Aos professores da Unicamp, em particular do IMECC, que, além de extremamente competentes, sempre estiveram dispostos a sanar qualquer dúvida.

Ao meu orientador Pedro José Catuogno, pela paciência e disposição em me auxiliar com o desenvolvimento do projeto.

À Capes, pelo apoio financeiro.

Aos colegas de turma, pela convivência e pelas inúmeras ajudas nesses dois anos.

Resumo

Nesse trabalho são explorados os conceitos iniciais, a história e algumas aplicações importantes da teoria de grafos. Clássicos, como o problema das sete pontes de Königsberg, o problema carteiro chinês, o problema do caixeiro viajante e a coloração de mapas servem de contexto e de apoio para o desenvolvimento da teoria, onde importantes teoremas são apresentados e demonstrados. Também são apresentadas as heurísticas de melhoramento K-OPT, Algoritmos Genéticos e Colônia de Formigas. Com o intuito de servir como material de apoio à introdução do estudo dos grafos, traz algumas sugestões de apresentação do tema e do desenvolvimento das aulas.

Palavras-chave: Grafos. Caixeiro Viajante. Pontes de Königsberg. Teorema das Quatro Cores. Heurísticas de Melhoramento.

Abstract

This work explores the following initial concepts, the history and some important applications of the graph theory. Classics, like the problem of The Seven Bridges of Königsberg, The Chinese Postman Problem, The Traveling Salesman Problem, and the coloring of maps, serve as surrounding context and as support for the development of the theory, where important theorems are presented and demonstrated. Also are presented the heuristics of K-OPT Improvement, Genetic Algorithms and Ant Colony Optimization. Aiming to serve as a material of support for the introduction of the study of graphs, bringing some suggestions of presentation of the theme and development of classes.

Keywords: Graphs. Traveling Salesman. Bridges of Königsberg. Four Color Theorem. Improvement Heuristic.

SUMÁRIO

Introdução	9
1. Origem da teoria: o problema das sete pontes de Königsberg	11
1.1. Definições básicas.....	12
1.2. A solução do problema das pontes de Königsberg.....	19
2. Grafos hamiltonianos	21
2.1. Caminhos e ciclos hamiltonianos.....	22
2.2. O Problema do Caixeiro Viajante.....	27
2.2.1. Método exaustivo.....	28
2.2.2. Método do vizinho mais próximo.....	30
2.2.3. Método da aresta de menor peso.....	31
2.2.4. Método repetitivo do vizinho mais próximo.....	34
3. Heurísticas de melhoramento	36
3.1. Heurística de melhoramento k-opt.....	36
3.1.1. Heurística 2-opt.....	37
3.1.2. Heurística 3-opt.....	40
3.2. Algoritmos genéticos.....	41
3.3. Colônia de formigas.....	47
3.3.1. Ant System.....	49
3.3.2. Ant Colony System.....	51
4. Grafos no Ensino Básico	54
4.1. O problema da casinha e as pontes de Königsberg.....	55
4.2. O problema das três casas.....	56
4.3. O problema do carteiro chinês.....	60
4.4. Coloração	66
4.5. O caixeiro viajante.....	70
Referências Bibliográficas	71

INTRODUÇÃO

A Teoria dos Grafos é uma área extremamente jovem quando se tomam como referências as origens do pensamento matemático. Surgiu em 1736 como um desafio hoje conhecido como “*As Pontes de Königsberg*”. O “*Problema do Caixeiro Viajante*” foi outro que ficou globalmente conhecido e estudado. Outro problema, proposto em meados do século XIX, denominado “*O Problema das Quatro Cores*”, com uma proposta extremamente simples, levou inúmeros matemáticos a dedicarem seu tempo numa análise mais aprofundada da situação. E para que serviu todo esse trabalho? Inicialmente para a própria teoria, que se desenvolveu muito. Durante esse desenvolvimento foi possível perceber inúmeras aplicações em problemas do cotidiano. Inúmeras heurísticas foram desenvolvidas para encontrar e para melhorar soluções de problemas de otimização; muitos algoritmos foram criados e hoje são amplamente utilizados nas mais diversas áreas. Problemas do cotidiano como, por exemplo, gerenciar tarefas conflitantes, otimizar o sistema de coleta de lixo e de distribuição de cartas e encomendas, determinar como deve ser desenhado o circuito de uma placa de circuito impresso e muitas outras situações podem ser modeladas através de grafos [35].

Apesar do potencial motivador que possui, o tópico Teoria dos Grafos não está presente formalmente no currículo do Ensino Básico brasileiro; mesmo abrangendo inúmeras situações que poderiam aproximar a matemática trabalhada nas salas de aula a problemas encontrados no cotidiano, esse conteúdo é citado apenas como uma sugestão a ser trabalhado em feiras de ciência, laboratórios de matemática ou para compor a parte diversificada do currículo [39].

Este trabalho traz uma parte da história do desenvolvimento da teoria dos grafos, alguns dos importantes teoremas que compõem tal tópico e apresenta sugestões de aplicações do conteúdo voltado aos alunos do Ensino Básico, preferencialmente do Ensino Médio, com o intuito de criar uma aproximação entre a matemática vista nas salas de aula e as situações cotidianas onde o conhecimento pode ser aplicado. É importante destacar que não há o intuito de suprimir teoremas complexos ou sugerir que o rigor matemático seja deixado em segundo plano. É apenas sugerido um tema motivador onde problemas de extrema simplicidade de

exposição se associam às experiências cotidianas de qualquer aluno. A partir dessa aproximação com o cotidiano espera-se que haja maior aceitação da matemática como uma importante ferramenta que pode ser utilizada nas mais diversas situações do dia a dia mesmo àqueles que não têm o intuito de desenvolver seus estudos na área das Ciências Exatas. Em parte, é esperado que a pergunta “para que serve isso?” seja respondida e, com essa resposta, o estudo da matemática no Ensino Médio possa ser visto como algo mais abrangente do que, simplesmente, uma disciplina a ser cobrada no vestibular.

CAPÍTULO 1

ORIGEM DA TEORIA: O PROBLEMA DAS SETE PONTES DE KÖNIGSBERG

O conceito de grafos remonta ao século XVIII com o famoso problema das Pontes de Königsberg, resolvido pelo célebre físico e matemático Leonhard Euler (1707-1783) [4].

Fundada em 1255 pelos Cavaleiros Teutónicos, Königsberg (atualmente Kaliningrado, Rússia) tinha seu centro dividido pelo Rio Pregel (hoje, Rio Pregolya) em quatro regiões que eram ligadas por sete pontes, conforme mostra a figura 1.1.

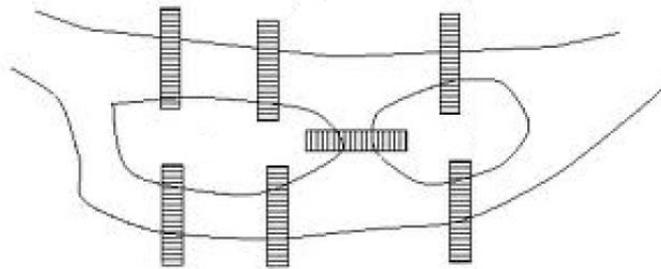


Figura 1.1: As Sete Pontes de Königsberg.

Conta a história que os habitantes dessa cidade queriam saber se era possível atravessar cada uma das sete pontes uma única vez e, ao final, retornar ao local de partida. Euler, em 1736, provou que não existia caminho que possibilitasse tais restrições. Para tanto, usou um raciocínio muito simples. Nomeou as margens e ilhas, transformou as pontes em linhas e suas intersecções em pontos, criando, possivelmente, o primeiro grafo da história, representado na figura 1.2.

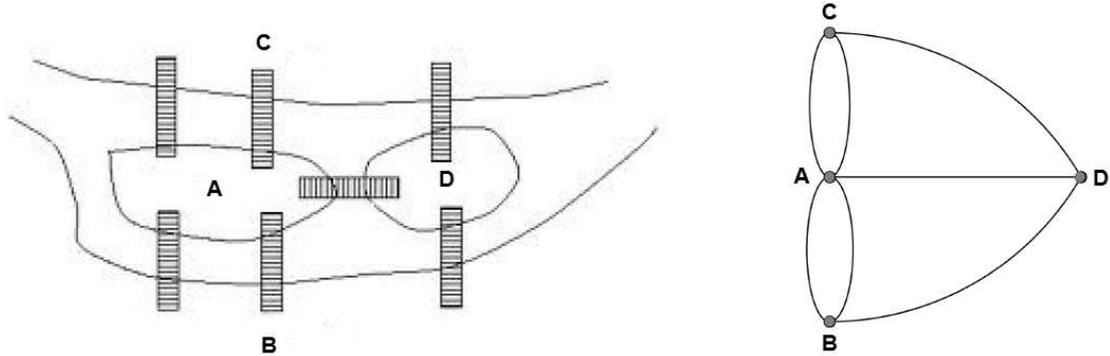


Figura 1.2: As Sete Pontes de Königsberg e seu grafo associado.

1.1 Definições básicas

De um modo geral, um grafo é um conjunto finito de pontos, chamados de vértices do grafo, e um conjunto finito de arcos, chamados de arestas do grafo. As extremidades de cada aresta devem ser vértices. Além disso, duas arestas quaisquer do grafo não podem ter pontos interiores em comum: ou são disjuntas ou se tocam apenas numa ou em duas das extremidades. Dependendo da aplicação, as arestas podem ser direcionadas e representadas por setas.

Definição 1.1.1. Um grafo G é definido por dois conjuntos finitos: um conjunto não-vazio, V , de elementos chamados vértices ou nós e um conjunto, que pode ser vazio, A , de elementos chamados arcos ou arestas, onde cada aresta associa-se a dois vértices.

Um digrafo é um grafo direcionado onde cada aresta a é associada a um par ordenado (i, j) de nós, com i sendo o início e j , o fim de a .

O número de vértices de um grafo será representado por $|V|$ e seu número de arestas por $|A|$.

É comum representar um grafo na forma de diagrama, como apresentado na figura 1.3.

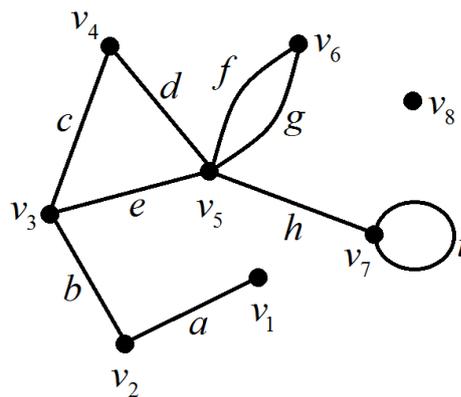


Figura 1.3: Representação gráfica de um grafo.

Grafos também podem ser representados por matrizes, o que é conveniente, por exemplo, para fins computacionais [5]. Uma das maneiras de fazer isso é através de uma *matriz de adjacência*. O procedimento é o seguinte:

- 1) Nomear os vértices.
- 2) Construir uma tabela (matriz) cujos rótulos das linhas e das colunas sejam os nomes dos vértices.
- 3) Em cada célula é posto o número de arestas que há entre o vértice da linha e o vértice da coluna.

A figura 1.3, por uma matriz de matriz de adjacência, fica assim representada:

$$\begin{array}{c}
 v_1 \\
 v_2 \\
 v_3 \\
 v_4 \\
 v_5 \\
 v_6 \\
 v_7 \\
 v_8
 \end{array}
 \begin{bmatrix}
 v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 & v_8 \\
 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\
 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
 0 & 0 & 1 & 1 & 0 & 2 & 1 & 0 \\
 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
 \end{bmatrix}$$

Formalmente, uma matriz de adjacência é uma matriz de dimensões $|V| \times |V|$ onde a entrada $M(i, j)$ é o número de arestas que ligam os vértices v_i e v_j .

$$M_A = (a_{ij})_{|V| \times |V|} = \begin{cases} \alpha, & \text{sendo } \alpha \text{ o número de aretas } (i, j) \in A. \\ 0, & \text{se } (i, j) \notin A. \end{cases}$$

Outra maneira é através de uma *matriz de incidência*. Neste caso cada elemento da matriz indica se a aresta incide sobre determinado vértice. O procedimento é o seguinte:

- 1) Nomear os vértices e aretas.
- 2) Construir uma tabela (matriz) cujos rótulos das linhas sejam os nomes dos vértices e das colunas sejam os nomes das aretas.
- 3) Em cada célula é colocado o número 1 se o vértice i incide sobre a aresta j e 0, caso contrário.

A figura 1.3, por uma matriz de matriz de incidência, fica assim representada:

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>	<i>i</i>
v_1	1	0	0	0	0	0	0	0	0
v_2	1	1	0	0	0	0	0	0	0
v_3	0	1	1	0	1	0	0	0	0
v_4	0	0	1	1	0	0	0	0	0
v_5	0	0	0	1	1	1	1	1	0
v_6	0	0	0	0	0	1	1	0	0
v_7	0	0	0	0	0	0	0	1	1
v_8	0	0	0	0	0	0	0	0	0

Definição 1.1.2. Uma aresta $i-j$ é incidente nos vértices i e j . Analogamente, é dito que um vértice é incidente nas aretas às quais está associado.

Por exemplo, na figura 1.3 a aresta b é incidente nos vértices v_2 e v_3 e o vértice v_2 é incidente nas aretas a e b .

Definição 1.1.3. Duas aretas incidentes num mesmo vértice são chamadas *arestas adjacentes* e dois vértices incidentes numa mesma aresta são *vértices adjacentes*.

Na figura 1.3 a aresta b é adjacente às aretas a , c e e ; o vértice v_2 é adjacente a v_1 e a v_3 .

A *vizinhança* de um vértice v é o conjunto dos vértices adjacentes a v e denotada por $N_G(v)$. Também é possível definir uma vizinhança onde v está incluído, chamada de *vizinhança fechada* e denotada por $N_G[v]$.

Definição 1.1.4. O grau de um vértice é o número de arestas incidentes neste vértice.

Denota-se grau de um vértice v por $d(v)$. Dessa maneira, na figura 1.3 tem-se, por exemplo, $d(v_4) = 2$.

Quando um vértice possui grau zero é chamado *vértice isolado*; um vértice de grau 1 é dito *vértice pendente*; um *vértice ímpar* possui um número ímpar de arestas e um *vértice par* possui um número par de arestas.

Definição 1.1.5. Define-se grau de um grafo como sendo a soma do grau de cada um dos vértices que o compõem.

Na figura 1.3 tem-se $\sum_{i=1}^8 n(v_i) = 1 + 2 + 3 + 2 + 5 + 2 + 3 + 0 = 18$; assim, diz-se

que esse grafo possui grau 18 e esse resultado será denotado por $\text{grau}(G) = 18$.

Teorema 1.1.6. Todo grafo possui grau par dado por

$$\sum_{v \in V(G)} n(v) = 2 \cdot |A|$$

Demonstração: Quando os graus dos vértices são contados, na verdade contam-se as extremidades das arestas uma vez. Cada aresta tem duas extremidades, portanto, cada aresta é contada duas vezes.

Definição 1.1.7. Um grafo que possui um único vértice e não possui arestas é chamado de *grafo nulo*.



Figura 1.4: Grafo nulo.

Definição 1.1.8. A aresta que parte de um vértice e retorna ao mesmo vértice recebe o nome de *laço*.

Na figura 1.3 a aresta i é um laço; observe que um laço conta duas vezes para o grau de um vértice.

Definição 1.1.9. Quando duas ou mais arestas forem incidentes a um mesmo par de vértices elas serão chamadas de *arestas em paralelo*. Neste caso o grafo recebe o nome de *multigrafo*.

Na figura 1.3 as arestas f e g estão em paralelo.

Definição 1.1.10. Um grafo simples é um grafo sem laços e sem arestas em paralelo. Observe que, num grafo simples e não orientado, a matriz de adjacência é

$$M_A = (a_{ij})_{|V| \times |V|} = \begin{cases} 1, & \text{se existe aresta entre os vértices } i \text{ e } j. \\ 0, & \text{caso contrário.} \end{cases}$$

Definição 1.1.11. Um grafo é chamado de *grafo regular* se todos os seus vértices possuírem o mesmo grau, ou seja, se em todos os vértices incidirem o mesmo número de arestas. Se o grau dos vértices for r , é dito que o grafo é regular de grau r ou ainda, que o grafo é r -regular.

Definição 1.1.12. Um grafo é completo quando existe uma aresta entre dois vértices quaisquer do grafo. Ou, em outras palavras, todos os seus vértices são adjacentes. O grafo completo de n vértices é denotado por K_n . Essa notação é uma homenagem ao matemático polonês Kasemir Kuratowski que foi o primeiro a obter, em 1930, uma caracterização de planaridade por meio deste tipo de grafo.

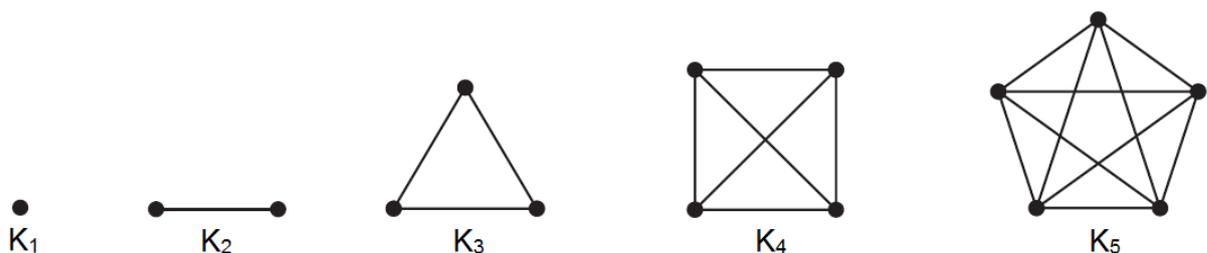


Figura 1.4: Grafos completos.

Definição 1.1.13. Um grafo planar é um grafo que admite uma representação gráfica em que as arestas só se encontrem (possivelmente) nos vértices a que são incidentes. Caso contrário o grafo é dito não-planar.

Apesar de definir planaridade através do desenho, não bastaria a observação do mesmo para caracterizar um grafo como planar ou não. Um grafo cujas arestas se cruzem não é, necessariamente, não-planar. Pode ocorrer que haja outra forma de desenhar o mesmo grafo de tal forma que suas arestas não se cruzem. Como exemplo veja a figura 1.5. Nela é representado o grafo K_4 em várias configurações, das quais apenas uma possui cruzamento entre as arestas.

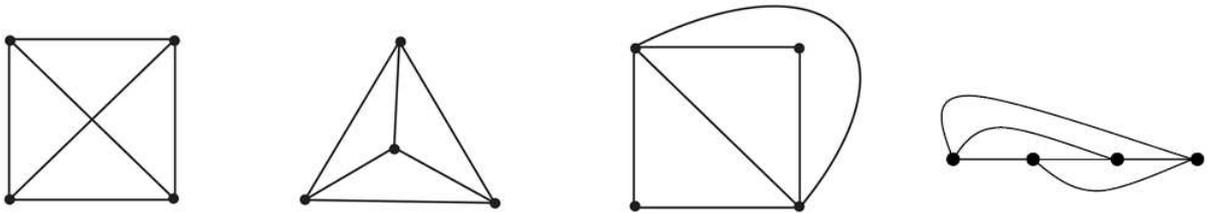


Figura 1.5: Grafo K_4 e sua representação plana.

Definição 1.1.14. Dois grafos, G e H , são *isomorfos* se existir uma correspondência biunívoca entre os vértices de G e os vértices de H que preserve a relação de adjacência entre vértices e arestas.

Em outras palavras, é possível obter o grafo H a partir de uma nova rotulação dos vértices de G .

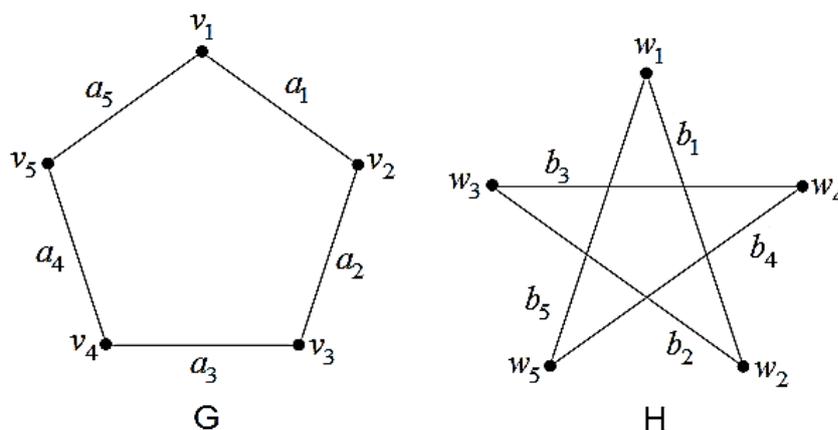


Figura 1.6: Grafos isomorfos.

Definição 1.1.15. Sendo G um grafo, define-se como um *passeio* (walk) ou *percurso*, uma sequência finita de vértices e arestas. Cada vértice da sequência é incidente na aresta que o precede e que o sucede. Essa sequência deve iniciar e acabar em algum vértice. Quando o vértice inicial e final são diferentes, tem-se um passeio aberto; quando o vértice final e inicial são os mesmos, tem-se um passeio fechado. Uma *trilha* (trail) é um passeio onde não se repetem arestas. Se o vértice final coincidir com o inicial, tem-se uma trilha fechada. Um *caminho simples* (path) é uma trilha sem repetição de vértices. Quando o vértice inicial coincide com o final, tem-se um *caminho fechado*, um *circuito* ou um *ciclo*. O comprimento de um caminho num grafo não ponderado é igual ao número de arestas que compõem o caminho. Num grafo ponderado o comprimento de um caminho é a soma dos pesos de cada aresta do caminho.

Um grafo é *conexo* se, para cada dois quaisquer de seus vértices, v_i e v_j , existe, no grafo, um caminho, tendo v_i como vértice inicial e v_j como vértice final.

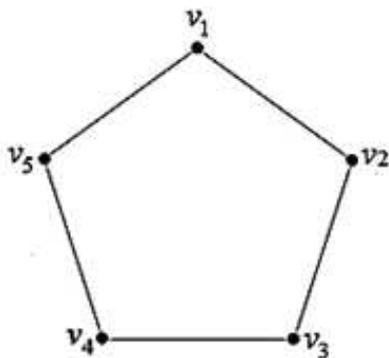


Figura 1.7: Grafo conexo.

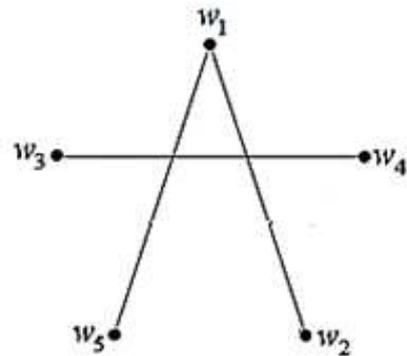


Figura 1.8: Grafo não conexo.

Um grafo admite um “*passeio de Euler*” se existe, nesse grafo, um caminho do qual fazem parte todas as arestas do grafo e onde um ponto móvel pode “passear” pelas arestas do grafo, percorrendo todas elas uma única vez. Se esse caminho começa e termina num mesmo vértice então tal caminho recebe o nome de *Ciclo Euleriano* e, nesse caso, é dito que o grafo é um *Grafo Euleriano*.

Definição 1.1.16. Um grafo conexo que não contém ciclos é chamado de *árvore*. Um grafo que não contém ciclos é uma floresta, ou seja, uma floresta é uma reunião disjunta de uma ou mais árvores.

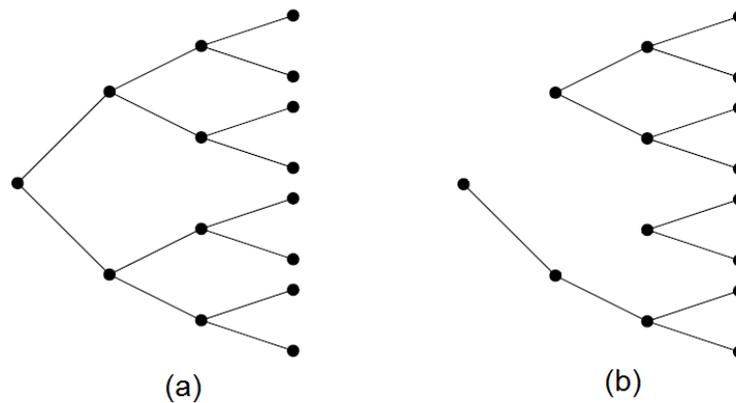


Figura 1.9: Uma árvore (a) e uma floresta formada por três árvores (b).

Definição 1.1.17. Uma *árvore geradora* de um grafo G é um grafo que contém cada vértice de G e é uma árvore.

1.2 A solução do problema das Pontes de Königsberg

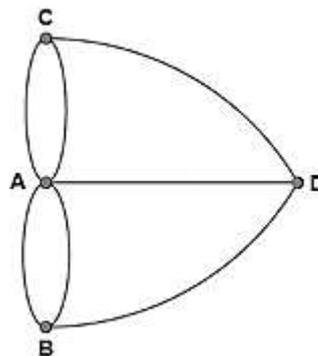


Figura 1.9: Grafo das Pontes de Königsberg.

Analisando a figura 1.9, que foi a representação utilizada por Euler, é fácil perceber que em cada vértice chegam três ou cinco arestas, isto é, um número ímpar de arestas chega em cada vértice. Considere, por exemplo, um vértice com três arestas, como B. A primeira vez que se chegar a esse vértice pode-se sair por algum dos dois restantes, mas, quando chegar novamente nesse vértice pelo terceiro arco, não há como sair dele sem retrair uma aresta.

Existiriam duas possibilidades que dariam certo:

- 1) se esse vértice com três arestas fosse o último. Nesse caso, as outras duas arestas serviriam apenas para atravessar uma ponte e voltar pela outra.
- 2) se se começasse por esse vértice e depois, em um momento posterior, chegasse de volta a ele, saísse e não mais voltasse.

Portanto, um vértice com um número ímpar de arestas tem de ser o primeiro ou o último da trajetória. Isto é, pode haver, no máximo, dois vértices com um número ímpar de arestas ligadas a eles. No caso das pontes de Königsberg, existem quatro vértices com um número ímpar de arestas, logo, esse caminho não é possível.

CAPÍTULO 2

GRAFOS HAMILTONIANOS

A teoria dos grafos contou ainda com o importante auxílio do matemático, físico e astrônomo irlandês Sir William Rowan Hamilton (1805 - 1865) que, ao inventar um jogo simples, chamado “The Icosian Game”, deu origem ao estudo de grafos Hamiltonianos [32].



Figura 2.1: The Icosian Game.

O jogo foi proposto na forma de um dodecaedro onde cada vértice recebeu o nome de uma importante cidade. O problema: é possível começar em uma cidade, visitar todas as outras uma única vez e retornar à cidade de partida?

O jogo é mais fácil de ser imaginado projetando o dodecaedro no plano:

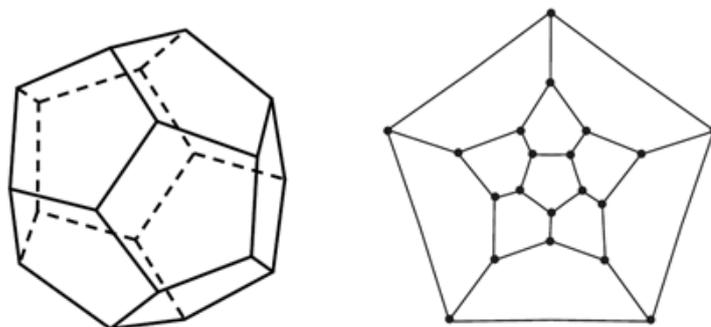


Figura 2.2: Dodecaedro e seu grafo associado.

2.1 Caminhos e ciclos hamiltonianos

Definição 2.1.1. Um caminho que contém todos os vértices de um grafo G é dito *Caminho Hamiltoniano* de G . Do mesmo modo, um ciclo que contém todos os vértices de G é chamado *Ciclo Hamiltoniano* ou *Circuito Hamiltoniano*. Se G contém um ciclo hamiltoniano, então G é um *Grafo Hamiltoniano*.

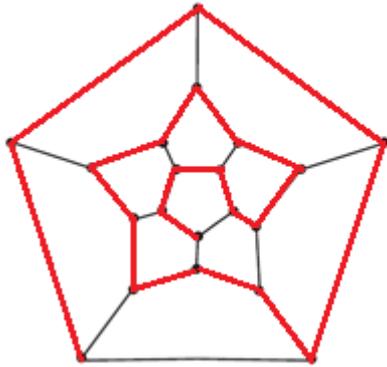


Figura 2.3: Caminho Hamiltoniano.

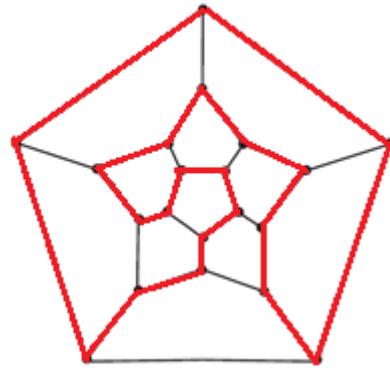


Figura 2.4: Ciclo Hamiltoniano.

Verificar se um grafo é hamiltoniano é bem diferente e muito mais difícil de verificar se é euleriano. Apesar de terem sido estudados por vários séculos, até hoje não há uma boa caracterização dos grafos hamiltonianos. Há diversas famílias de grafos para os quais existe um circuito hamiltoniano e também é possível estabelecer certas condições que implicam na não-existência de um circuito, mas uma caracterização geral ainda não foi encontrada.

É muito fácil se convencer da existência de um ciclo hamiltoniano em um grafo: basta exibir tal caminho. Tome, por exemplo, o ciclo da figura 2.4. No entanto, é difícil, em geral, se convencer da não-existência de tal ciclo. Por exemplo, o grafo da figura 2.5 não possui um ciclo hamiltoniano, mas não existe um argumento simples e geral para demonstrar esse fato.

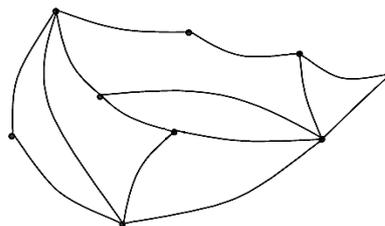


Figura 2.5: Grafo não hamiltoniano.

A questão da existência ou não de um ciclo hamiltoniano para um determinado grafo é mais difícil que sua contrapartida euleriana [23]. Não há qualquer teorema que decida definitivamente a questão. A experiência sugere que quanto mais arestas melhor, no entanto, a distribuição destas arestas entre os vértices também é importante.

Os teoremas a seguir fornecem condições suficientes para a existência de um ciclo hamiltoniano [30]. Isto é, se as condições do enunciado forem satisfeitas por um certo grafo G , então ele admitirá um ciclo hamiltoniano. No entanto, essas condições não são necessárias, ou seja, há grafos que não satisfazem às hipóteses e admitem ciclos hamiltonianos, como os representados na figura 2.8.

Teorema 2.1.2. (Dirac, 1952) Um grafo conexo, simples com n vértices, $n \geq 3$, possui um circuito hamiltoniano se o grau de cada vértice é pelo menos $n/2$.

Demonstração: Suponha que G não seja hamiltoniano. Então existe um grafo simples não hamiltoniano maximal G de ordem $n \geq 3$ que satisfaz a condição do teorema, ou seja, G é não hamiltoniano, mas para qualquer par de vértices não adjacentes x, y em G , tem-se que o grafo $G + xy$ é hamiltoniano. Claramente G não é completo pois todo grafo completo com pelo menos 3 vértices é obviamente hamiltoniano. Portanto, existem vértices x e y não adjacentes em G . Considere o grafo $H = G + xy$. Pela maximalidade de G , segue que H é hamiltoniano. Logo, todo circuito hamiltoniano em H deve conter a aresta xy . Assim, G tem um caminho hamiltoniano $P = (x = v_1, v_2, \dots, v_n = y)$.

Sejam

$$U = \{v_i : v_i \text{ é adjacente a } x\},$$

$$W = \{v_i : v_{i-1} \text{ é adjacente a } y\}.$$

Nessas condições, $U \cap W = \emptyset$ pois, caso contrário $C = (v_1, v_i, v_{i+1}, \dots, v_n, v_{i-1}, v_{i-2}, \dots, v_1)$, que contém os vértices de P , seria um circuito hamiltoniano em G , contrariando a escolha de G . Por outro lado, como $x \notin U \cup W$, resulta que $|U \cup W| < n$. Logo,

$$n > |U \cup W| = |U| + |W| - |U \cap W| = d(x) + d(y) \geq \frac{n}{2} + \frac{n}{2} = n.$$

Uma contradição.

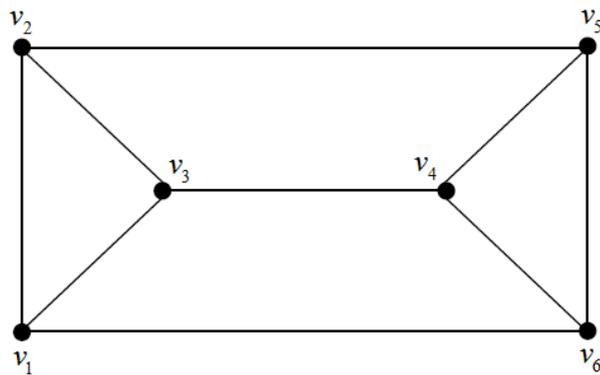


Figura 2.6: Grafo hamiltoniano (Dirac).

Observe, na figura 2.6, que:

$$n = 6$$

$$d(v_1) = d(v_2) = d(v_3) = d(v_4) = d(v_5) = d(v_6) = 3$$

Então,

$$d(v) = 3 \quad \forall v \in V(G) \quad \text{e} \quad \frac{n}{2} = \frac{6}{2} = 3$$

Logo, G é hamiltoniano pelo Teorema de Dirac.

A prova do Teorema 2.1.2 motivou o seguinte resultado, cuja prova segue analogamente. Note que o Teorema 2.1.2 é um corolário do Teorema 2.1.3.

Teorema 2.1.3. (Ore, 1960) Seja G um grafo conexo, simples com n vértices, $n \geq 3$. Se $d(x) + d(y) \geq n$, para todo par de vértices x e y não adjacentes, $x, y \in V(G)$ tem-se que G é um grafo hamiltoniano.

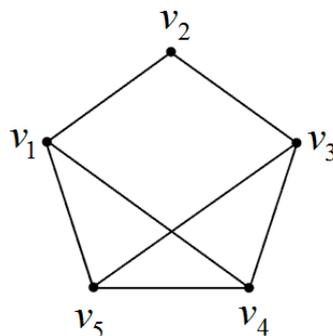


Figura 2.7: Grafo hamiltoniano (Ore).

Observe, na figura 2.7, que:

$$n = 5$$

$$d(v_1) = d(v_3) = d(v_4) = d(v_5) = 3$$

$$d(v_2) = 2 < \frac{5}{2} = 2,5$$

Logo, não vale o Teorema de Dirac. Mas,

$$d(v_1) + d(v_3) = 6 > n = 5$$

$$d(v_2) + d(v_4) = d(v_2) + d(v_5) = 5 = n$$

Assim, G é hamiltoniano pelo Teorema de Ore.

Esses teoremas não são muito informativos. O que eles dizem, intuitivamente, é que se um grafo contém muitas arestas e se elas são bem distribuídas, ele é hamiltoniano. No limite, tem-se um grafo completo.

Definição 2.1.4. Um grafo simples com n vértices, $n \geq 3$, é denominado um *Grafo Ciclo*, e denotado por C_n , quando seus vértices são $v_1, v_2, v_3, \dots, v_{n-1}, v_n$ e arestas são apenas as arestas $v_1v_2, v_2v_3, \dots, v_{n-1}v_n, v_nv_1$.

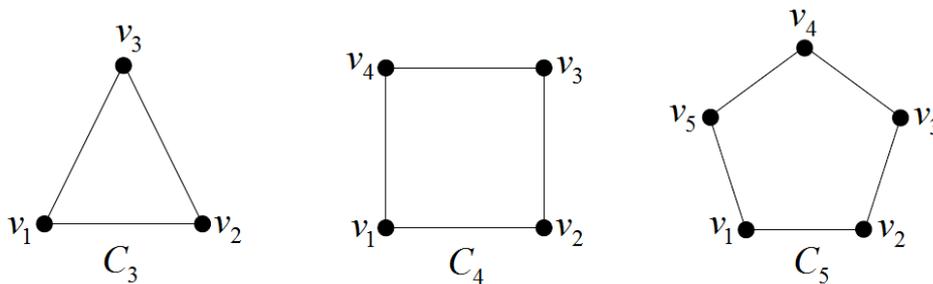


Figura 2.8: Alguns grafos ciclo.

Considere o grafo C_5 apresentado na figura 2.8. Nele, tem-se que:

$$n = 5$$

$$d(v_1) = d(v_2) = d(v_3) = d(v_4) = d(v_5) = 2 < \frac{n}{2} = \frac{5}{2} = 2,5$$

Logo, não vale o Teorema de Dirac.

E ainda

$$d(v_1) + d(v_3) = d(v_1) + d(v_4) = d(v_1) + d(v_3) = d(v_2) + d(v_4) = d(v_2) + d(v_5) = \\ = d(v_3) + d(v_5) = 4 < n = 6$$

Logo, não vale o Teorema de Ore.

Veja que o grafo C_5 não satisfaz as condições do Teorema de Dirac e nem do Teorema Ore e, no entanto, é hamiltoniano.

Com base no Teorema de Ore, Bondy e Chvátal modificaram a hipótese de modo a propor um resultado mais efetivo.

Teorema 2.1.5. (Bondy–Chvátal, 1976) Seja G um grafo de ordem $n \geq 3$, e sejam dois vértices não adjacentes, x e y , tais que $d(x) + d(y) \geq n$. O grafo $G + xy$ é hamiltoniano se, e somente se, o grafo G for hamiltoniano.

Demonstração:

(\Leftarrow) Se o grafo G é hamiltoniano, significa que contém um ciclo hamiltoniano. O acréscimo de uma aresta não mudará essa condição, de modo que essa parte da demonstração é imediata.

(\Rightarrow) Esta parte da demonstração será feita por redução ao absurdo. Suponha que $G + xy$ seja hamiltoniano, mas G , não. Assim, $G + xy$ possui um ciclo hamiltoniano contendo a aresta xy , ou seja, existe um caminho $P = (x_1, \dots, x_n)$ em G partindo de $x_1 = x$ até $x_n = y$, passando por todos os vértices de G . Agora, tome um vértice x_i qualquer. Se x_i for adjacente a x_1 , ($2 \leq i \leq n$), então x_{i-1} não poderá ser adjacente a x_n , pois, se assim fosse, $x_1 x_i x_{i+1} \dots x_n x_{i-1} x_{i-2} \dots x_1$ seria um ciclo hamiltoniano em G . Mas, com isso, os vértices conectados a $x_n = y$ podem ser todos os outros exceto aqueles conectados a $x_1 = x$, num total máximo de $n-1$. Expressando em termos de graus: $d(x_n) \leq (n-1) - d(x_1)$. Mas isso significa que $d(x) + d(y) \leq n-1$, contrariando uma das hipóteses do teorema. Logo, por absurdo, conclui-se que G deve ter um ciclo hamiltoniano, e, portanto, G é um grafo hamiltoniano.

Vamos, agora, passar a um problema prático intimamente ligado aos conceitos vistos até o momento: o problema do caixeiro viajante.

2.2 O Problema do Caixeiro Viajante

O Problema do Caixeiro Viajante (PCV) é um dos problemas mais estudados no campo da pesquisa operacional, mas até hoje não foi encontrado um algoritmo computacionalmente eficiente para resolvê-lo [25]. Consiste na procura por um circuito que possua a menor distância, que, partindo de uma cidade, visita cada cidade precisamente uma vez e regressa à cidade inicial. De forma geral, sua formulação é: dado um grafo completo valorado G , deseja-se determinar o valor do menor ciclo hamiltoniano de G .

Definição 2.2.1. Um grafo é valorado ou ponderado quando suas arestas possuírem pesos.

Pode-se representar um grafo simples, valorado e não direcionado por uma matriz, de maneira análoga a matriz de adjacência apresentada no capítulo 1.

$$M = (a_{ij})_{|V| \times |V|} = \begin{cases} \beta, & \text{sendo } \beta \text{ o peso da aresta } (i, j). \\ 0, & \text{se não houver aresta entre } i \text{ e } j. \end{cases}$$

Considere um vendedor que resida numa cidade **A** e que possua clientes em sua própria cidade e também em quatro cidades vizinhas: **B**, **C**, **D** e **E**. Para planejar uma viagem afim de atender a todos os clientes passando por cada uma das cidades uma única vez e retornando à cidade inicial, de forma a minimizar a distância percorrida, criou o grafo valorado representado na figura 2.9, onde o peso de cada aresta representa a distância, em quilômetros, entre cada cidade.

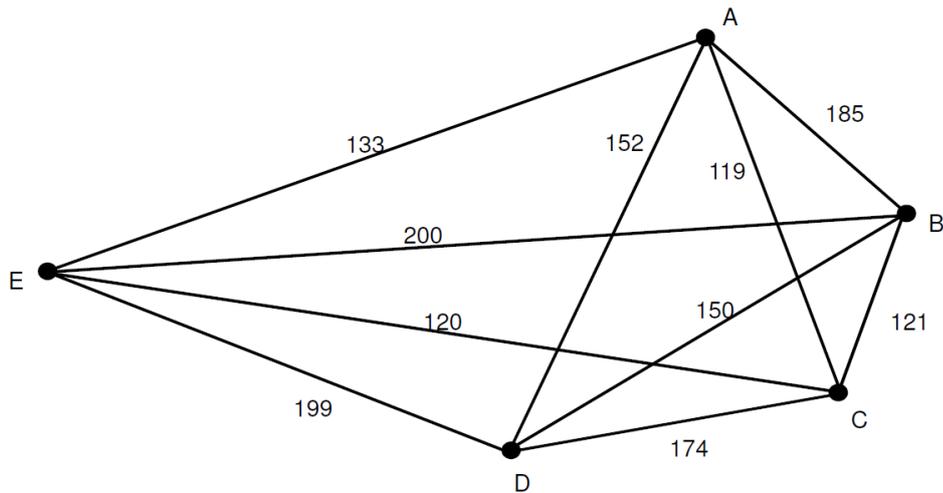


Figura 2.9: Cidades e suas distâncias.

Matricialmente, a situação acima pode ser descrita pela matriz:

$$M = \begin{bmatrix} 0 & 185 & 119 & 152 & 133 \\ 185 & 0 & 121 & 150 & 200 \\ 119 & 121 & 0 & 174 & 120 \\ 152 & 150 & 174 & 0 & 199 \\ 133 & 200 & 120 & 199 & 0 \end{bmatrix}$$

Como tal vendedor possui clientes em apenas cinco cidade, é possível calcular todas as possíveis distâncias utilizando o chamado *Método Exaustivo*.

2.2.1 Método Exaustivo

Partindo de A , o vendedor possui 4 opções para escolher a próxima cidade. Feita essa escolha, restam-lhe 3 opções para a seguinte e, assim, sucessivamente. Portanto, o número de trajetos possíveis é $4 \cdot 3 \cdot 2 \cdot 1 = 24 = 4!$.

De forma geral, um caminho em Kn é determinado por uma sequência de vértices distintos. Para ser um ciclo hamiltoniano, todos os vértices devem ocorrer na sequência, e o último vértice deverá ser igual ao primeiro. Logo, existem $n!$ sequências distintas. No entanto, num ciclo não interessa qual é o vértice inicial, já que se pode começar em qualquer vértice. Assim, fixando o vértice inicial obtém-se

um total de $(n-1)!$ ciclos hamiltonianos distintos em Kn . Considerando os casos em que não interessa o sentido em que o ciclo é percorrido, pode-se identificar um ciclo com o seu *ciclo inverso*, obtendo-se um total de $\frac{(n-1)!}{2}$ ciclos hamiltonianos em Kn .

	Ciclo Hamiltoniano	Distância Percorrida (km)	Ciclo Inverso
1	A,B,C,D,E,A	$185 + 121 + 174 + 199 + 133 = 812$	A,E,D,C,B,A
2	A,B,C,E,D,A	$185 + 121 + 120 + 199 + 152 = 777$	A,D,E,C,B,A
3	A,B,D,C,E,A	$185 + 150 + 174 + 120 + 133 = 762$	A,E,C,D,B,A
4	A,B,D,E,C,A	$185 + 150 + 199 + 120 + 119 = 773$	A,C,E,D,B,A
5	A,B,E,C,D,A	$185 + 200 + 120 + 174 + 152 = 831$	A,D,C,E,B,A
6	A,B,E,D,C,A	$185 + 200 + 199 + 174 + 119 = 877$	A,C,D,E,B,A
7	A,C,B,D,E,A	$119 + 121 + 150 + 199 + 133 = 722$	A,E,D,B,C,A
8	A,C,B,E,D,A	$119 + 121 + 200 + 199 + 152 = 791$	A,D,E,B,C,A
9	A,C,D,B,E,A	$119 + 174 + 150 + 200 + 133 = 776$	A,E,B,D,C,A
10	A,C,E,B,D,A	$119 + 120 + 200 + 150 + 152 = 741$	A,D,B,E,C,A
11	A,D,B,C,E,A	$152 + 150 + 121 + 120 + 133 = 676$	A,E,C,B,D,A
12	A,D,C,B,E,A	$152 + 174 + 121 + 200 + 133 = 780$	A,E,B,C,D,A

Tabela 2.1: Enumeração dos ciclos hamiltonianos.

Observando-se os resultados, são identificados dois *Ciclos Ótimos de Hamilton*: A – D – B – C – E – A e sua rota inversa, A – E – C – B – D – A, cujas distâncias são 676 km.

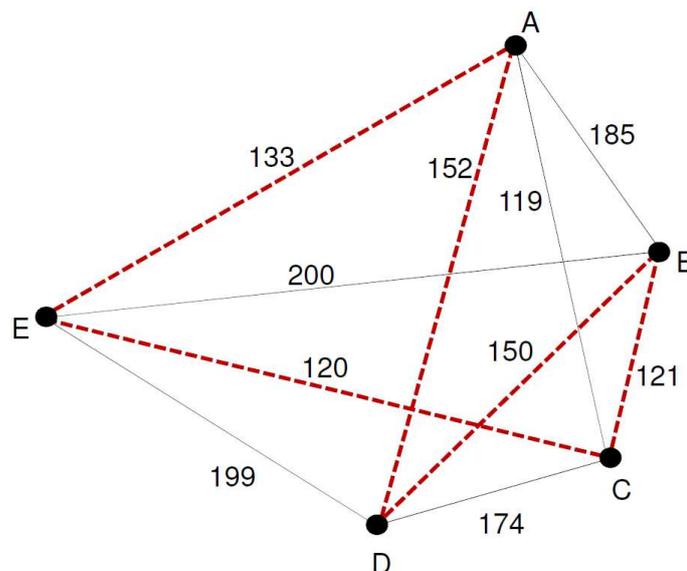


Figura 2.10: Ciclo ótimo de Hamilton.

O problema desse método surge quando o número de cidades cresce. Considere, por exemplo, uma rota que possua apenas 15 cidades. O número de ciclos possíveis é $14! = 87.178.291.200$, que seriam determinados em, aproximadamente, 0,09 segundo por uma máquina, cujo desempenho computacional possua ordem de grandeza em teraflop/s, equipada com um programa que conseguisse examinar um trilhão de ciclos por segundo. Agora, considere que a rota possua 25 cidades, que pode ser considerado um número bastante conservador. Essa mesma máquina teria que determinar os $24! = 620.448.401.733.239.439.360.000$ ciclos possíveis para encontrar o resultado ótimo. Mantendo seu poder de processamento, levaria mais de 19.000 anos para encontrar a melhor rota. Esse crescimento exponencial do tempo inviabiliza a determinação de todos os ciclos existentes. Por esse motivo, foram desenvolvidas técnicas, chamadas *heurísticas de construção* [6], para se determinar um ciclo sem oferecer garantias de que a solução seja ótima, mas que seja viável, especialmente no que diz respeito ao consumo de tempo para encontrar soluções de boa qualidade.

2.2.2 Método do Vizinho Mais Próximo

Esse método (algoritmo heurístico) consiste em, a partir de determinada cidade (vértice), escolher, dentre as rotas (arestas) adjacentes, a que tem menor distância (peso) e partir para a cidade correspondente. Basta continuar a construir o circuito, uma cidade de cada vez, sempre indo para a cidade que representa o vizinho mais próximo e que ainda não tenha sido visitada. Deve-se prosseguir dessa maneira até que todas as cidades tenham sido visitadas. Chegando à última cidade, basta retornar à inicial.

Para o caso apresentado anteriormente e representado pela figura 2.9, o ciclo seria construído seguindo os seguintes passos:

P1: partindo de *A*, seguir para *C*;

P2: de *C*, seguir para *E*;

P3: de *E*, seguir para *D* (a menor distância seria entre *E* e *A*, mas isso fecharia o circuito sem que todas as cidades fossem visitadas);

P4: de *D*, partir para *B*;

P5: retornar a A.

A partir desse método o ciclo criado foi A – C – E – D – B – A, totalizando 773 quilômetros. A melhor rota tem comprimento 676 quilômetros. Assim, encontra-se uma rota com 97 quilômetros a mais que a ideal, um erro relativo de $\frac{97}{676} \approx 14,35\%$.

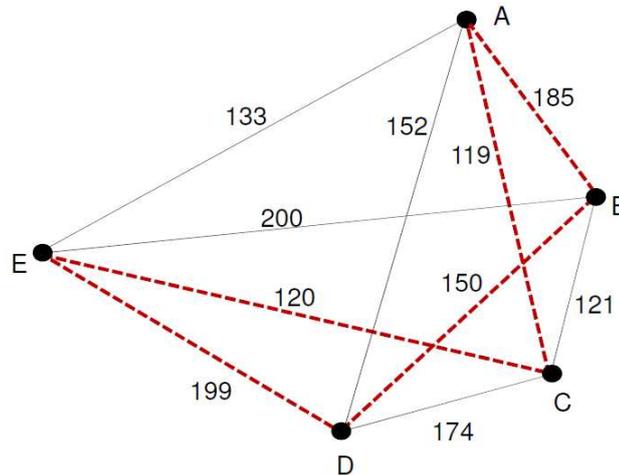


Figura 2.11: Vizinho mais próximo - trajeto final.

Analisando os resultados encontrados, é possível perceber que o método exaustivo (ou Força Bruta) é um exemplo do que é formalmente conhecido como um *algoritmo ineficiente*, que é um algoritmo cujo número de passos necessários à obtenção de uma solução cresce desproporcionalmente com o tamanho do problema.

Por outro lado, o método do vizinho mais próximo é um exemplo de *algoritmo heurístico eficiente*, pois, a cada passo há uma melhor escolha a ser feita com base num critério apropriado e bem conhecido. No entanto, o resultado pode estar distante de ser um circuito ótimo de Hamilton. Devido a essa situação, outros métodos (algoritmos heurísticos) foram desenvolvidos.

2.2.3 Método da Aresta de Menor Peso

Também chamado *Algoritmo Heurístico da Ligação mais Econômica*, tal método é bastante simples; primeiramente a aresta de menor comprimento do grafo deve ser escolhida. Utilizando o mesmo princípio deve-se escolher outra aresta dentre as restantes e, assim, sucessivamente. Vale ressaltar que as arestas não precisam

ser adjacentes, mas, na escolha da próxima aresta, não escolher uma aresta que forme um ciclo quando houver vértices ainda a visitar e não permitir que três ou mais arestas sejam incidentes a um mesmo ponto, afinal, um ciclo hamiltoniano usa exatamente duas arestas por vértice.

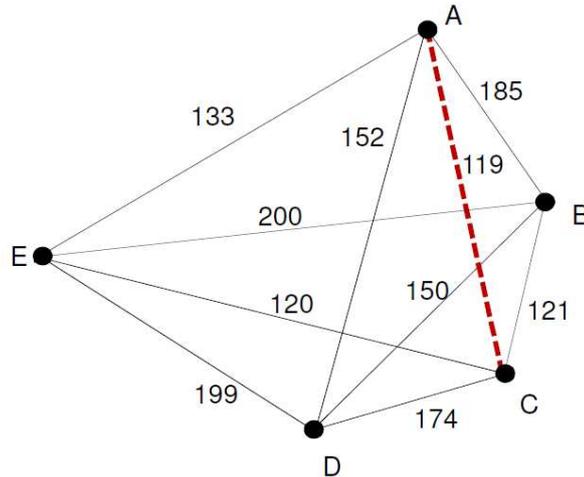


Figura 2.12: Método da aresta de menor peso - 1ª decisão.

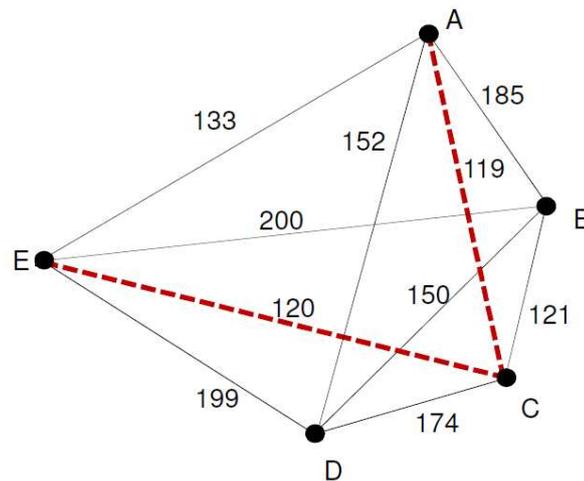


Figura 2.13: Método da aresta de menor peso - 2ª decisão.

A aresta de menor peso entre as restantes é $BC = 121$ km, porém ela não pode ser escolhida pois C teria três arestas incidentes. A próxima escolha seria a aresta $AE = 133$ km, mas também não deve ser escolhida pois os vértices B e D ainda não foram visitados e fecharia o ciclo $A - C - E - A$.

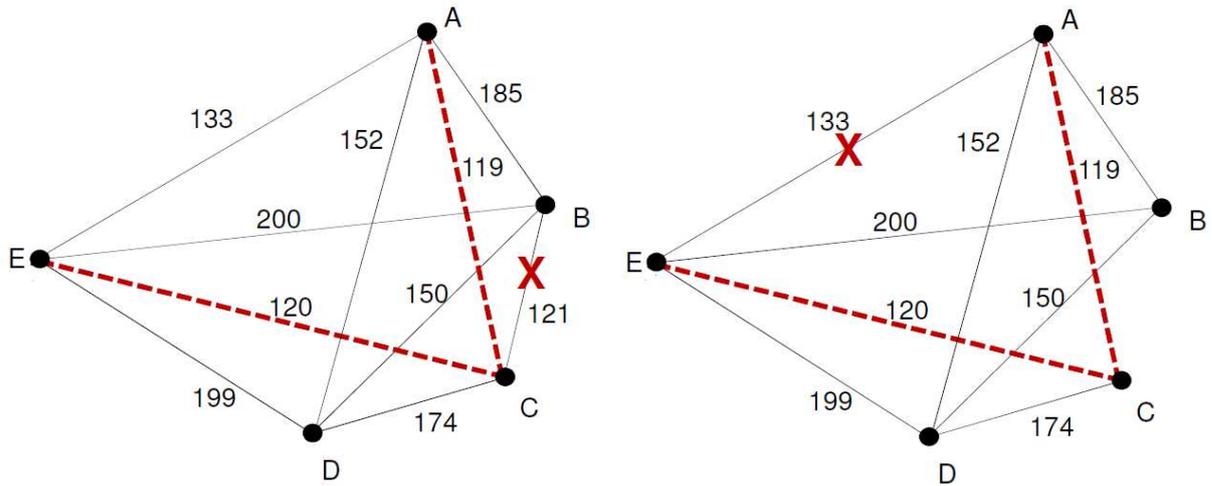


Figura 2.14: Método da aresta de menor peso - restrições de escolhas.

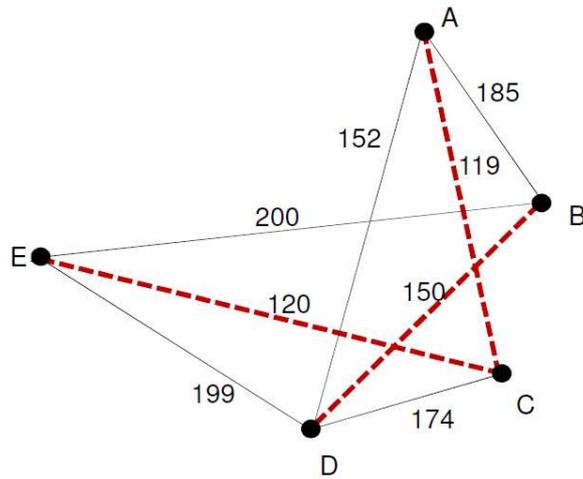


Figura 2.15: Método da Aresta de Menor Peso - 3ª decisão.

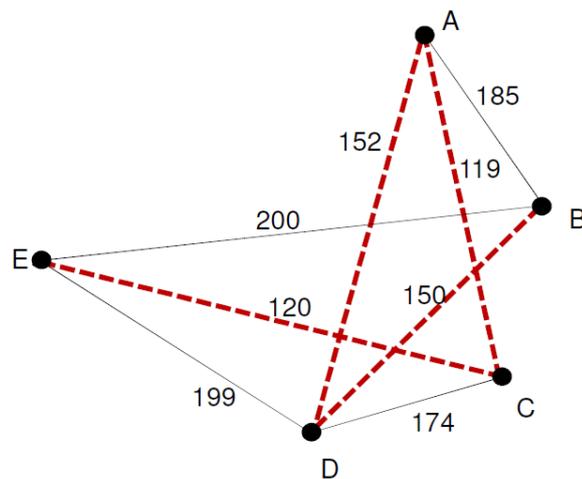


Figura 2.16: Método da Aresta de Menor Peso - 4ª decisão.

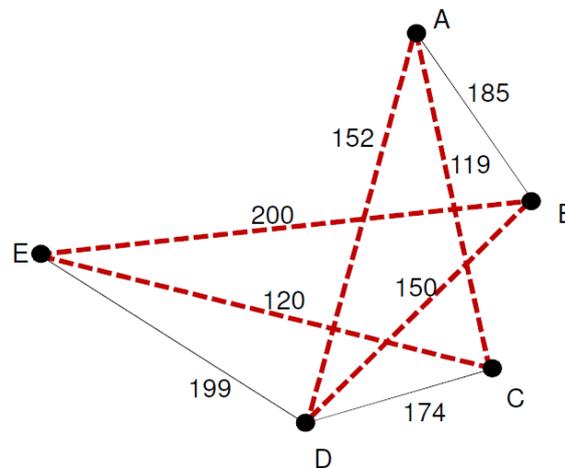


Figura 2.17: Método da aresta de menor peso - trajeto final.

Assim, o ciclo criado foi $A - C - E - B - D - A$, totalizando 741 quilômetros. Lembrando que a melhor rota tem comprimento 676 quilômetros, encontra-se uma rota com 65 quilômetros a mais que a ideal, um erro relativo de $\frac{65}{676} \approx 9,62\%$.

2.2.4 Método Repetitivo do Vizinho Mais Próximo

O *algoritmo heurístico repetitivo do vizinho mais próximo* é uma variação do *algoritmo heurístico do vizinho mais próximo*. Para utilizá-lo, basta aplicar integralmente, para cada vértice, o processo de construção do circuito do vizinho mais próximo partindo, a cada vez que o processo for iniciado, de um vértice diferente dos escolhidos anteriormente. Isso acontece porque o circuito hamiltoniano obtido depende do vértice de partida e, ao se alterar este vértice, pode-se obter um circuito hamiltoniano diferente que, em alguns casos, mais se aproxima do ciclo ótimo. No caso em que o vértice de partida estiver definido e este não coincidir com o do circuito hamiltoniano escolhido então tem-se que rescrever este mesmo circuito tomando como vértice inicial o vértice previamente estabelecido.

Aplicando o algoritmo heurístico repetitivo do vizinho mais próximo ao problema proposto anteriormente, os circuitos de menores custos obtidos para cada vértice inicial são os apresentados na tabela 2.2.

Ciclo Hamiltoniano	Distância (km)
A,C,E,D,B,A	773
B,C,A,E,D,B	722
C,A,E,D,B,C	722
D,B,C,A,E,D	722
E,C,A,D,B,E	741

Tabela 2.2: Resultado obtidos pelo algoritmo repetitivo do vizinho mais próximo.

Utilizando a heurística do vizinho mais próximo o melhor ciclo encontrado possui 773 quilômetros. Utilizando a heurística repetitiva do vizinho mais próximo foi possível encontrar outros três ciclos com distâncias de 722 km, e mais um ciclo com distância 741 km. Como nos melhores resultados o vértice inicial não é A, como foi predeterminado no problema, é necessário reescrever o ciclo encontrado tomando A como vértice inicial. Assim, o melhor ciclo encontrado é $A - E - D - B - C - A$.

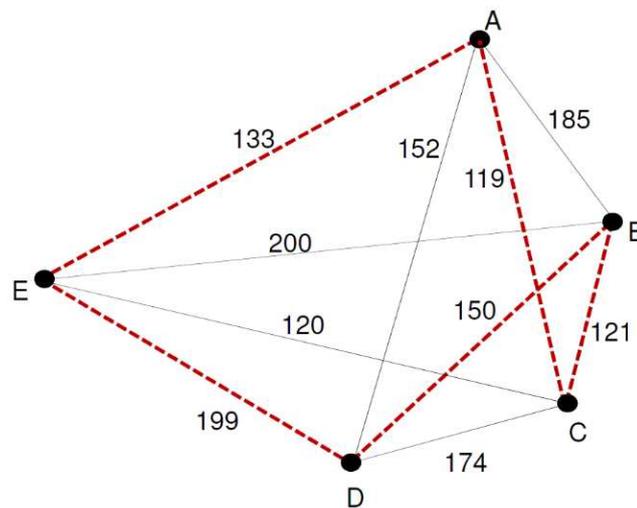


Figura 2.18: Algoritmo heurístico repetitivo do vizinho mais próximo - trajeto final.

CAPÍTULO 3

HEURÍSTICAS DE MELHORAMENTO

Devido à dificuldade de se encontrar uma solução ótima para um problema que envolva o ciclo hamiltoniano, heurísticas e meta-heurísticas (estratégias genéricas, esqueletos de algoritmos para a construção de heurísticas) de melhoramento, foram desenvolvidas [6]. Enquanto os procedimentos de construção seguem alguma regra que determina o vértice e a posição de sua inserção na rota em formação, os procedimentos de melhoramento partem de uma rota inicial conhecida e tentam reduzir o comprimento total do percurso explorando o conceito de ótimo local. Dentre muitas, se destacam *K-optimal (k-opt)*, *Algoritmos Genéticos (AG)*, *Colônia de Formigas (CF)*, *Simulated Annealing (SA)*, *Busca Tabu (BT)* e *Greedy Randomized Adaptive Search Procedure (GRASP)*. Neste capítulo serão abordadas as heurísticas 2-opt e 3-opt, Algoritmos Genéticos e Colônia de Formigas; os dois últimos fazem parte dos *algoritmos bio-inspirados*, que são algoritmos onde a natureza é utilizada como inspiração para o desenvolvimento de técnicas para a resolução de problemas complexos.

3.1 Heurística de melhoramento k-opt

Procedimentos onde, geralmente, terminam em um ótimo local, os métodos de melhoramento *2-opt* e *3-opt* foram propostos, respectivamente, por Croes, em 1958 [7], e Lin, em 1965 [28]. Funcionam de maneira semelhante, diferindo apenas no número de arcos envolvidos na análise. Em 1973, Lin e Kernighan [29] propuseram o método *k-opt*, que consiste na remoção de k arcos de uma rota e sua substituição por outros k arcos, com a finalidade de diminuir a distância total percorrida. Quanto maior o valor de k , melhor a precisão do método, porém maior é o esforço computacional. Por esse motivo, na prática, os métodos 2-opt e 3-opt são mais frequentes.

A figura 3.1 representa a ideia proposta pelo procedimento k-opt, onde podem ser encontrados mínimos locais e, eventualmente, o mínimo global.

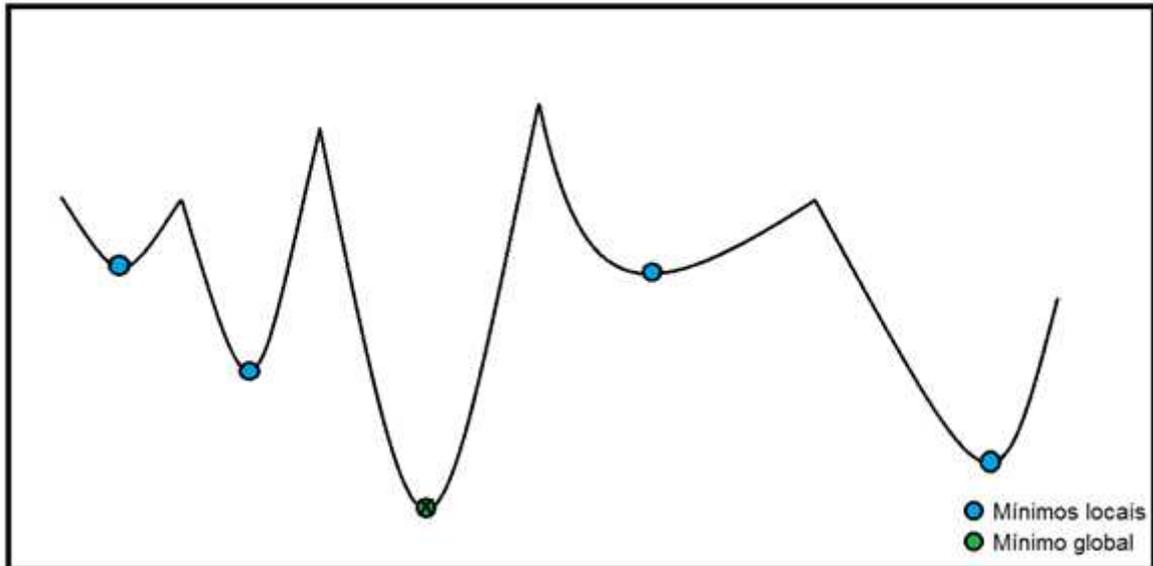


Figura 3.1: Mínimos locais e mínimo global

3.1.1 Heurística 2-opt

A heurística 2-opt é uma das técnicas mais conhecidas para melhorar uma solução para o problema do caixeiro viajante. Sua utilização é simples e segue de acordo com os passos indicados no pseudocódigo:

1. Encontre um ciclo inicial utilizando algum algoritmo de construção;
2. Tente minimizar o custo do ciclo utilizando o procedimento 2-opt: a partir de dois arcos (a, b) e (c, d) não adjacentes do ciclo, verifique a possibilidade da sua substituição pelos arcos (a, c) e (d, b) . Caso esta substituição promova uma redução de custo, efetue a troca e retorne ao passo 2. Caso contrário, mantenha os arcos originais e verifique o resultado da troca de outros dois arcos não adjacentes;
3. Execute o passo 2 até que não seja mais possível obter reduções de custo.

Algoritmo 1: Pseudocódigo 2-opt

A solução final que não pode ser melhorada mudando apenas 2 arestas é chamada de *uma solução 2-opt*.

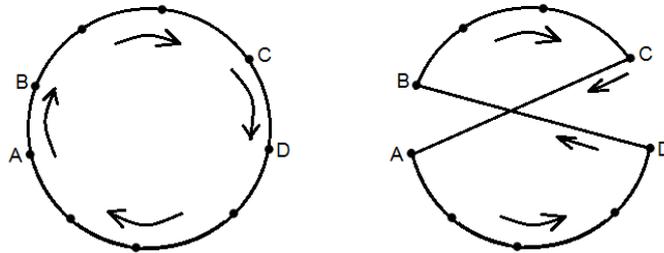


Figura 3.2: Heurística 2-opt.

Recordando o problema apresentado no capítulo 2: um vendedor reside numa cidade **A** e necessita visitar clientes em sua própria cidade e também em quatro cidades vizinhas, **B**, **C**, **D** e **E**, cujas distâncias, em quilômetros, são representadas na figura 3.3.

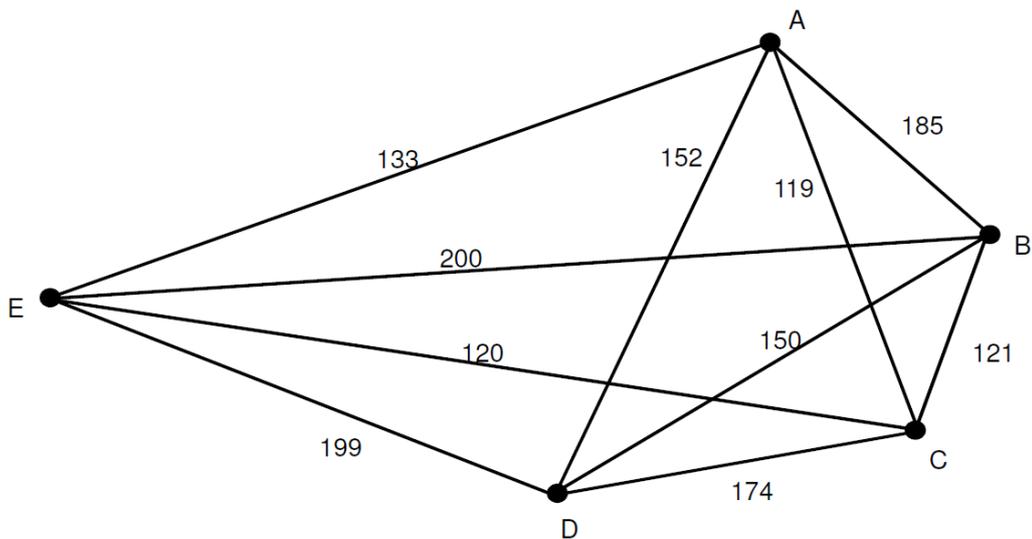


Figura 3.3: Distâncias (km).

O ciclo $A - C - E - D - B - A$ foi determinado utilizando-se o Algoritmo do Vizinho Mais Próximo. A distância encontrada foi de 773 km o que supera a rota ideal em, aproximadamente, 14,35%.

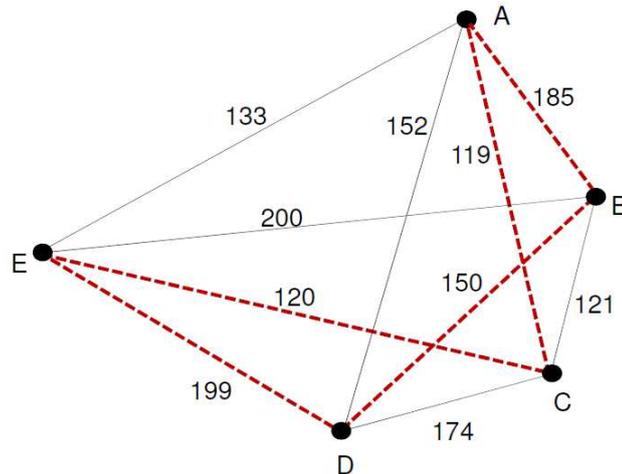


Figura 3.4: Heurística 2-opt: ciclo inicial.

Substituindo-se os arcos (A, C) e (E, D) por (A, E) e (C, D) obtém-se o ciclo $A - E - C - D - B - A$, com distância de 762 km, que representa um resultado melhor que o original. Assim, a substituição deve ser mantida e o procedimento 2-opt deve ser reiniciado.

Substituindo-se os arcos (A, E) e (C, D) por (A, C) e (E, D) obtém-se o ciclo $A - C - E - D - B - A$, com distância de 773 km, que representa um resultado pior que o anterior. Assim, a substituição deve ser desfeita e deve-se procurar outro par de arestas não adjacentes para se efetuar um novo teste.

Substituindo-se os arcos (A, E) e (D, B) por (A, D) e (E, B) obtém-se o ciclo $A - D - C - E - B - A$, com distância de 831 km, que representa um resultado pior que o anterior. Assim, a substituição deve ser desfeita e deve-se procurar outro par de arestas não adjacentes para se efetuar um novo teste.

Substituindo-se os arcos (E, C) e (D, B) por (E, D) e (C, B) obtém-se o ciclo $A - E - D - C - B - A$, com distância de 812 km, que representa um resultado pior que o anterior. Assim, a substituição deve ser desfeita e deve-se procurar outro par de arestas não adjacentes para se efetuar um novo teste.

Substituindo-se os arcos (E, C) e (B, A) por (E, B) e (C, A) obtém-se o ciclo $A - E - B - D - C - A$, com distância de 776 km, que representa um resultado pior que o anterior. Assim, a substituição deve ser desfeita e deve-se procurar outro par de arestas não adjacentes para se efetuar um novo teste.

Substituindo-se os arcos (C, D) e (B, A) por (C, B) e (D, A) obtém-se o ciclo $A - E - C - B - D - A$, com distância de 676 km, que representa um resultado melhor

que o anterior. Assim, a substituição deve ser mantida e o procedimento 2-opt deve ser reiniciado.

A tabela 3.1 fornece os resultados das demais trocas:

Trocar	Por	Circuito	Distância	Resultado
(A, E) e (C, B)	(A, C) e (E, B)	A,C,E,B,D,A	741 km	pior
(A, E) e (B, D)	(A, B) e (E, D)	A,B,C,E,D,A	777 km	pior
(E, C) e (B, D)	(E, B) e (C, D)	A,E,B,C,D,A	780 km	pior
(E, C) e (D, A)	(E, D) e (C, A)	A,E,D,B,C,A	722 km	pior
(C, B) e (D, A)	(C, D) e (B, A)	A,E,C,D,B,A	762 km	pior
(C, B) e (A, E)	(C, A) e (B, E)	A,C,E,B,D,A	741 km	pior
(B, D) e (A, E)	(B, A) e (D, E)	A,B,C,E,D,A	777 km	pior
(B, D) e (E, C)	(B, E) e (D, C)	A,E,B,C,D,A	780 km	pior
(D, A) e (E, C)	(D, E) e (A, C)	A,E,D,B,C,A	722 km	pior
(D, A) e (C, B)	(D, C) e (A, B)	A,E,C,D,B,A	762 km	pior

Tabela 3.1: Resultados 2-opt.

Constata-se que o ciclo $A - E - C - B - D - A$ é *uma solução 2-opt* para o problema proposto. Tal solução representa um ótimo local que, neste caso, é também um ótimo global. No entanto, tal resultado não pode ser garantido; aliás, não há qualquer garantia de que uma solução 2-opt se aproxime do ótimo global. Somente se pode concluir que a solução encontrada é ótima para a vizinhança considerada no momento da escolha do ciclo inicial. Outro item a ser considerado é até quando deve ser aplicado o procedimento. Caso sejam muitas as trocas, é possível determinar algum critério de parada, como, por exemplo, número de iterações, tempo de execução ou até que se atinja um resultado pré-determinado.

3.1.2 Heurística 3-opt

De maneira análoga à heurística 2-opt pode-se criar a heurística 3-opt. Nesse caso são considerados três arcos ao invés de dois, o que resulta em sete possíveis combinações, conforme pode ser visto na figura 3.5. Destas, apenas quatro combinações (4, 5, 6 e 7) representam trocas entre três arcos; as demais (1, 2 e 3) correspondem a trocas do tipo 2-opt.

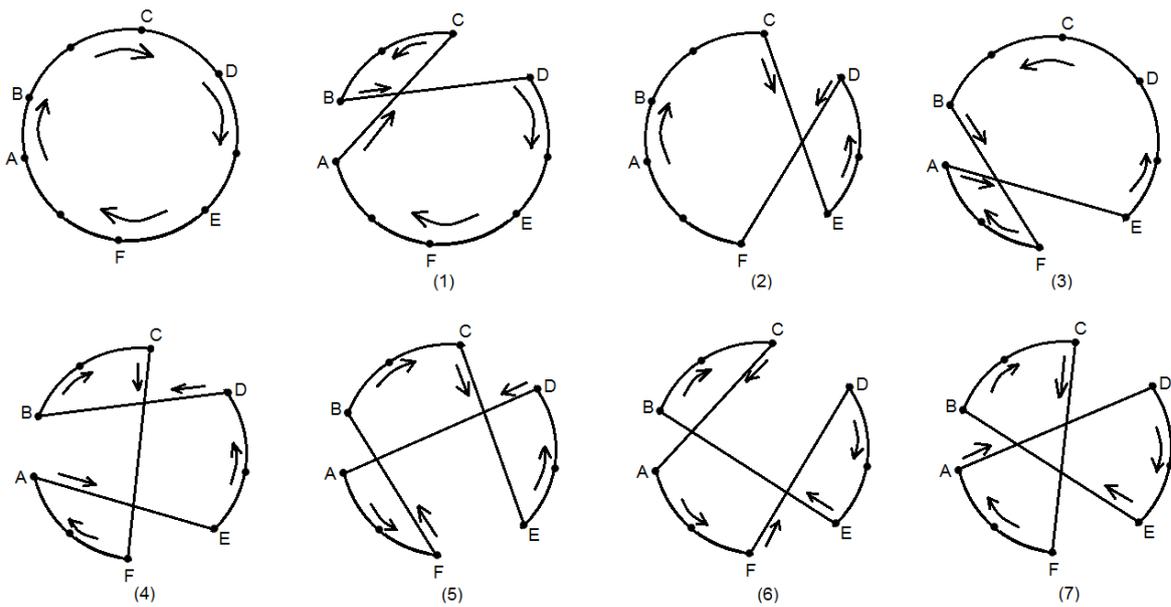


Figura 3.5: Heurística 3-opt

3.2 Algoritmos Genéticos

No início da década de 1970 John Holland, utilizando os postulados de seleção natural descritos por Charles Darwin, considerou que poderia criar algoritmos computacionais capazes de manipular cadeias de informações de forma a construir organismos complexos, melhores adaptados. Assim, apenas organismos melhores adaptados sobreviveriam em detrimento de outros, menos adaptados. Assim como na natureza, o algoritmo não tinha a informação de qual problema estava resolvendo, mas conseguia, através de uma função de avaliação, fazer com que somente os organismos com maiores condições de chegar ao resultado conseguissem sobreviver e se reproduzir, passando para os descendentes a sua carga genética e aumentando assim a chance de continuar, enquanto os menos adaptados eram eliminados. Em 1975, no seu livro *Adaptation in Natural and Artificial System*, John Holland publicou o primeiro trabalho cunhando a expressão *Algoritmos* [22].

Atualmente são utilizados para resolver problemas de busca e otimização encontrados no cotidiano. Buscam a melhor solução utilizando um processo iterativo onde a busca se dá a partir de uma população inicial que, combinando seus melhores representantes, obtém-se uma nova população que passa a substituir à anterior. A

cada nova iteração é gerada uma nova população que apresenta novas soluções para o problema em questão.

A representação de soluções em um algoritmo genético é feita através de uma *string* de caracteres, chamada de cromossomo ou indivíduo. Existem vários tipos de codificação para se representar um cromossomo: codificação binária, ponto flutuante, inteiros, etc. Para o Problema do Caixeiro Viajante, cada cromossomo pode ser uma sequência que representa a ordem em que as cidades são visitadas. Os cromossomos são classificados de acordo com uma função de avaliação, e existem várias técnicas para seleção e cruzamento desses indivíduos. Para o problema do caixeiro viajante, a função de avaliação soma a distância entre cada cidade do cromossomo. Um cromossomo é considerado melhor que outro quando a sua distância total é menor.

Basicamente, o que um algoritmo genético faz é criar uma população de possíveis respostas para o problema a ser tratado (inicialização) para depois submetê-la ao processo de evolução, constituído pelas seguintes etapas [17]:

1. Criar a população inicial com indivíduos definidos aleatoriamente;
2. **Enquanto** critérios de parada não forem atingidos **faça**:
3. Avaliar a função objetivo para os indivíduos;
4. Atribuir o fitness de cada indivíduo;
5. Selecionar os indivíduos para reprodução usando um operador de seleção;
6. Reproduzir os indivíduos usando os operadores de crossover e mutação;
7. **fim-enquanto**;
8. **Retornar** melhor indivíduo.

Algoritmo 2: Pseudocódigo Algoritmo Genético

- 1) Geração da população inicial: a população inicial pode ser gerada de forma totalmente aleatória. Entretanto, recomenda-se incluir soluções obtidas através de uma ou mais boas heurísticas ou, então, assegurar que a população esteja bem distribuída ao longo do espaço de soluções.
- 2) Avaliação: é feita uma análise para que se estabeleça quão bem as soluções encontradas respondem ao problema proposto; a avaliação é feita pela função de aptidão, usada para medir a habilidade do indivíduo de sobreviver e se

reproduzir. Quando o cromossomo representa, por exemplo, uma sequência de cidades a serem visitadas por um caixeiro viajante, a função para o cálculo da aptidão pode ser o inverso da distância total percorrida [17].

3) Seleção: existem diversos métodos para a seleção dos indivíduos para o cruzamento, porém, todos levam em consideração a aptidão de cada um dos indivíduos da população. Os principais métodos são:

3.1) Torneio (*Tournament selection*): sorteiam-se aleatoriamente k indivíduos e seleciona-se o melhor, com base na função de aptidão. Na maioria das vezes, $k = 2$.

3.2) Seleção Truncada (*Truncation selection*): consiste em definir somente uma fração de melhores indivíduos que podem ser selecionados, e todos eles possuem a mesma probabilidade de seleção.

3.3) Roleta (*Roulette wheel selection*): as aptidões de todos os indivíduos da população são calculadas e os valores obtidos corresponderão, proporcionalmente, aos setores de uma roleta. Assim, girando a roleta, a chance de os setores maiores serem selecionados será maior.

4) Cruzamento (*crossover*): o operador de cruzamento tem por finalidade combinar as informações genéticas de dois indivíduos, denominados pais, para gerar novos indivíduos, denominados filhos, formando assim a nova geração. Um método bastante comum para combinar dois indivíduos é o cruzamento com um único ponto, chamado de ponto de corte ou cruzamento. Uma posição para cruzamento, k , do vetor indivíduo é selecionada aleatoriamente e os genes de cada indivíduo, após este ponto k , são trocados, produzindo dois novos descendentes. O PCV tem uma característica peculiar para o cruzamento, que é a permutação. Um indivíduo não pode conter cidades repetidas, portanto o cruzamento deve manter o controle de quais cidades já foram colocadas no caminho do indivíduo para que não haja repetição. Para tanto, alguns operadores de cruzamento foram desenvolvidos especificamente para o PCV [33]. Veja, a seguir, alguns exemplos desses operadores.

4.1) PMX (Partially Mapped Crossover): o operador PMX busca preservar a ordem e a posição de alguns genes do cromossomo. Trata-se de um procedimento de reparo para evitar a ocorrência de elementos repetidos e funciona da seguinte maneira:

- 4.1.1) Escolhem-se aleatoriamente dois pontos de corte aplicados a ambos os pais;
- 4.1.2) Geram-se dois pré-filhos, trocando-se os segmentos centrais entre os pais;
- 4.1.3) Alguns elementos podem estar duplicados e outros faltando. Através dos segmentos centrais estabelece-se uma relação de mapeamento, corrigindo-se os pré-filhos e obtendo-se os filhos.

Por exemplo, suponha dois pais (P_1 e P_2) os quais representam possíveis soluções para o PCV para um trajeto de nove cidades:

$$P_1 = (A B C D E F G H I)$$

$$P_2 = (D E B A H G F I C)$$

Selecionando-se arbitrariamente dois pontos de corte:

$$P_1 = (A B C | D E F G | H I)$$

$$P_2 = (D E B | A H G F | I C)$$

Os segmentos entre os pontos de corte são trocados:

$$F_1 = (X X X | A H G F | X X)$$

$$F_2 = (X X X | D E F G | X X)$$

A troca representa o mapeamento $A \leftrightarrow D$, $H \leftrightarrow E$, $G \leftrightarrow F$ e $F \leftrightarrow G$. O passo seguinte corresponde ao preenchimento das cidades dos pais originais aonde não ocorra conflito:

$$F_1 = (X B C | A H G F | X I)$$

$$F_2 = (X X B | D E F G | I C)$$

Finalmente, o primeiro gene "X" do filho F_1 , o qual deveria ser A, será substituído por D, conforme o mapeamento $A \leftrightarrow D$. Da mesma maneira o segundo gene "X" no filho F_1 é substituído por E. Esse mesmo processo é repetido para o filho F_2 , gerando os seguintes filhos:

$$F_1 = (D B C | A H G F | E I)$$

$$F_2 = (A H B | D E F G | I C)$$

- 4.2) OX (Order Crossover): O operador OX É uma variação do operador PMX com um procedimento de reparo diferente:

- 4.2.1) Escolhem-se aleatoriamente dois pontos de corte que irão determinar uma sub-rota;

- 4.2.2) Um pré-filho é criado copiando-se a sub-rote na posição correspondente do Pai 1;
- 4.2.3) Os elementos da sub-rote são excluídos do Pai 2;
- 4.2.4) A partir do segundo ponto de corte, as posições vazias são preenchidas com os elementos não excluídos do Pai 2;
- 4.2.5) Quando o fim do cromossomo for atingido, continua-se a partir da primeira posição deste;

Utilizando o mesmo exemplo do operador PMX tem-se:

$$P_1 = (A B C | D E F G | H I)$$

$$P_2 = (D E B | A H G F | I C)$$

Os segmentos entre os dois pontos de corte são copiados para os filhos:

$$F_1 = (X X X | D E F G | X X)$$

$$F_2 = (X X X | A H G F | X X)$$

A partir do segundo ponto de corte de um pai, as cidades existentes do outro pai são copiadas na mesma ordem, omitindo símbolos já existentes no cromossomo. Chegando ao final da *string*, continua-se a partir da primeira posição. A sequência de cidades no segundo pai, começando do segundo ponto de corte, é (I C D E B A H G F). Após remover as cidades D, E, F e G, as quais já se encontram no primeiro filho, sobram, na ordem, as cidades I, C, B, A e H. Essa sequência é colocada no primeiro filho, começando no segundo ponto de corte, ficando:

$$F_1 = (B A H | D E F G | I C)$$

Fazendo a mesma operação para o segundo filho obtém-se:

$$F_2 = (C D E | A H G F | I B)$$

- 4.3) Operador CX (*Cycle Crossover*): o operador CX toma alguns pontos de um dos pais e seleciona os pontos remanescentes do outro pai. Os pontos do primeiro pai não são selecionados aleatoriamente e, sim, aqueles que ocupam o mesmo conjunto de posições nos dois cromossomos pais. Mantêm-se alguns genes do pai 1, gerando o filho 1. As demais posições são preenchidas com os demais genes faltantes na sequência em que eles aparecem no pai 2. O mesmo procedimento é utilizado para gerar o filho 2.

Utilizando o mesmo exemplo dos operadores supracitados, tem-se:

$$P_1 = (A B C D E F G H I)$$

$$P_2 = (D E B A H G F I C)$$

A primeira cidade do filho F_1 é retirada do pai P_1 . Dessa forma, tem-se:

$$F_1 = (A X X X X X X X X)$$

A segunda cidade a ser considerada é a primeira cidade do segundo pai, ou seja, a cidade D. A ordem que essa cidade irá ocupar no filho F_1 , é a posição na qual ela se encontra no pai P_1 . Dessa maneira, tem-se:

$$F_1 = (A X X D X X X X X)$$

A terceira cidade a ser considerada seria a cidade A, mas isso fecha um ciclo e, portanto, as demais posições devem ser preenchidas com os genes do pai P_2 . Assim, tem-se como resultado:

$$F_1 = (A E B D H G F I C)$$

Da mesma forma, o filho F_2 resulta em:

$$F_2 = (D B C A E F G H I)$$

5) Mutação: o operador de mutação tem por finalidade manter a variedade genética das populações geradas. Busca-se recuperar boas características, eventualmente, perdidas nos processos de seleção e cruzamento, e também evita-se ficar preso em determinadas regiões do espaço de busca ou em pontos de ótimo locais. O método de mutação depende do tipo de representação adotada. Para cromossomos que têm sua representação baseada na ordem, como no PCV, existem operadores de mutação específicos, semelhante ao que ocorre no cruzamento.

5.1) Mutação por inversão: dois pontos de corte são selecionados aleatoriamente e a sequência entre estas duas posições do cromossomo é invertida. Por exemplo, escolhendo-se arbitrariamente dois pontos de corte na sequência (A B C D E F G H I), tem-se:

$$(A B C | D E F G | H I)$$

Invertendo-se a posição dos genes, resulta em:

$$(A B C | G F E D | H I)$$

5.2) Mutação por substituição: consiste em selecionar uma sequência ao acaso e inseri-la em uma posição aleatória. Por exemplo, escolhendo-

se na sequência (A B C D E F G H I) os genes C, D, E e F e inserindo-os em outra posição, pode-se obter a sequência:

(A B G H C D E F I)

Uma variação para esse operador de mutação consiste na inversão da sequência quando inserida na posição aleatoriamente escolhida.

- 5.3) **Mutação por inserção:** é um caso particular da mutação por substituição em que a sequência selecionada para inserção em uma posição aleatória contém apenas um gene. Selecionam-se aleatoriamente duas posições no mesmo cromossomo e os respectivos elementos são trocados. Por exemplo, na sequência (A B C D E F G H I) os genes C e F poderiam ser trocados, resultando na sequência (A B F D E C G H I).
- 6) **Atualização:** os indivíduos criados nesta geração são inseridos na população; uma maneira mais simples de fazer isto é, simplesmente, substituir toda a população antiga pela nova, independentemente da sua aptidão. Entretanto, é recomendável preservar a informação genética dos melhores indivíduos dentre os ascendentes, repassando-a para a próxima geração, operação esta chamada de *elitismo*. Isto pode ser feito substituindo os descendentes menos aptos pelos ascendentes mais aptos.
- 7) **Finalização:** verificam-se se as condições de encerramento da evolução foram atingidas, retornando para a etapa de avaliação em caso negativo e encerrando a execução em caso positivo.

3.3 Colônia de Formigas

Inteligência de Enxames é o termo utilizado para sistemas onde o comportamento coletivo de agentes interagindo localmente com o ambiente provoca o surgimento de padrões globais coerentes e funcionais. Ela possibilita a base com a qual é possível resolver problemas coletivamente sem a necessidade de um controle central ou modelo global [27].

O *Algoritmo Colônia de Formigas* trata-se de um algoritmo inspirado no comportamento de colônias de formigas reais [11]. As formigas descobrem os caminhos mais curtos entre o formigueiro e as fontes de alimento sem o auxílio de

quaisquer pistas. Isso ocorre devido a um processo de cooperação indireta que modifica o meio ambiente, guiando as formigas pelo menor caminho [12]. Após ter encontrado uma fonte de alimento, ao retornar para a colônia, a formiga deposita no trajeto uma substância volátil denominada *feromônio*, que é uma estrutura química de comunicação e sinalização. Cada colônia tem um cheiro, o que distingue as formigas de um ninho das formigas de outro ninho. Os feromônios secretados pelas formigas informam qual o tipo de trabalho que deve ser feito, informa se há perigo por perto, se há necessidade de recrutamento de operárias para carregarem alimento rapidamente para a colônia, entre outros comportamentos específicos. Esse sistema complexo de comunicação torna as formigas aptas a colonizarem diversos ambientes.

Caso existirem diversas trilhas de feromônio conduzindo a uma mesma fonte de alimento, a seleção é probabilística baseada na concentração de feromônio em cada uma delas [41]. As formigas que percorrem o menor caminho da colônia até a fonte de alimento retornam ao formigueiro primeiro, fazendo com que nele haja uma concentração maior de feromônio, atraindo mais formigas por esse caminho.

Pode-se ilustrar o procedimento de forma simples: considere as duas primeiras formigas que partem a procura de alimento por dois caminhos distintos. Suponha que as formigas deixem a colônia ao mesmo tempo, com a mesma velocidade. A formiga que segue pelo caminho mais curto chega ao alimento após um intervalo de tempo t_1 , enquanto a formiga que segue pelo caminho mais longo leva $t_2 = t_1 + \Delta t$ para chegar ao mesmo lugar. Desconsiderando-se o tempo gasto para coletar o alimento, que deve ser igual para ambas as formigas, a primeira retorna à colônia após um intervalo de tempo de $2t_1$, enquanto a segunda leva $2t_2 = 2(t_1 + \Delta t)$. Isso significa que, durante o intervalo de tempo $[2t_1, 2t_2]$, de duração igual a $2\Delta t$, a situação encontrada é uma densidade de feromônios duas vezes maior na seção inicial do caminho mais curto do que na seção inicial do caminho mais longo, pois a formiga que optou pelo caminho mais curto já percorreu o percurso inteiro por duas vezes, isto é, na ida e na volta. A segunda formiga ainda estará no seu caminho de volta, fazendo com que a seção inicial de seu caminho só tenha sido percorrida uma única vez, durante a viagem de ida ao alimento. Então, para uma formiga que deixe a colônia em busca de alimento durante o intervalo de tempo $2\Delta t$, o caminho mais curto parecerá duas vezes mais atrativo, devido à sua densidade superior de feromônios. Esse processo deixa o caminho mais curto cada vez mais atrativo, conforme o tempo

passa. Assim, de maneira cooperativa, a colônia tem a capacidade de selecionar o menor caminho para uma determinada fonte de alimento.

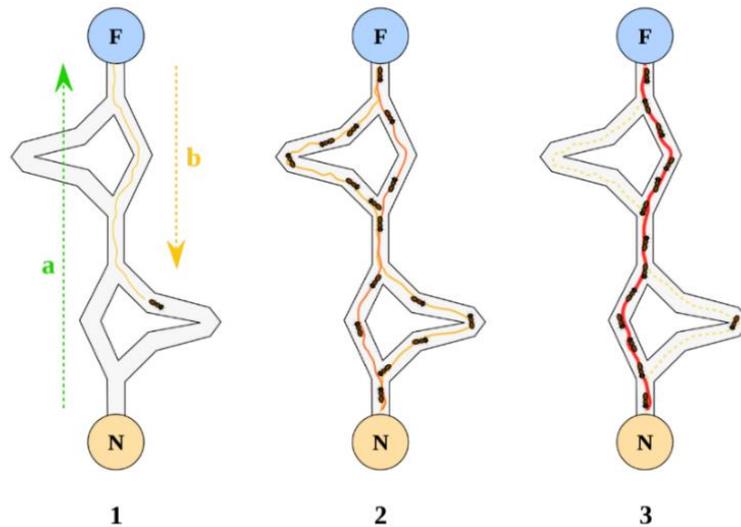


Figura 3.6: Formigas - Esquema baseado em Goss et al (1989) [20].

No esquema representado na figura 3.6, tem-se que:

- 1) Formiga inicia a exploração (a), ao encontrar uma fonte de alimento (F), retorna ao ninho (N) depositando feromônio (b).
- 2) Demais formigas são recrutadas para a exploração da área.
- 3) Após certo tempo, a maioria das formigas escolhem a menor rota. Eventualmente, algumas formigas são liberadas para explorar/patrolhar o território.

3.3.1 Ant System

O Ant System (AS) foi o primeiro algoritmo desenvolvido inspirado em colônias de formigas [11]. Nesse algoritmo, a probabilidade da formiga k , que está na cidade i , escolher como destino a cidade j é dada por:

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{il}(t)]^\alpha [\eta_{il}]^\beta}, & \text{se } j \in N_i^k \\ 0, & \text{caso contrário} \end{cases}$$

Onde:

- τ_{ij} é a quantidade de feromônio do caminho (i, j);
- $\eta_{ij} = \frac{1}{d_{ij}}$ é chamado de “visibilidade” da cidade j a partir da cidade i; representa a atratividade da formiga visitar a cidade j depois de visitar a cidade i. É dado pelo inverso da distância d_{ij} da cidade i à cidade j.
- α e β são parâmetros que permitem determinar, respectivamente, a influência do feromônio e da informação heurística (visibilidade).
- N_i^k é o conjunto das cidades ainda não visitadas pela formiga k.

Em relação ao feromônio associado ao caminho (i, j) ocorrem dois eventos: a *evaporação* e o *depósito*. A evaporação evita o acúmulo infinito de feromônio no caminho (i, j) e, conseqüentemente, permite esquecer decisões ruins tomadas no passado e o depósito, que é feito pelas formigas que passam sobre (i, j), guia as outras formigas na escolha das suas soluções definindo o quão bom é seguir por esse caminho.

Após todas as formigas construírem suas rotas o feromônio é atualizado pela fórmula:

$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \sum_{k=1}^m \Delta\tau_{ij}^k(t)$$

Onde:

- $\rho \in [0,1]$ é a taxa de evaporação do feromônio;
- m é o número de formigas;
- $\Delta\tau_{ij}^k$ é a quantidade de feromônio que a formiga k deposita no caminho (i, j). É dado por:

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{L_k}, & \text{se (i, j) pertence à rota} \\ 0, & \text{caso contrário} \end{cases}$$

onde Q é uma constante que influencia na velocidade de convergência do algoritmo e L_k mostra a distância total caminhada pela k-ésima formiga neste ciclo.

1. **para** t indo de 1 até um número máximo de iterações **faça**
2. **para** k indo de 1 até m **faça**
3. **enquanto** a formiga k não encontrar uma rota completa selecione a próxima cidade a ser visitada segundo a regra $p_{ij}^k(t)$;
4. **fim enquanto**
5. calcule a distância L_k da rota descoberta pela formiga k ;
6. **se** $L_k < L^*$ **então** $S^* \leftarrow S_k$;
7. **fim para**
8. atualize o feromônio segundo o processo τ_{ij} ;
9. **fim para**
10. **retornar** S^* ;

Algoritmo 3: Pseudocódigo Ant System

3.3.2 Ant Colony System

O Ant Colony System (ACS), proposto por Dorigo e Gambardella [10], consiste num melhoramento do Ant System. Seu desempenho é superior ao AS quando aplicado a problemas de tamanhos bem maiores. Assim como o AS, a aplicação original proposta para o ACS é o problema do caixeiro viajante.

Uma diferença significativa entre o AS e o ACS é a atualização dos feromônios. No caso do AS, ela é feita de forma global e unificada, após o término de cada iteração, sendo realizada simultaneamente a evaporação de feromônios em todos os arcos e a deposição de novos feromônios por todos os caminhos percorridos, proporcionalmente à qualidade dos caminhos encontrados. No ACS, a atualização dos feromônios ocorre de forma local após a ação de cada formiga, representando a evaporação dos feromônios pelo seu caminho, quanto de forma global após o fim de cada iteração, recompensando apenas o melhor caminho encontrado pelas formigas com certa quantidade de feromônios proporcional à qualidade da solução encontrada.

No entanto, a principal diferença entre tais algoritmos, está na regra de decisão das formigas. No AS é probabilística, de forma que os caminhos com maiores concentrações de feromônios são os mais prováveis de serem escolhidos pelas

formigas. No ACS, a regra é um pouco mais complexa, sendo dita pseudoaleatória, já que trabalha simultaneamente com a regra probabilística (desbravadora) do AS e com uma regra determinística (exploratória). Tem-se um parâmetro extra chamado q_0 , variando de 0 a 1, que define quanto poder de exploração de novos caminhos deve ser dado às formigas.

Sorteia-se um número aleatório q . Se este número for maior que q_0 , a regra de decisão da formiga é equivalente àquela do modelo AS, com a diferença que o expoente α da variável τ_{ij} é igual a 1. Esta é a parte probabilística do modelo, que permite às formigas experimentar novos caminhos, viabilizando a descoberta de rotas de comprimentos menores. Por outro lado, se o número aleatório sorteado q for menor que q_0 , a formiga tomará sua decisão baseada no conhecimento disponível sob a forma de depósitos de feromônios e distâncias, isto é, a regra de decisão da formiga será, para todo j pertencente a N_i^k , dada por

$$j = \arg \max_{l \in N_i^k} \{ [\tau_{il}(t)] [\eta_{il}]^\beta \}$$

O processo de atualização local do feromônio é feito sempre que uma formiga sai do vértice i e vai para o vértice j e não somente no final como descrito no algoritmo AS. Essa atualização é feita seguindo a fórmula:

$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \rho\tau_0$$

onde ρ ($0 < \rho < 1$) é a taxa de decaimento local do feromônio e τ_0 representa a quantidade inicial de feromônio depositado.

Já na atualização global, a ideia é recompensar os caminhos que pertencem às rotas mais curtas. Assim que as formigas terminam seus trajetos, somente a melhor formiga deposita feromônio por onde passou. Essa atualização é dada pela equação:

$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \rho\Delta\tau_{ij}(t)$$

onde

$$\Delta\tau_{ij}(t) = \begin{cases} \frac{1}{L^k}, & \text{se } (i, j) \text{ pertence ao melhor caminho} \\ 0, & \text{caso contrário} \end{cases}$$

em que L^k é a distância total percorrida na rota construída pela melhor formiga da iteração. Quanto menor a rota, maior a quantidade de feromônio depositada. Esse

procedimento se repete até que um número máximo de iterações tenha sido alcançado ou caso não se verifique mais melhorias nas soluções encontradas.

CAPÍTULO 4

GRAFOS NO ENSINO BÁSICO

Somente o desenvolvimento tecnológico e o amplo acesso à informação já seriam suficientes para se planejar uma alteração do sistema educacional, que carece de atualizações. Cada vez mais é necessário conectar o ensino oferecido nas salas de aula às demandas do mundo que se encontra fora das paredes da escola. É preciso que o docente desenvolva metodologias de ensino que motivem o aluno a “querer aprender”. Isso se dá com todas as disciplinas oferecidas do Ensino Básico, mas a matemática, em particular, necessita urgentemente de uma revisão de métodos e, talvez, até mesmo de conteúdo.

Dados do *Programme for International Student Assessment (PISA)* 2012 [24] colocam o Brasil obtendo 58ª posição (entre sessenta e cinco países) com 391 pontos, estando 67,1% dos participantes abaixo do nível 2, isto é, com baixo aproveitamento, e apenas 0,8% dos participantes no Nível 5 ou 6, que representam os melhores desempenhos. Tais números mostram a urgência de uma revisão na forma como é ensinada a matemática no Brasil.

A Matemática Discreta está presente em muitos cursos de nível superior, mas no Ensino Básico ela não é formalmente apresentada. É trabalhada, sem ser citada, na Teoria dos Conjuntos e em “problemas de combinatória”, mas sempre repleta de leis e algoritmos que, em várias situações, surgem sem qualquer motivação, o que muitas vezes desestimula o aluno que está se familiarizando com a matemática. Esse perfil da Educação Básica propicia uma introdução à teoria dos grafos como um caminho para trazer a matemática para um mundo mais próximo do dia a dia dos alunos. Permite, de forma simples, a construção das ideias básicas que permeiam os processos algorítmicos, além de ser uma área riquíssima em aplicações, as quais nos remetem a problemas realmente contextualizados, interessantes e atuais, além de possibilitar o desenvolvimento de atividades multidisciplinares.

Analisando os PCN's [40], observa-se que não contemplam o desenvolvimento de tópicos da matemática discreta e, conseqüentemente, a teoria de

grafos nela contida. No entanto, nas *Orientações Curriculares para o Ensino Médio* encontram-se alguns indicativos da pertinência de se trabalhar com grafos no ensino médio. Neste documento são apresentados alguns tópicos que podem ser trabalhados em feiras de ciência, laboratórios de matemática ou ainda para compor a parte diversificada do currículo [39].

Pode-se despertar o interesse de grande parte dos estudantes utilizando um desafio de simples exposição e fácil compreensão. Isso estimula uma saudável competição para ver quem encontra mais rapidamente uma solução. É possível, inclusive, criar grupos para que seus integrantes possam discutir os possíveis passos a serem seguidos no intuito de se resolver o problema.

Neste capítulo serão apresentados alguns problemas e sugestões de aplicação e desenvolvimento de aulas ou projetos envolvendo a teoria dos grafos voltada aos alunos do Ensino Médio.

4.1 O Problema da Casinha e as Pontes de Königsberg

Apresentando a figura 4.1, questione se é possível desenhá-la sem tirar o lápis do papel e sem passar pela mesma linha duas vezes.

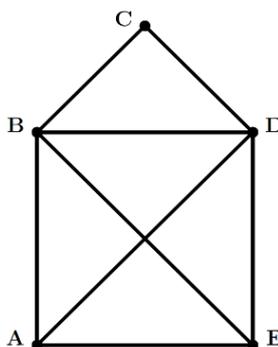


Figura 4.1: Problema da casinha

É bem possível que alguns alunos encontrem uma solução. Após a apresentação e análise das soluções, pode-se enfatizar que todas as soluções encontradas começam no vértice *A* e terminam no vértice *E* ou Começam em *E* e terminam em *A*. Questione se seria possível iniciar e terminar a figura num mesmo vértice. Ou se seria possível, partindo de *B*, encontrar uma solução. Obviamente,

nessas condições, nenhuma solução será encontrada. Nesse momento, pode-se apresentar o Problema das Pontes de Königsberg.

Após um tempo de reflexão, o professor pode expor a solução apresentada por Euler ou exibir um dos inúmeros vídeos que falam sobre o tema. Por exemplo, pode apresentar o vídeo *Pontes de Königsberg*, que tem duração de 2 min 17 s e está disponível em: <https://www.youtube.com/watch?v=RdN1JwTaUos>. Veja que, além de ser uma deixa para a apresentação dos grafos eulerianos, há espaço para a biografia de Euler, com suas principais contribuições ao desenvolvimento matemático científico e também é possível criar uma ligação com os conteúdos de Geografia e História, que poderiam abordar as características e os principais fatos ocorridos na região.

Após essa introdução, é possível trabalhar com os temas e definições apresentados no primeiro capítulo desse trabalho. Pode-se, inclusive, apresentar a ideia de matrizes e suas várias aplicações a problemas presentes no cotidiano.

4.2 O Problema das Três Casas

A origem deste problema é desconhecida, mas foi primeiramente mencionada por Henry Ernest Dudeney, em 1913, na revista britânica *The Strand Magazine*. O problema consiste em fornecer água, gás e eletricidade, de W, G e E, a cada uma das três casas, A, B e C, sem que qualquer tubulação cruze com outra.

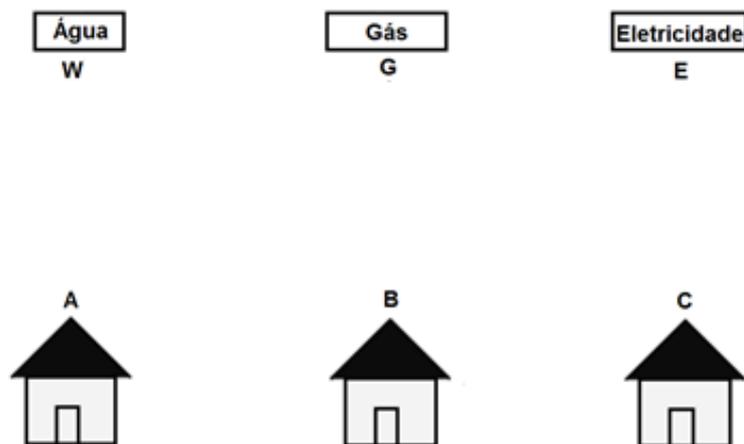


Figura 4.2: Problema água-gás-eletricidade.

Dê um tempo para que os alunos pensem a respeito. É possível que alguns já conheçam o problema, e isso o torna ainda mais atraente, pois é improvável que a maioria saiba como convencer alguém da solução, isto é, da impossibilidade de tal ligação ser feita nas condições apresentadas.

Nesse momento, os alunos já devem estar familiarizados com as definições e notações usados em teoria dos grafos. Sendo assim, é possível apresentar os teoremas e definições a seguir.

Teorema 4.2.1. (Euler, 1750) Seja G um grafo simples conexo planar e sejam n , m e f o número de vértices, número de arestas e número de faces respectivamente de uma representação plana de G . Então $n - m + f = 2$.

Demonstração: Qualquer grafo conexo G pode ser construído a partir de uma árvore geradora adicionando arestas a ela, uma a uma, até que o grafo G seja obtido. Observe que, sendo T_G uma árvore geradora de G , T_G tem n vértices, $n - 1$ arestas e 1 face e isso significa que:

$$n - m + f = n - (n - 1) + 1 = 2$$

ou seja, a relação é válida para a árvore geradora de G .

Perceba ainda que, em qualquer etapa da construção do grafo G a partir de sua árvore geradora, a adição de uma aresta não altera o valor de $n - m + f$. Cada vez que uma aresta é adicionada, tal aresta deve conectar dois vértices diferentes e “cortar” uma face existente em duas. Isso deixa n fixo, aumenta m em uma aresta e f em uma face. Assim, $n - m + f$ fica inalterado.

Caso G não seja uma árvore, considere C um ciclo em G e a uma aresta de C . Então o grafo $G - a$ é conexo, com n vértices, $m - 1$ arestas e $f - 1$ faces pois, ao se remover uma aresta de C , duas faces de G são unidas. Assim, $n - m + f$ fica inalterado, pois, $n - m + f = n - (m - 1) + (f - 1)$.

Corolário 4.2.2. Se G é um grafo conexo e planar com n , vértices, $n \geq 3$, e m arestas, tem-se que $3n - m \geq 6$.

Demonstração: Cada face de um grafo é delimitada por mínimo três arestas. Logo, $2n \geq 3f$, pois cada aresta é compartilhada por duas faces. Substituindo, na fórmula de Euler, f por $\frac{2}{3}m$, tem-se $n - m + \frac{2}{3}m \geq 2$, ou seja, $3n - m \geq 6$.

Com esse resultado é possível concluir que o grafo K_5 não é planar. Nesse grafo é $n = 5$ e $m = 10$, portanto, $3n - m = 3 \cdot 5 - 10 = 5 < 6$.

Definição 4.2.3. Um grafo G é *bipartido* se $V(G)$ pode ser particionado em conjuntos X e Y de modo que toda aresta de G tem um extremo em X e outro em Y . Como consequência desta definição, X e Y são conjuntos independentes. Um grafo bipartido G será *bipartido completo* se, para qualquer par de vértices x, y com $x \in X$ e $y \in Y$, vale que $x, y \in A(G)$. Um grafo bipartido completo é denotado por $K_{p,q}$ com p vértices em X e q vértices em Y .

Corolário 4.2.4. Se G é um grafo conexo e planar com n vértices, $n \geq 3$, m arestas e sem triângulos, isto é, se não possui K_3 como subgrafo, então $m \leq 2n - 4$.

Demonstração: Como não há circuitos de comprimento três, todos os circuitos têm quatro ou mais arestas, ou seja, cada face tem grau pelo menos 4. Assim, a soma dos graus das faces é, no mínimo, $4f$ e, portanto, $2m \geq 4f$. Da fórmula de Euler vem que $f = m - n + 2$, ou seja, $m - n + 2 \leq \frac{m}{2}$ que resulta em $2n - m \geq 4$.

Com esse resultado é possível concluir que o grafo bipartido completo $K_{3,3}$ não é planar. Nesse grafo não há circuitos com três arestas e tem-se que $n = 6$ e $m = 9$. Portanto, $2n - m = 2 \cdot 6 - 9 = 3 < 4$.

Definição 4.2.5. Um grafo H é homeomorfo a outro grafo, G , se H puder ser obtido de G pela inserção de vértices de grau dois em pontos intermediários de suas arestas.

Teorema 4.2.6. (Kuratowski, 1930) Um grafo G é planar se, e somente se, G não contém subgrafo homeomorfo ao $K_{3,3}$ ou K_5 [21].

Retornando ao problema água-gás-eletricidade, representado pela figura 4.2. O grafo $K_{3,3}$ é uma ilustração do problema, e como ele é não planar, então, não é

possível fazer as conexões desejadas no problema sem que haja cruzamento de arestas.

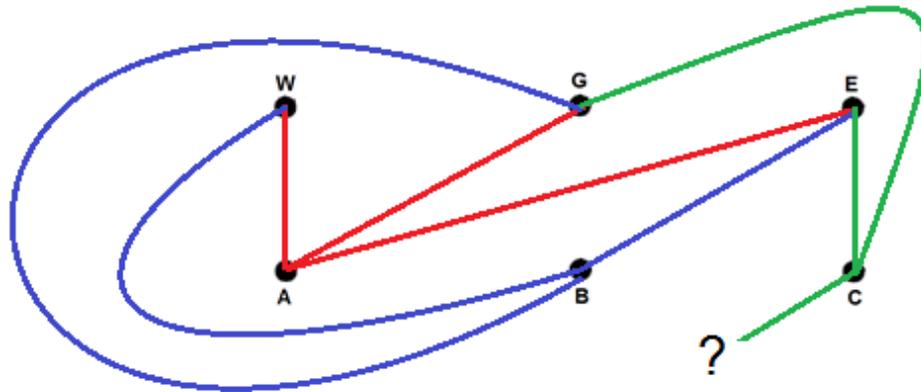


Figura 4.3: Problema água-gás-eletricidade na forma de grafo.

Observe que, além de resolver o problema do abastecimento de água-gás-eletricidade, é demonstrado o teorema da chamada Fórmula de Euler para poliedros.

Outra forma de demonstrar a fórmula de Euler para poliedros foi apresentada por Cauchy e segue, em linhas gerais, os passos apresentados a seguir.

Considere um poliedro convexo, P , com V vértices, A arestas e F faces. Inicialmente retira-se uma de suas faces. Como resultado obtém-se uma nova figura com o mesmo número de vértices, o mesmo número de arestas e uma face a menos. Nessa nova figura existem arestas que pertencem a apenas uma face, chamadas arestas livres. Considere, agora, a figura obtida ao se “esticar” de modo conveniente essa nova figura a partir de suas arestas livres, achatando-a até que ela se torne uma figura plana. Tal figura plana tem o mesmo número V de vértices, A de arestas e $F-1$ faces. Em seguida traçam-se, nos polígonos da figura plana, diagonais que não se cortam, obtendo uma decomposição de cada uma de suas faces em triângulos. Observe que, a cada diagonal traçada, a figura plana terá o mesmo número de vértices e aumenta-se o número de arestas e o número de faces em uma unidade cada um. Assim, o número de seus vértices menos o número de suas arestas mais o número de suas faces é igual a $V - (A+1) + (F-1+1)$, ou seja, $V - A + F - 1$. Portanto, o resultado permanece inalterado. Ao fim do processo, existem somente triângulos com apenas uma aresta livre, com duas arestas livres ou sem arestas livres. A seguir, retira-se da figura uma a uma as faces triangulares, seguindo os seguintes critérios:

Se existe triângulo com duas arestas livres, retira-se esse triângulo, diminuindo um vértice, duas arestas e uma face. Na figura restante, o número de vértices menos o número de arestas mais o número de faces é igual a $(V-1)-(A-2)+(F-1-1)$, ou seja, $V-A+F-1$ e, portanto, o resultado permanece inalterado e a figura continua tendo somente triângulos dos três tipos citados.

Se não existir triângulos com duas arestas livres, retira-se um triângulo com uma aresta livre, diminuindo uma aresta e uma face. Para a figura que resta, o número de vértices menos o número de arestas mais o número de faces é igual a $V-(A-1)+(F-1-1)$, ou seja, $V-A+F-1$. Assim, o resultado permanece inalterado e a figura continua tendo somente triângulos dos tipos citados anteriormente.

O procedimento deve continuar até que reste apenas um triângulo. Como, para o triângulo, o número de vértices menos o número de arestas mais o número de faces é igual a 1, e em todas as retiradas este resultado permanece inalterado e igual a $V-A+F-1$, conclui-se que $V-A+F-1=1$, ou seja, $V-A+F=2$.

4.3 Problema do Carteiro Chinês

Dado um mapa rodoviário contendo as distâncias entre cidades, qual caminho deve ser escolhido de forma a minimizar a distância percorrida desde uma cidade de origem a uma outra cidade de destino? Tal situação pode ser modelada através de um grafo onde os vértices representam as cidades, as arestas ponderadas são as estradas e os pesos das arestas são as distâncias entre as cidades. Problemas desse tipo são conhecidos como *Problema do Caminho Mínimo*.

Goodrich e Tamassia [19] relatam que, ao se tentar descobrir o caminho mais curto entre dois vértices quaisquer de um grafo, o algoritmo de Dijkstra, concebido pelo cientista da computação holandês Edsger Wybe Dijkstra, é uma das primeiras opções. Tal algoritmo encontra o menor caminho entre dois vértices, s e u , do grafo quando todos os arcos possuem comprimentos não negativos. Utiliza um procedimento iterativo determinando, na primeira iteração, o vértice mais próximo do vértice inicial s , na segunda iteração, o segundo vértice mais próximo de s e assim sucessivamente, até que o vértice final u seja alcançado.

Seja $G = (V, A)$ um grafo orientado e s um vértice de G :

1. Atribua zero à estimativa de custo mínimo do vértice s (raiz) e infinito às demais;
2. Atribua um valor qualquer aos antecessores;
3. **enquanto** houver vértice aberto:
 4. Seja k um vértice ainda aberto cuja estimativa seja a menor dentre todos os vértices abertos. Feche k ;
 5. **para** todo vértice j ainda aberto que seja sucessor de k **faça**:
 6. Some a estimativa do vértice k com o custo do arco (k, j) ;
se esta soma for melhor que a estimativa anterior para o vértice j , **então** substitua-a e anote k como precedente de j ;
7. **fim para**
8. **fim enquanto**

Algoritmo 5: Algoritmo de Dijkstra

Um problema que envolve a ideia de caminhos mínimos é o *Problema do Carteiro Chinês* (PCC), proposto originalmente pelo matemático chinês Mei-Ku Kwan, em 1962 [26], que pode ser descrito do seguinte modo: um carteiro, a partir de um posto dos correios, deve fazer a distribuição das cartas pelas casas que lhe são destinadas e regressar ao posto de maneira a percorrer a menor distância possível. Obviamente o carteiro deve percorrer pelo menos uma vez cada uma das ruas a fim de fazer a distribuição por todas as casas.

Esse problema pode ser reformulado em termos de grafos ponderados nos quais se procura um circuito de Euler. Nesse grafo ponderado, as ruas correspondem às arestas, os vértices são os cruzamentos das ruas, e o peso de cada aresta é o seu comprimento [18]. Se o grafo for euleriano basta encontrar um circuito, que configura um percurso do carteiro, onde o custo (distância percorrida pelo carteiro) é a soma dos pesos das arestas que o constituem. Na determinação desse circuito pode-se utilizar diversos algoritmos como, por exemplo, o Algoritmo de Hierholzer, que consiste em, a partir de um vértice qualquer, percorrer arestas até retornar ao vértice inicial. Porém, dessa forma pode ser obtido um ciclo que não inclua todas as arestas do grafo. Assim, enquanto houver um vértice que possui arestas ainda não exploradas deve-se começar um caminho nesse vértice e tentar voltar a ele, usando somente

arestas ainda não percorridas. Quando não houver mais arestas não percorridas, basta combinar os ciclos determinados a partir de um vértice comum.

1. Escolher um vértice $v \in V(G)$ como vértice de partida.
2. Determinar um circuito C que se inicie e termine em v .
3. **Se** C inclui todas as arestas do grafo, **então** C representa um circuito de Euler. **Senão**, escolher um vértice $v \in V(C)$, do qual parta uma aresta que não pertença a C .
4. Determinar um circuito C' que se inicie e termine em v , utilizando somente arestas que não pertençam a C .
5. Juntar os circuitos C e C' a partir de um vértice comum e denotar o novo circuito por C .
6. Voltar à linha 3.

Algoritmo 4: Algoritmo de Hierholzer.

Como exemplo, observe a figura 4.4, que descreve um circuito construído utilizando-se o algoritmo de Hierholzer.

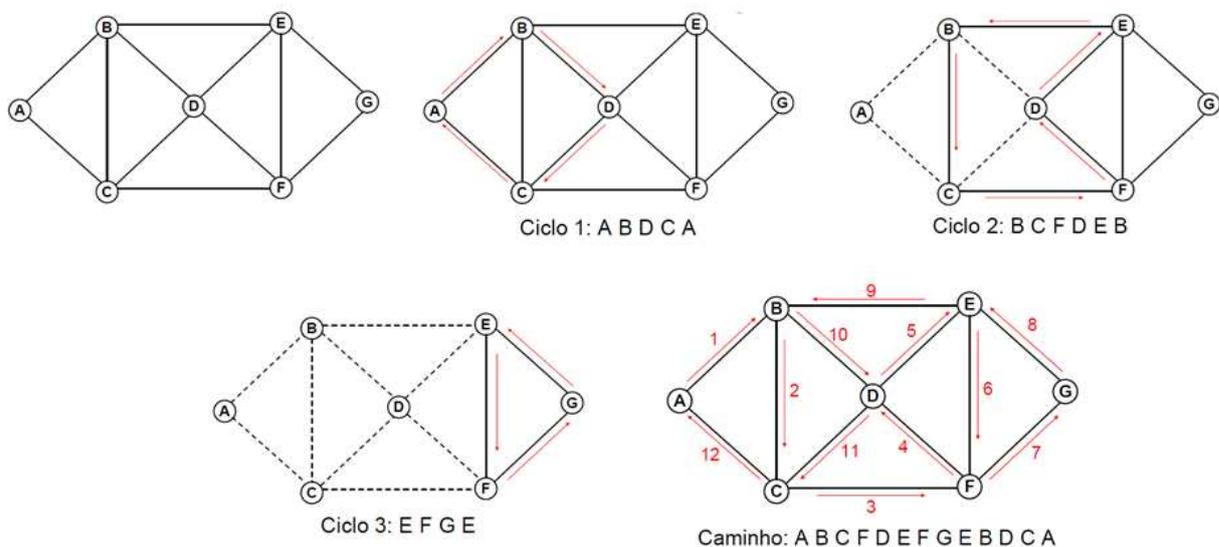


Figura 4.4: Exemplo do Algoritmo de Hierholzer.

Inicialmente, é determinado o ciclo A, B, D, C, A. Assim, as arestas (A, B), (B, D), (D, C) e (C, A) devem ser retiradas. Em seguida é determinado o ciclo B, C, F, D, E, B e, portanto, as arestas (B, C), (C, F), (F, D), (D, E) e (E, B) também são

retiradas. Por fim, determina-se o ciclo E, F, G. Combinando os ciclos a partir de um vértice comum, determina-se o caminho A, B, C, F, D, E, F, G, E, B, D, C, A.

No entanto, se o grafo não admite um circuito de Euler, é preciso adicionar arestas paralelas às já existentes de modo a transformar os vértices de grau ímpar em vértices de grau par, isto é, deve-se eulerizar o grafo. Para isso é necessário combinar dois a dois todos os seus vértices de grau ímpar. Esse problema de combinação é chamado de “Pairwise Matching”, e foi resolvido por Edmonds e Johson em 1973 [13].

O Algoritmo do Carteiro Chinês pode ser encontrado na literatura tendo a seguinte sequência [16]:

1. Ler o grafo $G = (V, A)$;
2. **Se** todos os nós do grafo original, G , possuírem grau par, **então** determinar um ciclo euleriano em G ;
3. **Senão**, organizar um grafo completo, R_T , da seguinte forma:
4. Reunir todos os vértices de grau ímpar no grafo R_T e associar a cada par de vértices v_i e v_j no grafo, uma aresta (i, j) com peso igual ao caminho mais curto que liga v_i a v_j no grafo G ;
5. Determinar o emparelhamento perfeito de peso mínimo, M^* , em R_T . Para cada aresta pertencente a M^* , associar uma nova aresta em G no caminho mínimo que ela representa, obtendo um grafo G_a ;
6. Determinar a solução do carteiro chinês que é representada por um ciclo euleriano em G_a .

Algoritmo 5: Algoritmo do Carteiro Chinês

Como exemplo de solução do PCC, considere o grafo G mostrado na figura 4.5. Deseja-se determinar um ciclo de extensão mínima com início e fim no vértice (a).

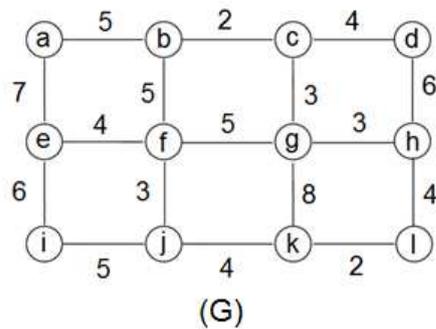


Figura 4.5: Problema do Carteiro Chinês – exemplo.

Observe que os vértices $\{b, c, e, h, j, k\}$ possuem quantidades ímpares de arestas; nesse caso, deve-se proceder a eulerização do grafo G .

A figura 4.6 mostra o grafo completo, R_T , formado a partir dos vértices com quantidades ímpares de arestas do grafo original e o grafo aumentado, G_a , que é formado a partir da replicação de arestas incidentes nos vértices de grau ímpar e que serão atravessadas mais de uma vez.

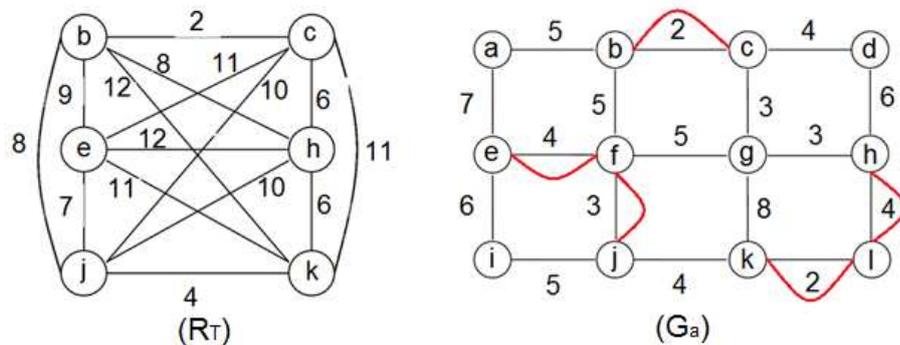


Figura 4.6: Grafo eulerizado.

A partir de do grafo G_a pode ser definido um ciclo do carteiro, dado nesse caso pela sequência de vértices $(a, b, c, b, f, g, c, d, h, l, h, g, k, l, k, j, f, j, i, e, f, e, a)$, com um custo total de 91 u.c.

É importante ressaltar que o caso analisado anteriormente é para grafos não direcionados mas vale lembrar que as principais abordagens do Problema do Carteiro Chinês são [14]:

1. **Problema do Carteiro Chinês Não Direcionado (PCCND)**, onde se deseja gerar um percurso de custo mínimo sobre um grafo $G = (V, A)$, valorado e

conexo, a partir de um vértice $v_0 \in V$, origem. Exemplo: Cidades somente com ruas de mão dupla.

2. **Problema do Carteiro Chinês Direcionado (PCCD)**, onde se deseja gerar um percurso de custo mínimo sobre um grafo direcionado (dígrafo) $G = (V, A)$, valorado e fortemente conexo (f -conexo), a partir de um vértice origem $v_0 \in V$. Exemplo: Cidades somente com ruas de mão única.

3. **Problema do Carteiro Chinês Misto (PCCM)**, onde se deseja gerar um percurso de custo mínimo sobre um grafo $G = (V, A)$, com algumas arestas direcionadas e outras não direcionadas, valorado e fortemente conexo (f -conexo), a partir de um vértice origem $v_0 \in V$. Exemplo: Cidades com ruas de mão dupla e mão única.

Para a apresentação do Problema do Carteiro Chinês pode-se partir de um grafo já construído e pedir para que seja determinada uma rota que minimize o percurso de um carteiro que deve percorrer, a partir do vértice 1, todas as ruas (representadas pelas arestas do grafo) da figura 4.7, onde o peso de cada aresta é a distância, em metros, que deve ser percorrida em cada rua.

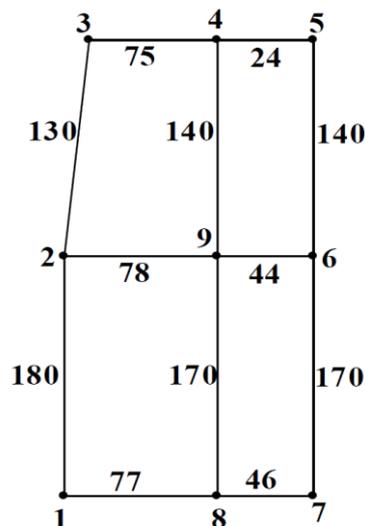


Figura 4.7: Problema do carteiro.

Após aguardar até que surjam soluções apresenta-se uma forma de resolver o problema. A solução pode ser apresentada identificando-se os pares de vértices de grau ímpar (2, 4, 6 e 8) e determinando o caminho mais curto entre esses vértices para saber quais arestas duplicar.

Os pares de vértices possíveis formar são (2, 4) e (6, 8) ou (2, 6) e (4, 8) ou (2, 8) e (4,6).

O Caminho mais curto entre os vértices 2 e 4 é dado pela sequência de vértices 2, 3, 4; entre os vértices 6 e 8 a sequência é 6, 9, 8 e a soma é 419 metros.

O Caminho mais curto entre os vértices 2 e 6 é dado pela sequência de vértices 2, 9, 6; entre os vértices 4 e 8 a sequência é 8, 9, 4 e a soma é 432 metros.

O Caminho mais curto entre os vértices 2 e 8 é dado pela sequência de vértices 2, 9, 8; entre os vértices 4 e 6 a sequência é 4, 5, 6 e a soma é 412 metros.

Os pares (2, 8) e (4, 6) fornecem a menor distância, assim o grafo aumentado fica como representado na figura 4.8.

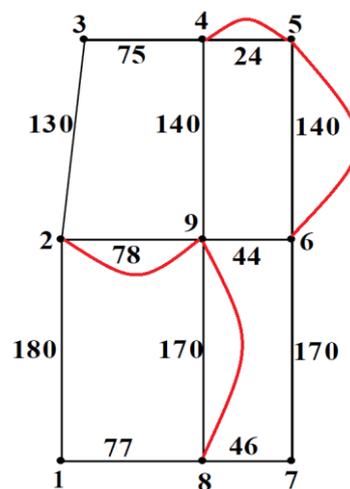


Figura 4.8: Problema do carteiro eulerizado.

Analisando-se o grafo eulerizado basta encontrar um percurso de Euler, como, por exemplo, a sequência de vértices 1, 2, 3, 4, 5, 6, 7, 8, 9, 2, 9, 4, 5, 6, 9, 8, 1, onde a distância percorrida é 1686 metros.

4.4 Coloração

O Teorema das 4-cores é muito conhecido em teoria dos grafos pelas muitas tentativas falhas de sua demonstração tanto por matemáticos amadores quanto por grandes nomes da matemática [38].

O teorema das 4-cores surgiu logo após o problema das 4-cores. No ano de 1852 o então estudante Francis Guthrie estava colorindo mapas dos condados da Inglaterra e, ao fazer tal coloração, observou que eram necessárias apenas 4 cores diferentes para colorir um mapa sem que condados fronteiros tivessem a mesma cor. Intrigado e sem conseguir sucesso na demonstração de que com 4 cores seria possível pintar qualquer mapa, foi perguntar ao irmão, Frederick Guthrie, se isso se passava com qualquer mapa. Frederick, não conseguindo sanar a dúvida de seu irmão, foi perguntar a seu professor em Cambridge, Augustus de Morgan que, muito entusiasmado, compartilhou o problema com seus estudantes e outros notáveis matemáticos. Dentre estes, estava Sir William Hamilton, que não demonstrou grande interesse pelo problema, respondendo a De Morgan, quatro dias depois, que não tinha a intenção de debruçar-se sobre a questão.

No ano de 1878 Arthur Cayley retomou o problema em um trabalho literário editado pela London Mathematical Society, a qual o próprio Cayley era presidente na época. Um ano após a retomada do problema das 4-cores o matemático Alfred Bray Kempe fez a primeira demonstração do teorema, mas em 1890 o matemático Percy John Heawood apontou alguns erros na demonstração de Kempe. No mesmo artigo, Heawood lamentava não ter sido capaz de obter nenhuma demonstração alternativa do teorema. Conseguiu, no entanto, provar o *Teorema das Cinco Cores*, isto é, demonstrar que não são necessárias mais do que cinco cores para colorir um mapa plano onde países de fronteira comum devem possuir cores diferentes. Paralelamente, em 1880 Peter Guthrie Tait fez outra demonstração que seguiu aceita até 1891 quando Petersen mostrou ser falha.

Dos anos de 1891 até 1922 o teorema das 4-cores foi ganhando contribuições que ajudaram Philip Franklin, em 1922, a mostrar que o teorema era válido para mapas com no máximo 25 regiões.

De 1922 a 1976 o teorema das 4-cores seguiu ganhando contribuições de outros matemáticos e, no ano de 1976, Kenneth Appel e Wolfgang Haken conseguiram reduzir os casos possíveis a 1482 configurações gráficas e colocaram um computador para verificá-los um a um [1] [2]. Com isso foi demonstrado o teorema das 4-cores, embora não muito aceito pela comunidade matemática por ser uma demonstração feita utilizando computador. Em 1989, Appel e Haken [3] publicaram uma segunda versão desta demonstração onde respondem a várias críticas e corrigem algumas falhas detectadas na primeira versão. Em 1994 Robertson,

Sanders, Seymour e Thomas [36] apresentaram uma prova semelhante, mas com a utilização de “apenas” 633 configurações, cuja verificação ainda foi feita por computador.

Como o tema traz muitos nomes importantes à tona, é possível trabalhar um tópico envolvendo a história da matemática nesse período de mais de 150 anos. Também fica claro que problemas de fácil exposição podem apresentar grande dificuldade de demonstração, além de levantar uma discussão a respeito do uso de tecnologia no auxílio da resolução de problemas.

Pode-se iniciar o tema apresentado alguns mapas e pedindo aos alunos que os pintem com a menor quantidade de cores possível, de tal forma que regiões fronteiriças possuam cores diferentes. É importante ressaltar que apenas um ponto comum não representa uma fronteira. Pode ser interessante apresentar alguns mapas que podem ser coloridos com menos que quatro cores. Por exemplo, pode ser solicitado aos alunos que pintem mapas com as regiões do Brasil separadamente. Pode-se iniciar com a Região Sul, seguir para Sudeste, Centro-Oeste, Norte e Nordeste. Perceba que o grau de dificuldade vai aumentando se essa ordem for seguida. Para finalizar, pode-se solicitar que o mapa do Brasil seja colorido nos mesmos moldes das regiões anteriores.

Para associar a coloração de mapas a grafos, basta, num mapa planar, marcar um vértice no interior de cada região ("capital" do país/estado), em seguida, para cada dois países/estados vizinhos, desenhar uma aresta ligando as duas capitais, cortando a fronteira comum de ambos. Dessa maneira, o problema de coloração de mapas recai sobre o problema de coloração de vértices de um grafo.

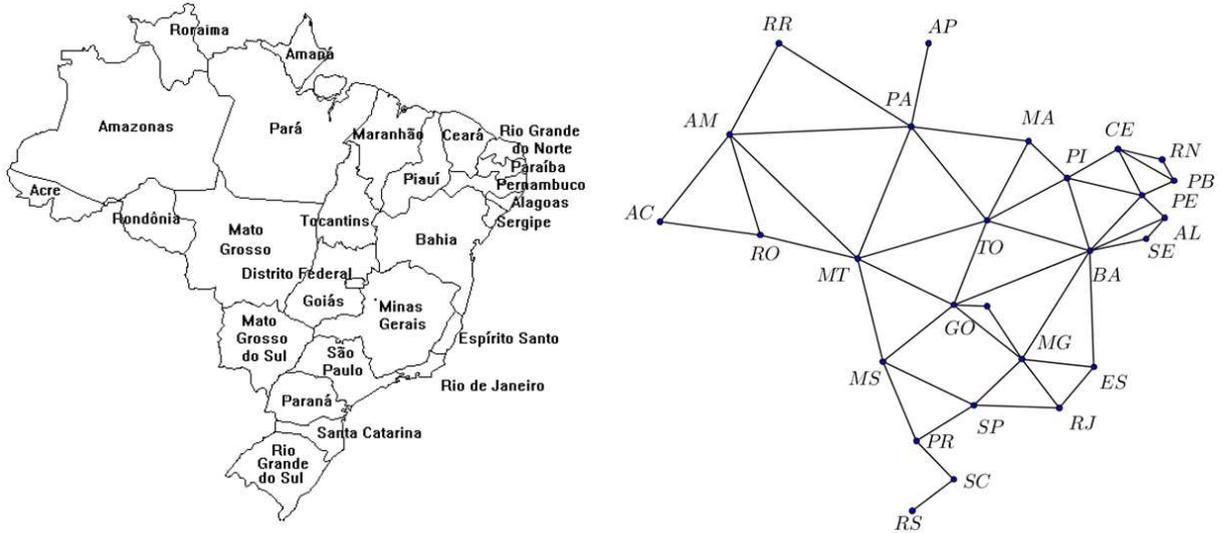


Figura 4.9: Grafo associado ao mapa do Brasil

Seja $G = (V, A)$ um grafo e $C = \{c_i, \leq i \leq n\}$ um conjunto de cores. Uma coloração de vértices de G é uma atribuição de cores de C para os vértices, de maneira que a vértices adjacentes sejam atribuídas cores diferentes. Uma k -coloração é uma coloração que consiste de k cores diferentes.

O número cromático de um grafo G , representado por $\chi(G)$, é o menor inteiro positivo k tal que exista uma k -coloração desse grafo. Nesse caso, é dito que essa coloração é mínima para G e que G é k -cromático.

Determinar o número cromático de um grafo qualquer, em geral, é difícil. O que se pode afirmar sem qualquer dificuldade é que se um grafo tem n vértices, então seu número cromático é $\chi(G) \leq n$. Apesar dessa característica, são várias as aplicações de coloração. Além de colorir mapas, pode-se resolver problemas de escalonamento de tarefas como, por exemplo, definir os horários da prova final das disciplinas de um curso de forma que não haja um aluno com duas provas no mesmo horário; definir como devem ser armazenados produtos incompatíveis, tais como explosivos; determinar como deve ser feita a alocação de frequências em sistemas de comunicação de maneira a evitar interferência, dentre outros. Para um maior aprofundamento no tema, pode-se consultar Boaventura e Jurkiewicz [4].

4.5 O Caixeiro Viajante

O tema pode ser iniciado descrevendo um possível cruzeiro marítimo que é iniciado numa ilha A e que deve visitar uma única vez outras três ilhas, B, C e D, de forma que o navio retorne à ilha A percorrendo a menor distância possível. Considere que as distâncias entre as ilhas são de 30 km entre A e B, 30 km entre B e C, 25 km entre C e D, 40 km entre A e D, 35 km entre B e D e 50 km entre A e C. Pode-se sugerir que modelem o problema através de um grafo. Com apenas quatro vértices, é fácil calcular as distâncias de todas as possíveis rotas.

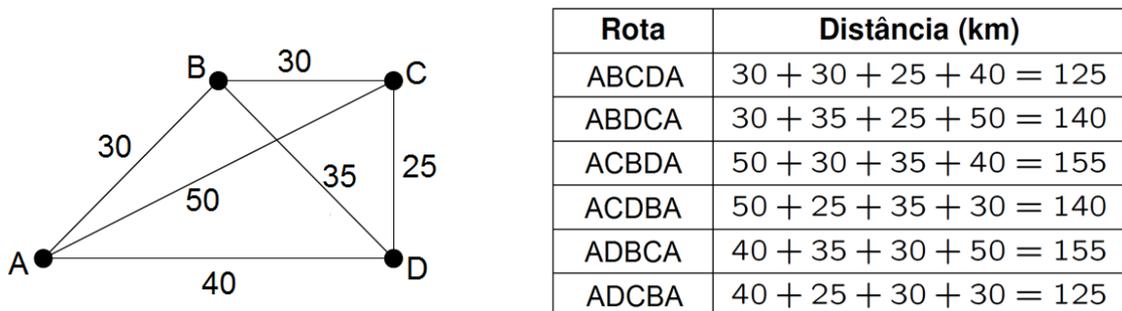


Figura 4.10: Ilhas.

Pode ser interessante propor o mesmo problema aumentando uma ilha. Nesse caso já surge alguma dificuldade, pois são $4! = 24$ rotas possíveis. Mesmo sem apresentar a solução para cinco ilhas, é possível propor que o problema seja solucionado com seis ilhas, o que representa $5! = 5040$ rotas possíveis.

Apresentando, formalmente, o Problema do Caixeiro Viajante é possível trabalhar a história dos grafos hamiltonianos com suas principais definições e teoremas. Utilizando-se de um grafo com cinco vértices pode-se mostrar o funcionamento de algumas heurísticas de construção de rotas, como as citadas no capítulo 2 desse trabalho. Também é interessante mostrar que foram desenvolvidas heurísticas de melhoramento, como as citadas no capítulo 3.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] APPEL, K. L.; HAKEN, W. *Every planar map is four-colorable, part I: discharging*. Illinois Journal of Mathematics, v. 21, p. 429-490, 1977.
- [2] APPEL, K. L.; HAKEN, W. *Every planar map is four-colorable, part II: reducibility*. Illinois Journal of Mathematics, v. 21, p. 491-567, 1977.
- [3] APPEL, K.L.; HAKEN, W. *Every Planar Graph Map is Four Colorable*. American Mathematical Society, 1989.
- [4] BOAVENTURA-NETTO, P. O.; JURKIEWICZ, S. *Grafos: Introdução e Prática*. 1. reimpressão. São Paulo: Edgar Blucher Ltda, 2009.
- [5] BOAVENTURA-NETTO, P. O. *Teoria e Modelos de Grafos*. 2. ed. São Paulo: Edgard Blucher Ltda, 1996.
- [6] BODIN, L.; GOLDEN, B.; ASSAD, A.; BALL, M. *Routing and Scheduling of Vehicles and Crews: The State of the Art*. An International Journal Computers & Operations Research, v. 10, n. 2, p. 79-92, 1983.
- [7] CROES, G. *A method for solving travelling salesman problems*. Operations Research, v. 6, p. 791-812, 1958.
- [8] CARDOSO, D. M. *Sobre o Teorema das Quatro Cores*. Departamento de Matemática, Universidade de Aveiro, 2004.
- [9] DORIGO, M.; BIRATTARI, M.; STUTZLE, T. *Ant colony optimization: Artificial ants as a computational intelligence technique*. IEEE Computational Intelligence Magazine, v. 1, n. 4, p. 28-39, 2006.

- [10] DORIGO, M.; GAMBARDELLA, L. M. *Ant colony system: A cooperative learning approach to the traveling salesman problem*. IEEE Transactions on Evolutionary Computation, v.1, n.1, p. 1-24, 1997.
- [11] DORIGO, M.; MANIEZZO, V.; COLORNI, A. *Ant system: Optimization by a colony of cooperating agents*. IEEE Transactions on Systems, Man, and Cybernetics, Part B, v. 26, p. 29-41, 1996.
- [12] DORIGO, M.; STÜTZLE, T. *Ant Colony Optimization*. Scituate, MA, USA: Bradford Company, 2004.
- [13] EDMONDS, J.; JOHNSON, E. L. *Euler Tours and the Chinese Postman Problem*. Mathematical Programming 5, 1973.
- [14] EISELT, H. A; GENDRAU, M.; LAPORTE, G. *Arc routing problems, part 1: the chinese postman problem*. Operations Research, v.43, 2. ed., p. 231-242, 1995.
- [15] GOLDBARG, M. C.; GOLDBARG, E. F. G. *Grafos: Conceitos, Algoritmos e Aplicações*. Rio de Janeiro: Elsevier, 2012.
- [16] GOLDBARG, M. C.; LUNA, H. P. L. *Otimização Combinatória e Programação Linear*, 2 ed., Rio de Janeiro: Elsevier, 2005.
- [17] GOLDBERG, D. E. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Boston, MA, USA: Addison-Wesley Professional, 1989.
- [18] GONÇALVES, A. L. *Grafos: Aplicações ao Jogo*. Dissertação de Mestrado. Universidade Portucalense. Porto, Portugal, 2007.
- [19] GOODRICH, M. T.; TAMASSIA, R. *Estrutura de dados e algoritmos em Java*. Porto Alegre: Bookman, 2007.

- [20] GOSS, S; ARON, S.; DENEUBOURG, J. L., PASTEELS, J.M. *Self-organized shortcuts in the argentine ant*. *Naturwissenschaften*, Springer-Verlag, v. 76, n. 12, p. 579-581, 1989.
- [21] HARARY, F. *Graph Theory*. Oxford: Addison-Wesley Publishing Company, 1969.
- [22] HOLLAND, J. H. *Adaptation in natural and artificial systems*. Ann Arbor, MI. The University of Michigan Press, 1975.
- [23] HUNTER, D. J. *Fundamentos da matemática discreta*. Tradução Paula Porto Martins. Rio de Janeiro: LTC, 2011.
- [24] INEP *PISA 2012: Resultados*. Brasília, 2016. Disponível em: <<http://portal.inep.gov.br/internacional-novo-pisa-resultados>>. Acesso em 07/05/2016.
- [25] JURKIEWICZ, S. *Grafos - Uma Introdução*. 2012. Disponível em: <<http://www.obmep.org.br/docs/apostila5.pdf>>. Acesso em 05/03/2016.
- [26] KWAN, M. *Graphic Programming using odd or even points*. *Chinese Math*, 1, p. 273-277, 1962.
- [27] LI, R. *A Simulation System for Hierarchical Routing Using Ant Based Control*. Thesis (Master) - Faculty of Information Technology and Systems, Delft University of Technology. Netherlands, 2004.
- [28] LIN, S. *Computer solutions of the traveling-salesman problem*. *Bell System Technology Journal*, v. 44, p. 2245-2269, 1965.
- [29] LIN, S.; KERNIGHAN, B. W. *An Effective Heuristic Algorithm for the Traveling Salesman Problem*. *Operations Research*, v. 21, p. 498-516, 1973.
- [30] LOPES, A. S. *Métodos Finitos em Matemática*. Mestrado em Matemática para Professores. Faculdade de Ciências do Porto. Porto, Portugal, 2009.

- [31] LUCCHESI, C. L. *Introdução à Teoria dos Grafos*. 12^o. Colóquio Brasileiro de Matemática. Rio de Janeiro: IMPA, 1979.
- [32] ORE, O. *Graphs and their Uses*. 2. ed. Washington: The Mathematical Association of America, 1990.
- [33] POTVIN, J. Y. *Genetic algorithms for the traveling salesman problem*. Annals of Operations Research. v. 6, p.339-370, 1996.
- [34] RANGEL, S. *Teoria do Grafos, Notas de aula*. IBILCE, Unesp, 2002-2013
- [35] ROBERTS, F. S. *Applied Combinatorics*. Rutgers University, 1984.
- [36] ROBERTSON, N.; SANDERS, D.; SEYMOUR, P.; THOMAS, R. *A new proof of the Four-Colour Theorem*. Electronic Research announcements of the Amer. Math. Soc., v. 2, n. 1, 1996.
- [37] SAATY, T. L.; KAINEN, P. C. *The four-color problem, assaults and conquest*. New York: Dover, 1986.
- [38] SANTOS, J. P. O.; MELLO, M. P.; MURARI, I. T. C. *Introdução à Análise Combinatória*. Campinas, Editora da Unicamp, 1998.
- [39] SECRETARIA DA EDUCAÇÃO BÁSICA. *Orientações Curriculares para o Ensino Médio*. v. 2. Ciências da Natureza, Matemática e suas Tecnologias. Brasília: MEC, 2006.
- [40] SECRETARIA DE EDUCAÇÃO MÉDIA E TECNOLÓGICA. *Parâmetros Curriculares Nacionais: Ensino Médio*. v. 3. Brasília: MEC, 1998.
- [41] SOUZA, B. A.; SILVA, J. P. S.; FERREIRA, N. R. *Configuração Ótima de Redes de Distribuição Aplicando Um Algoritmo Colônia de Formigas*. EEE PES Transmission and Distribution Latin America Conference and Exposition, São Paulo, 2010.