

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CAMPUS CORNÉLIO PROCÓPIO
DIRETORIA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

GUSTAVO DIAS DE OLIVEIRA

**UM ALGORITMO EFICIENTE PARA ESTIMAR OS MOMENTOS
ESPECTRAIS DE GRAFOS GRANDES NÃO DIRIGIDOS COM PESOS**

DISSERTAÇÃO DE MESTRADO

CORNÉLIO PROCÓPIO

2020

GUSTAVO DIAS DE OLIVEIRA

**UM ALGORITMO EFICIENTE PARA ESTIMAR OS MOMENTOS
ESPECTRAIS DE GRAFOS GRANDES NÃO DIRIGIDOS COM PESOS**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Informática da Universidade Tecnológica Federal do Paraná – UTFPR como requisito para a obtenção do título de “Mestre em Informática”.

Orientador: Prof. Dr. André Yoshiaki Kashiwbara

CORNÉLIO PROCÓPIO

2020

Dados Internacionais de Catalogação na Publicação

O48 Oliveira, Gustavo Dias de

Um algoritmo eficiente para estimar os momentos espectrais de grafos grandes não dirigidos com pesos / Gustavo Dias de Oliveira. – 2020.
49 f. : il. color. ; 31 cm.

Orientador: André Yoshiaki Kashiwabara.
Dissertação (Mestrado) – Universidade Tecnológica Federal do Paraná. Programa de Pós-Graduação em Informática. Cornélio Procópio, 2020.
Bibliografia: p. 45-49.

1. Teoria dos grafos. 2. Análise espectral. 3. Representações dos grafos. 4. Informática - Dissertações. I. Kashiwabara, André Yoshiaki, orient. II. Universidade Tecnológica Federal do Paraná. Programa de Pós-Graduação em Informática. III. Título.

CDD (22. ed.) 004

Biblioteca da UTFPR - Câmpus Cornélio Procópio

Bibliotecário/Documentalista responsável:
Romeu Righetti de Araujo – CRB-9/1676

AGRADECIMENTOS

Primeiramente a **Deus**, que através de Sua permissão, eu pude desenvolver este trabalho, e chegar à esta conquista de vida e sucesso que cheguei hoje. Ele nunca me desamparou nos momentos mais desafiadores e, através da sabedoria de Seu filho **Jesus**, eu pude vencer todos estes desafios, me intuindo a melhor forma de resolver os problemas, e desenvolver este trabalho da melhor forma possível.

Meus sinceros agradecimentos ao meu professor orientador **Prof. Dr. André Yoshiaki Kashiwabara** pelas grandiosas ajudas, orientações e ensinamentos durante este programa de mestrado e desenvolvimento deste trabalho. Com ele, aprendi muita coisa valiosa para minha carreira acadêmica e autonomia.

Agradeço também aos meus professores avaliadores da banca de qualificação **Prof. Dr. Danilo Sipoli Sanches e Prof. Dr. Fabrício Martins Lopes**. Eles me deram valiosas sugestões de melhorias para este trabalho, que me proporcionaram feito evoluir.

Eu também gostaria de agradecer ao professor **Prof. Dr. Eduardo Filgueiras Damasceno**, que também me deu valiosas ajudas, me proporcionando os primeiros preparos e orientações para eu ingressar e seguir bem minha jornada durante o programa de mestrado. Agradeço-o também pelas valiosas ajudas e orientações durante meu processo seletivo de ingresso a este programa, com uma boa redação do texto de avaliação de escrita de um artigo científico, relevante para a avaliação do candidato.

Agradeço também a **minha família**, que nunca deixou de me apoiar em minhas decisões de vida e de carreira. E desde pequeno fizeram comigo um excelente trabalho de educação, superação de obstáculos e desenvolvimento de independência e autonomia, que me trouxeram a capacidade de chegar onde cheguei hoje.

Meus agradecimentos também às **minhas amigas** que, sempre quando saímos para momentos de lazer, me apoiaram e me deram forças, fazendo eu acreditar que eu conseguiria superar todos os obstáculos, chegando à minha vitória. Sempre me acolheram nos momentos de maiores dificuldades, fazendo com que eu reabastecesse as minhas energias, dando-me mais forças para realizar os trabalhos deste mestrado com qualidade e sucesso.

Eu também gostaria de agradecer aos **servidores da UTFPR** como **as psicólogas, médicas, secretárias e pedagoga**, que sempre me ajudaram e me acolheram nos meus momentos de dificuldades.

Agradeço também aos **meus colegas e líderes de trabalho**, que também sempre me apoiaram no meu programa de mestrado. Assim como também contribuíram para grandes evoluções e experiências profissionais na minha área, que também fizeram com que eu automaticamente evoluísse minhas habilidades acadêmicas durante o programa.

RESUMO

OLIVEIRA, Gustavo D.. UM ALGORITMO EFICIENTE PARA ESTIMAR OS MOMENTOS ESPECTRAIS DE GRAFOS GRANDES NÃO DIRIGIDOS COM PESOS. 50 f. Dissertação De Mestrado – Programa de Pós-graduação em Informática, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2020.

Um grafo se caracteriza por um conjunto de vértices e um conjunto arestas, cada uma dela conectando dois vértices. Os momentos espectrais de um grafo são utilizados para caracterizar a topologia do grafo. Os algoritmos para calcular os momentos espectrais, em geral, caracterizam-se por ter complexidade de ordem cúbica, o que inviabiliza o cálculo para grafos grandes na escala de milhões de nós, por demandar um tempo de processamento computacional muito grande. Este trabalho apresenta uma adaptação do algoritmo, descrito por Coheinstainer, para grafos não-dirigidos com pesos. O algoritmo devolve um resultado baseado em uma aproximação com diferença na quarta casa decimal, mas que ainda possa ser utilizado para caracterizar o grafo. O mesmo calculou os momentos para um grafo com mais de 800 mil nós em menos de 2,5 segundos. Esta implementação consiste em apresentar uma solução para calcular os momentos espectrais de grandes grafos em um tempo de processamento computacional viável. Dessa forma, a solução proposta é viável para aplicações em problemas que se possa utilizar este algoritmo em problemas de reconhecimento de padrões. Na implementação, foram feitas adaptações no algoritmo, melhorando para um consumo de tempo próximo à complexidade constante $O(1)$.

Palavras-chave: Grafos. Momento. Momento espectral. Momento espectral de grafo. Redes complexas. Grafos grandes.

ABSTRACT

OLIVEIRA, Gustavo D.. AN EFFICIENT ALGORITHM TO ESTIMATE THE SPECTRAL MOMENTS OF LARGE UNDIRECTED WEIGHTED GRAPHS. 50 f. Dissertação De Mestrado – Programa de Pós-graduação em Informática, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2020.

A graph is a collection of vertexes and edges, each one connecting two vertexes. We can use the spectral moments to characterize the topology of the graph. The algorithms that calculate the spectral moments have a cubic-order complexity, a fact that makes it unfeasible for large graphs with millions of nodes because it demands a considerable computational processing time. This work presents an adaption of the algorithm, described by Cohein-Steiner, for undirected weighted graphs. The algorithm returns a result based on an approximation with difference on the forth demail place, that can still be used to characterize the graph. This one calculated the moments for a graph with more than 800 thousands of nodes in less than 2.5 seconds. This implementation consists in presenting a solution for calculating the spectral moments of large graphs in feasible computational processing time, that makes able to use this algorithm in pattern recognizing problems. On the implementation, some adaptions were made on the algorithm, improving for a runtime close to $O(1)$.

Keywords: Graphs. Moment. Spectral moment. Graph spectral moment. Complex networks. Large graphs.

LISTA DE FIGURAS

FIGURA 1	– Exemplos de grafos	14
FIGURA 2	– Redes complexas (TOLEDO, 2019)	19
FIGURA 3	– Modelos de redes complexas	21
FIGURA 4	– Pesos para seleção por roleta (LIPOWSKI; LIPOWSKA, 2012)	26
FIGURA 5	– Relação com o erro médio absoluto	37
FIGURA 6	– Relação com o tempo de execução	39

LISTA DE TABELAS

TABELA 1	– Algoritmos de cálculo de autovalores de uma matriz com suas complexidades	25
TABELA 2	– Comparação do algoritmo de momentos espectrais de (COHEN-STEINER et al., 2018) com os valores reais. Para uma amostra de rede complexa de BA	35
TABELA 3	– Comparação do algoritmo de momentos espectrais de (COHEN-STEINER et al., 2018) com os valores reais. Para uma amostra de rede complexa de ER	35
TABELA 4	– Comparação do algoritmo de momentos espectrais de (COHEN-STEINER et al., 2018) com os valores reais. Para uma amostra de rede complexa de WS	35
TABELA 5	– Erros médios absolutos dos grafos	36
TABELA 6	– Relação entre s e erro médio absoluto	43
TABELA 7	– Relação entre o número de nós e o erro médio absoluto	43
TABELA 8	– Relação entre o número de arestas e o erro médio absoluto	44
TABELA 9	– Tempo de execução do algoritmo (BA)	44
TABELA 10	– Tempo de execução do algoritmo (ER)	44
TABELA 11	– Tempo de execução do algoritmo (WS)	45
TABELA 12	– Relação entre o número de arestas e o tempo de execução	45

LISTA DE SIGLAS

BA	Barabási-Albert
ER	Erdős-Rényi
WS	Watts-Strogatz

SUMÁRIO

1	INTRODUÇÃO	11
1.1	OBJETIVOS	13
1.2	ORGANIZAÇÃO DO TEXTO	13
2	FUNDAMENTAÇÃO TEÓRICA	14
2.1	GRAFOS	14
2.1.1	Matriz de adjacência	15
2.1.2	Matriz laplaciana	15
2.1.3	Matriz laplaciana normalizada	15
2.1.4	Matriz estocástica	16
2.1.5	Momentos espectrais de um grafo	17
2.2	APLICAÇÕES PARA ESPECTRO DE GRAFOS	17
2.2.1	Química	17
2.2.2	Física	18
2.2.3	Computação	18
2.3	REDES COMPLEXAS	19
2.4	APLICAÇÕES PARA ESPECTRO DE GRAFOS GRANDES	20
2.5	APLICAÇÃO PARA MOMENTOS ESPECTRAIS DE GRAFOS GRANDES NÃO DIRIGIDOS COM PESO	22
2.6	ALGORITMO PARA CÁLCULO DE MOMENTOS ESPECTRAIS DE GRAFOS GRANDES	22
2.7	ALGORITMOS NUMÉRICOS PARA CÁLCULO DE AUTOVALORES DE UMA MATRIZ	23
2.8	INTERPRETAÇÃO PARA OS AUTOVALORES DE UMA MATRIZ DE ADJACÊNCIA	24
2.9	ALGORITMOS DE ROLETA	26
2.9.1	Algoritmo de Lipowski	27
2.10	ERROS E DISPERSÃO EM CÁLCULO NUMÉRICO	28
3	ADAPTAÇÃO E IMPLEMENTAÇÃO DO ALGORITMO COHEN-STEINER PARA GRAFOS NÃO DIRIGIDOS E COM PESOS	30
3.1	CÁLCULO DOS L PRIMEIROS MOMENTOS ESPECTRAIS	30

4 EXPERIMENTOS	33
4.1 RECURSOS UTILIZADOS	33
4.2 AVALIAÇÃO DA ACURÁCIA DO ALGORITMO	34
4.3 TESTES DE ERROS ABSOLUTOS PARA DIFERENTES VALORES DE s , NÚMEROS DE NÓS E DE ARESTAS	36
4.3.1 Relação com o valor de s	36
4.3.2 Relação com o número de nós	36
4.3.3 Relação com o número de arestas	37
4.4 TESTES DE DESEMPENHO DO ALGORITMO	37
4.4.1 Relação com o número de nós	37
4.4.2 Relação com o número de arestas	39
4.4.3 Executando o algoritmo em uma rede real grande	40
5 CONCLUSÕES	41
A TABELAS DOS RESULTADOS DOS EXPERIMENTOS	42

1 INTRODUÇÃO

Existem diferentes sistemas que possuem objetos e relações entre esses objetos, tais como redes de computadores (ZAVLANOS; PRECIADO; JADBABAIE, 2011), sistemas distribuídos (ZAVLANOS; PRECIADO; JADBABAIE, 2011), relações de amizades entre pessoas (VAN-MARCKE, 1972), moléculas compostas por seus átomos (ESTRADA, 1998), redes neurais, entre outros. Estes podem ser representados formalmente com um modelo matemático denominado de **grafo** (MESBAHI; EGERSTEDT, 2010). Esta formalização nos permite realizar diversos trabalhos como análises de relação entre os objetos (DENG et al., 2014), previsões probabilísticas ou determinísticas de ações futuras no comportamento de um sistema (KILGOUR; HIPEL, 2005), reconhecimento de características para reconhecimento de padrões (Bottou; Bengio; Le Cun, 1997), entre outros. Isto pode ser feito através de algoritmos computacionais existentes na teoria dos grafos e de grafos aleatórios (Bottou; Bengio; Le Cun, 1997).

Um exemplo é o problema do **caixeiro viajante** (DORIGO; GAMBARDELLA, 1997). Este consiste, por exemplo, em um sistema de cidades interligadas por rodovias. O objetivo deste problema é, através de um algoritmo, alocar uma sequência de cidades, de forma que visite todas as cidades apenas uma única vez, e em um caminho menos custoso possível. Um exemplo de otimização de caminho é percorrer a menor distância, considerando que as rodovias têm uma quilometragem de uma cidade à outra, e que nem todas as cidades estão diretamente interligadas.

Um segundo exemplo é o problema de **roteamento de pacotes de rede** entre computadores e roteadores em uma rede local ou mundial de computadores. É necessário um algoritmo que percorra o caminho menos custoso para ter a menor latência e maior desempenho possível na comunicação em rede (FLAMMER; KALKWARF, 1990).

Ao representar sistemas modelando com grafos, muitas vezes os mesmos são construídos com um arranjo de ligações aleatórios entre seus nós em grande quantidade. Grafos deste tipo podem ser classificados como redes complexas (NOH; RIEGER, 2004; STROGATZ, 2001). O estudo das redes complexas é muito importante para ajudar a entender como um sistema pode evoluir ao longo do tempo, ou para modelar algum comportamento do sistema através de alterações na topologia do grafo. Para compreender como esses grafos se comportam, é importante aplicar cálculos estatísticos, como por exemplo, medir a média dos caminhos

mais curtos, calcular o coeficiente de *clustering*, heterogeneidade, *betweenness*, elasticidade, conectividade média entre os nós vizinhos, entre outros (COSTA et al., 2007; MENEZES et al., 2009; HOLME, 2005; TOLEDO, 2019).

Uma das formas de se estudar um grafo é calculando o seu **espectro**, medida utilizada para representar a conectividade entre os seus nós (TAKAHASHI et al., 2012). Existem diversas aplicações para espectro de grafos como na química para estudar a força de ligação e geometria das moléculas (MIEGHEM, 2010), na física para estudar as vibrações das membranas (CVETKOVIC; GUTMAN, 2009), e na computação para estudar problemas tais como redes complexas, rede mundial de computadores (internet) e reconhecimento de padrões (MIEGHEM, 2010).

Uma outra forma de se estudar um grafo é através dos seus **momentos espectrais** (Liu; Dong; Wang, 2017), que nos fornecem uma intuição sobre o formato da distribuição do seu espectro (VANMARCKE, 1972). Os momentos espectrais possuem diversas aplicações (Seção 2.2) como estudar a energia das moléculas, considerando as forças de ligação entre seus átomos (ESTRADA, 1998); em física, o problema das vibrações das membranas (VANMARCKE, 1972); medir o quão unida é uma sociedade, considerando as pessoas como os objetos relacionados entre si por relações de amizade (VANMARCKE, 1972); conectividade entre os píxeis de uma imagem, com o objetivo de fazer o processamento da mesma para reconhecer padrões (CVETKOVIC; SIMIĆ, 2010); entre outras.

Para diversos casos, existem aplicações onde demanda-se medir a conectividade entre objetos de um sistema muito grande, na escala de pelo menos milhões de objetos (PRECIADO; JADBABAIE, 2012). Para isto, são necessários algoritmos capazes de calcular momentos espectrais de grafos com uma quantidade de nós nesta escala.

Para calcular os momentos espectrais de um grafo, é necessário calcular os autovalores (Seção 2.7) da matriz que representa o grafo. Porém, o algoritmo que é utilizado para calcular os momentos espectrais, explicado na Seção 2.1.5, demanda multiplicações de matrizes N por N , onde N é no número de nós do grafo. Em uma multiplicação entre duas matrizes A e B é igual a $C = (c_{ij} = \sum_{k=0}^N a_{ik} \cdot b_{kj})$, a sua complexidade é cúbica ($O(n^3)$). Portanto, para um grafo com 1 milhão de nós, levaria $(10^6)^3 = 10^{18}$ operações, o que inviabiliza calcular computacionalmente através deste método clássico.

O trabalho de (COHEN-STEINER et al., 2018) descreve um algoritmo para cálculo dos momentos espectrais. O mesmo explora caminhos aleatórios no grafo e realiza uma amostragem desses caminhos para tentar estimar uma aproximação dos valores. Este algoritmo assume como entrada grafos não dirigidos e sem pesos. Grafos não dirigidos são aqueles que não têm sentido nas arestas, tornando indiferente sobre qual dos nós é origem e qual é destino (THULASIRAMAN; SWAMY, 1992). E grafos com pesos são aqueles que, nas suas arestas, contém um número real, que quanto maior o seu valor, maior o peso (JUNGNICKEL, 2008). A

contribuição deste mestrado, publicado no congresso “ The 8th Brazilian Conference on Intelligent Systems (BRACIS) - 2019” (Dias de Oliveira; Kashiwabara, 2019), foi adaptar o algoritmo original para otimizar o desempenho e também para generalizar o algoritmo para grafos com pesos. Enquanto o artigo original descreve como calcular apenas o l -ésimo momento, o algoritmo adaptado foi capaz de calcular, em menos de 2,5 segundos, os l primeiros momentos para um grafo com aproximadamente 840 mil nós.

Para avaliar o comportamento do algoritmo adaptado, foram realizados alguns experimentos. O primeiro realizou execuções do algoritmo para diferentes amostras de grafos com topologias diferentes, calculando os devidos valores reais, para validar sua assertividade. O segundo executou o algoritmo com diferentes tamanhos de grafo, avaliando os respectivos tempos de execução, para validar sua eficiência. E no terceiro, foi extraído um grafo grande do mundo real e, executando com esta entrada, foi possível validar se o algoritmo executa em um tempo eficiente.

A implementação está livremente disponível em <https://github.com/GuDiasOliveira/graph-spectral-moments>, futuros pesquisadores poderão melhorá-lo.

1.1 OBJETIVOS

Implementar de um algoritmo eficiente para estimar os momentos espectrais de um grafo grande.

1.2 ORGANIZAÇÃO DO TEXTO

O texto do presente trabalho está organizado como a seguir:

- O Capítulo 2 descreve a fundamentação teórica, que servem de base para o entendimento dos procedimentos realizados para desenvolver o trabalho e seus resultados e conclusões, no caso o algoritmo. Há alguns conceitos matemáticos como grafos, momentos espectrais de grafos, de matrizes, redes complexas, entre outros.
- O Capítulo 3 descreve o algoritmo que calcula momentos espectrais de um grafo não-dirigido e com pesos.
- O Capítulo 4 documenta todos os resultados do trabalho e de seus experimentos realizados para a produção dos mesmos. Também descreve a discussão de alguns resultados.
- O Capítulo 5 descreve um levantamento das conclusões do trabalho, baseados em seus resultados, considerando aspectos como possíveis aplicações, embasamento para trabalhos futuros, e contribuições para o avanço científico e tecnológico.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, são apresentados alguns conceitos relevantes como grafos, momentos espectrais de grafos, matriz laplaciana, matriz estocástica, redes complexas, entre outros; que servem de base teórica para o entendimento dos algoritmos, procedimentos e experimentos desenvolvidos. Os conceitos preliminares descritos neste capítulo foram baseados nos artigos (WEST et al., 2001; COHEN-STEINER et al., 2018).

2.1 GRAFOS

Um **grafo** $G = (V, E)$ é composto por um conjunto de nós (ou vértices) V e um conjunto de arestas E , que interligam dois nós com seu respectivo peso (**grafo com peso**) ou não (**grafo sem peso**). O grafo pode ser **direcionado**, quando uma aresta parte de um vértice de origem para um outro de destino em apenas um sentido, ou **não direcionado**, quando uma única aresta interliga dois vértices sem que haja um sentido (THULASIRAMAN; SWAMY, 1992; JUNGnickel, 2008).

A Figura 1 ilustra alguns exemplos de grafos.

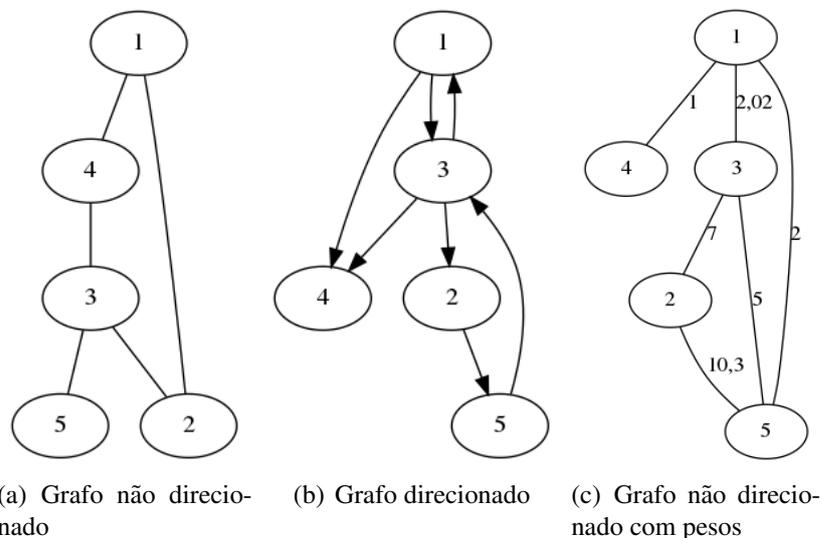


Figura 1: Exemplos de grafos

Existem diversas formas de representar um grafo utilizando matrizes, que serão discu-

tidas nas subseções a seguir.

2.1.1 Matriz de adjacência

A **matriz de adjacência** de um grafo representa os pesos das arestas nas ligações entre seus nós. Um grafo com N nós tem uma matriz de adjacência de números reais A com N linhas e N colunas, ou seja, $A \in \mathbb{R}^{N \times N}$. Na matriz de adjacência, o elemento na i -ésima linha e j -ésima coluna representa o peso da aresta que liga o vértice i do grafo ao vértice j , sendo que o valor 0 representa que não há aresta entre os dois nós (THULASIRAMAN; SWAMY, 1992; JUNGNICKEL, 2008).

Em um grafo não direcionado, por não ter diferença de sentido nas arestas, a sua matriz de adjacência é simétrica ($A = A^t \iff a_{ij} = a_{ji}$).

Em um grafo sem pesos, a sua matriz de adjacência é composta de elementos com apenas dois possíveis valores: 0, se não houver aresta; ou 1 se houver aresta.

2.1.2 Matriz laplaciana

Uma matriz de adjacência também tem sua correspondente matriz laplaciana L , que é definida como $L = D - A$. D é uma matriz diagonal, onde cada um de seus elementos d_{ii} corresponde ao grau do i -ésimo nó do grafo, e o restante dos seus elementos são 0. O grau de um vértice é igual a soma dos pesos de todas as suas arestas que partem dele ($d_i = \sum_{k=1}^N a_{ik}$, sendo d_i o grau do i -ésimo nó). Considerando que os grafos estudados neste trabalho não têm *loops* (arestas que ligam um nó a ele mesmo), ou seja, todos os elementos da diagonal principal da matriz A são 0, a matriz laplaciana é onde cada um dos elementos da diagonal l_{ii} é igual ao grau do i -ésimo nó e para cada um dos elementos restantes $l_{ij} = -a_{ij}$ (Equação 10) (LURIE, 1999).

$$L = \left(l_{ij} = \begin{cases} d_i, & i = j \\ -a_{ij}, & i \neq j \end{cases} \right) \quad (10)$$

2.1.3 Matriz laplaciana normalizada

Também existe a matriz laplaciana normalizada \mathcal{L} , que é definida como na Equação 11, para grafos sem pesos, não direcionados e sem *loops* (CHUNG, 1992).

$$\mathcal{L}(G) = \begin{cases} 1, & i = j \text{ e } d_j \neq 0 \\ -\frac{1}{\sqrt{d_i d_j}}, & a_{ij} \neq 0, \text{ ou seja, vértices } i \text{ e } j \text{ estão conectados} \\ 0, & \text{caso contrário} \end{cases} \quad (11)$$

Calcular o espectro do grafo para este tipo de matriz tem como vantagem a consistência dos seus autovalores para trabalhar com casos de geometria espectral e processos estocásticos, o que tem muitos resultados dados por grafos regulares genéricos (LURIE, 1999).

2.1.4 Matriz estocástica

Os conceitos sobre matriz estocástica foram abordados por (COHEN-STEINER et al., 2018).

A partir da matriz de adjacência, extraí-se a sua **matriz estocástica**, que representa um mapeamento, em um caminho aleatório no grafo, das probabilidades de sair de um nó e ir para um outro.

Uma **matriz estocástica** M tem um elemento m_{ij} , que representa a probabilidade de sair do nó i e ir para o nó j . Por ser uma matriz probabilística, seus elementos são números reais limitados no intervalo entre 0 e 1, incluindo ambos, ou seja, $M \in [0; 1]^{N \times N}$, para um grafo G com N nós.

Para se calcular a matriz estocástica M a partir da matriz de adjacência A , deve-se primeiramente considerar que a probabilidade de sair de um nó e ir para um outro é igual ao seu peso (evento) dividido pela soma dos pesos de todas as arestas que saem daquele nó (espaço amostral). Pela lógica, é possível inferir que, na matriz estocástica, um elemento é calculado dividindo o valor do elemento situado na mesma linha e coluna na A pela soma dos elementos da mesma linha de A . A Equação 12 esquematiza o cálculo da mesma. Vale também lembrar que a soma de cada linha de M é igual a 1 ($\sum_{k=1}^N m_{ik} = 1, \forall i = 1, 2, \dots, N$).

$$M = \left(m_{ij} = \frac{a_{ij}}{\sum_{k=1}^N a_{ik}} \right) \quad (12)$$

Se existe uma matriz inversível X que satisfaça a equação $B = X^{-1}CX$, então dizemos que B e C são similares e assim possuem os mesmos autovalores. A matriz estocástica M é similar a $I - \mathcal{L}$, então os autovalores de M são iguais a um menos os autovalores de \mathcal{L} . Definição de autovalores na Seção 2.7.

2.1.5 Momentos espectrais de um grafo

Os **momentos espectrais de um grafo** são os momentos da distribuição probabilística dos valores do seu espectro (conjunto de autovalores da sua matriz).

A primeira forma de calculá-los, é utilizando a matriz estocástica. O l -ésimo momento espectral é calculado pela soma dos elementos (traço) da matriz elevada a l , dividido pela quantidade de nós ($\frac{\text{Tr } M^l}{N}$) (MIEGHEM, 2010).

A segunda forma é utilizando a matriz laplaciana normalizada (Seção 2.1.3). O i -ésimo autovalor desta matriz corresponde ao i -ésimo momento espectral. Portanto, ao calcular os autovalores de \mathcal{L} , haver-se-á encontrado os N s primeiros momentos espectrais.

Os momentos espectrais nos dá uma intuição sobre o formato da distribuição do seu espectro (Liu; Dong; Wang, 2017), tais como: média ($l = 1$), variância ($l = 2$), assimetria ($l = 3$) (VERSLUIS; STRAETMANS, 2015) e curtose ($l = 4$) (SEIER; BONETT, 2003).

A matriz estocástica pode ser utilizada para se calcular os momentos espectrais da matriz laplaciana normalizada. Considerando que as matrizes laplaciana normalizada \mathcal{L} (Seção 2.1.3) e $I - M$ são similares, ambas possuem os mesmos autovalores. Portanto, cada autovalor λ_i de $I - M$ corresponde ao autovalor de M igual a $1 - \lambda_i$ (COHEN-STEINER et al., 2018). Esta observação será útil para se calcular o momento espectral do grafo.

2.2 APLICAÇÕES PARA ESPECTRO DE GRAFOS

Os artigos (CVETKOVIC; GUTMAN, 2009; SPIELMAN, 2007; MIEGHEM, 2010) apresentam alguns exemplos de aplicações para espectro de grafos.

2.2.1 Química

O estudo de espectro de grafos é motivado pela sua aplicação no estudo de moléculas, que podem ser representados como grafos.

Algumas aplicações para o espectro são o estudo das energias das moléculas, que está relacionado à conectividade, pois quanto mais conectados, mais forte será a ligação das moléculas; e o estudo da teoria orbital molecular, pois a geometria das moléculas está ligada à conectividade entre os átomos, por causa da lei de atração e repulsão nas ligações entre os átomos (MIEGHEM, 2010).

2.2.2 Física

Uma das motivações de estudo está na aplicação para o problema de vibrações de membranas. Os movimentos vibratórios estão relacionados com a conectividade entre as partículas vibradoras, pois uma maior ou menor conectividade, devido aos contatos entre si, influencia nas características da vibração como duração, frequência, amplitude das ondas vibratórias, geometria da direção da propagação das mesmas, e distância da propagação (relacionada à perda de potência no caminho da propagação) (CVETKOVIC; GUTMAN, 2009).

Uma outra aplicação está em um problema que envolvem moléculas diatômicas (que possui 2 átomos). Projetando uma visão bidimensional em uma superfície, é possível observar um arranjo das moléculas em um grid, onde todas as posições são ocupadas de diferentes maneiras sem sobrepor uma a outra. Para estudar este caso, modela-se grafos, onde fatoriza para subgrafos abrangentes (com todos os nós) 1-regular (todos os seus vértices têm exatamente 1 vizinho), com diferentes possibilidades de conexão entre nós. O problema está em enumerar todas possíveis maneiras de fatorizar grafos abrangentes 1-regular. Para isto, percorre-se caminhos no grafo e calcula-se seus autovalores da matriz de adjacência, ou seja, espectro do grafo (SPIELMAN, 2007).

2.2.3 Computação

Existem muitas aplicações para redes complexas na computação, tais como: otimização combinatoria, redes complexas, internet, mineração de dados, computação visual, reconhecimento de padrões, proteção anti-vírus, redes sociais, bancos de dados estatísticos e computação quântica (MIEGHEM, 2010).

Um dos exemplos é no reconhecimento de padrões e processamento de imagens, pois têm relação com conectividade entre distâncias entre amostras e *pixels*, respectivamente. A conectividade pode ser um fator relevante para a classificação de padrões ou imagens para algumas aplicações (MIEGHEM, 2010).

Outra aplicação é para redes complexas (definição na Seção 2.3), que são grafos grandes com uma topologia, mas não trivial, ou seja, não exata, mas sim como uma tendência de arranjos entre os nós. Um exemplo disso são relações de amizade entre pessoas que, na topologia, tende a ser mais conectada entre nós (pessoas) de mesma cidade ou bairro e menos conectadas entre cidades e bairros diferentes (MIEGHEM, 2010).

Um exemplo de aplicação, que também envolve redes complexas, é uma rede social. A conectividade, ou seja o espectro do grafo, pode ser métrica para analisar fatores como laços de amizade e união em uma comunidade ou grupo de pessoas (MIEGHEM, 2010).

2.3 REDES COMPLEXAS

Redes complexas (Figura 2) são grafos com uma grande quantidade de nós que possuem uma topologia não trivial, ou seja, não como uma regra que a restringe nesta topologia, mas uma tendência aleatória padronizada. Possui aplicações interdisciplinares na biologia, matemática, computação, química, redes sociais, entre outras (MENEZES et al., 2009; SILVA; ZHAO, 2016).

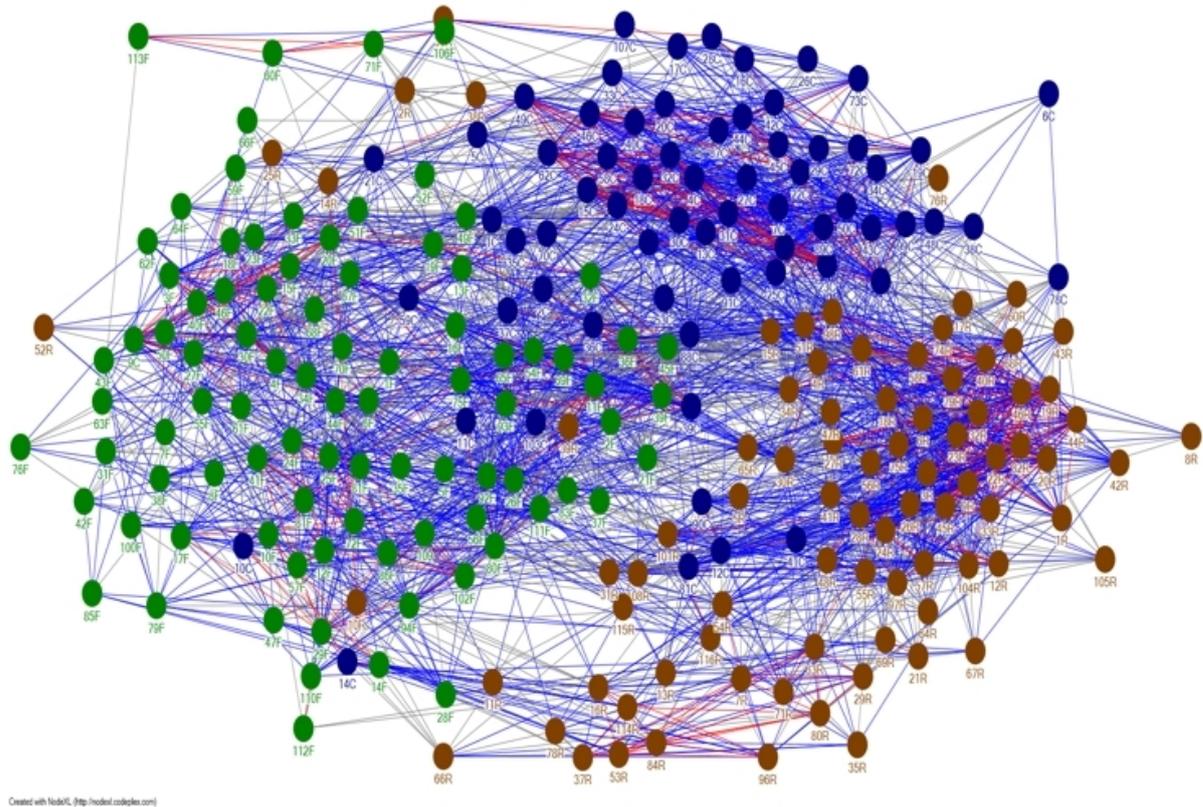


Figura 2: Redes complexas (TOLEDO, 2019)

São vários modelos teóricos existentes de redes complexas, alguns dos mais conhecidos são:

- **Redes aleatórias:** Modelada por Erdős e Rényi (ERDŐS; RÉNYI, 1960). Figura 3(a). As conexões entre nós estão distribuídas aleatoriamente, sem nenhuma padronização topológica. Trata-se de um grafo $G(N, p)$ com N nós, com arestas geradas com uma probabilidade constante p (NOH; RIEGER, 2004);
- **Redes de mundo pequeno:** Grafos cujos caminhos entre os nós são curtos (Figura 3(b)). Foi modelada por Duncan Watts e Steven Strogatz (WATTS; STROGATZ, 1998). Alguns exemplos deste tipo de rede no mundo real são redes de energia elétrica, redes neurais, redes sociais e mapas de estradas (WANG; CHEN, 2003);

- **Redes livre de escala:** Modelada por Barabási e Albert (BARABÁSI; ALBERT, 1999), tem uma topologia similar à estrela, de grafos simples. Alguns vértices, com alto grau, possuem muitos nós na sua periferia conectados, podem ser mencionados como HUBs e tem ligações com outros HUBs. Um exemplo deste modelo são a rede mundial de computadores, com HUBs sendo os roteadores das residências e empresas conectados uns aos outros, e cada um conectando seus dispositivos (nós da periferia) na sua rede local. Ela segue a lei de distribuição de potências, com a probabilidade de o nó ter k ligações $P(k) \sim k^{-\gamma}, k > 0, \gamma > 0$ (BARABÁSI; BONABEAU, 2003). Figura 3(c);
- **Redes aleatórias clusterizadas:** Os nós são divididos em grupos altamente conectados, e com poucas conexões entre nós de grupos diferentes (NEWMAN, 2003; MENEZES et al., 2009). Alguns exemplos do mundo real são: reconhecimento de padrões (FOGGIA et al., 2007), aprendizagem de máquina (Bezdek; Tsao; Pal, 1992), otimização de arquitetura de redes de micro-sensores sem fios (Heinzelman; Chandrakasan; Balakrishnan, 2002) e classificação de dados (ERMAN; ARLITT; MAHANTI, 2006);
- **Redes de periferias de núcleo:** Possui um subgrafo altamente denso, com o restante dos nós nas suas periferias conectados ao núcleo, com poucas arestas (Figura 3(d)) (HOLME, 2005).

2.4 APLICAÇÕES PARA ESPECTRO DE GRAFOS GRANDES

Existem diversas áreas nas quais se pode utilizar espectro de grafos (BANERJEE, 2008). Por exemplo:

- **Redes complexas** (definição na Seção 2.3). Por exemplo, em uma rede mundial de computadores com quase bilhões de nós interconectados. Calcular o espectro destas é uma forma de medir a sua conectividade (MIEGHEM, 2010);
- **Rede mundial de computadores.** Calcular o espectro do grafo é uma alternativa para se obter métricas e tentar otimizar o desempenho das entregas dos pacotes entre as estações, considerando fatores como as posições geográficas e latência entre dois roteadores. Para isto, calcular o espectro deste grande grafo é uma alternativa para solução (Gkantsidis; Mihail; Zegura, 2003);
- **Interações intramoleculares.** Como as forças intramoleculares estão relacionadas com as forças intermoleculares, há uma influência de energia de ligação entre as moléculas, que podem representar um grande grafo com milhões de moléculas. Calculando o espectro, é possível extrair características desta energia entre as moléculas (ESTRADA, 1996);

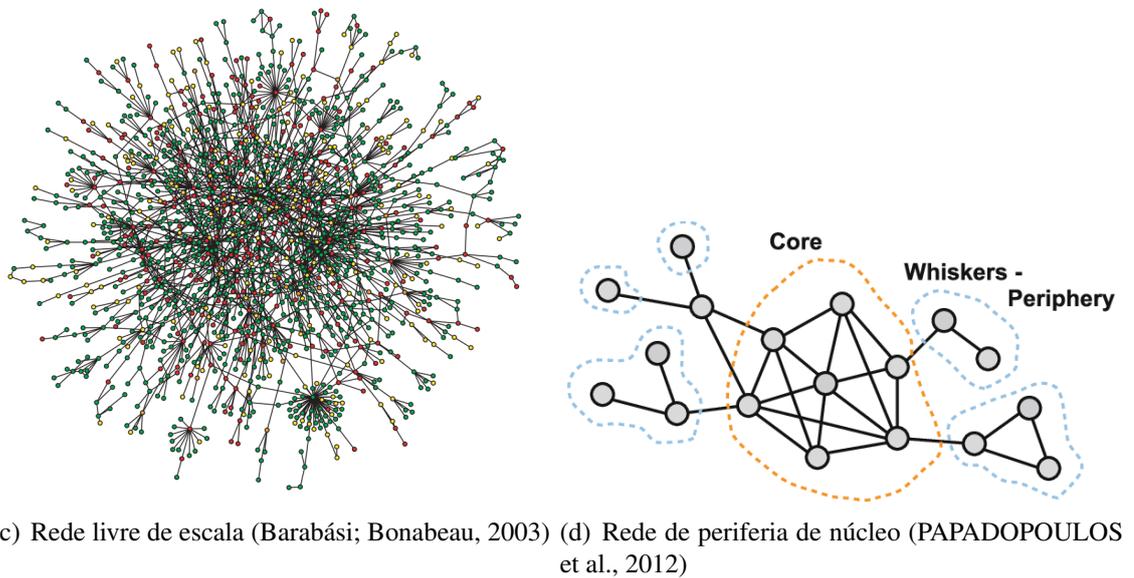
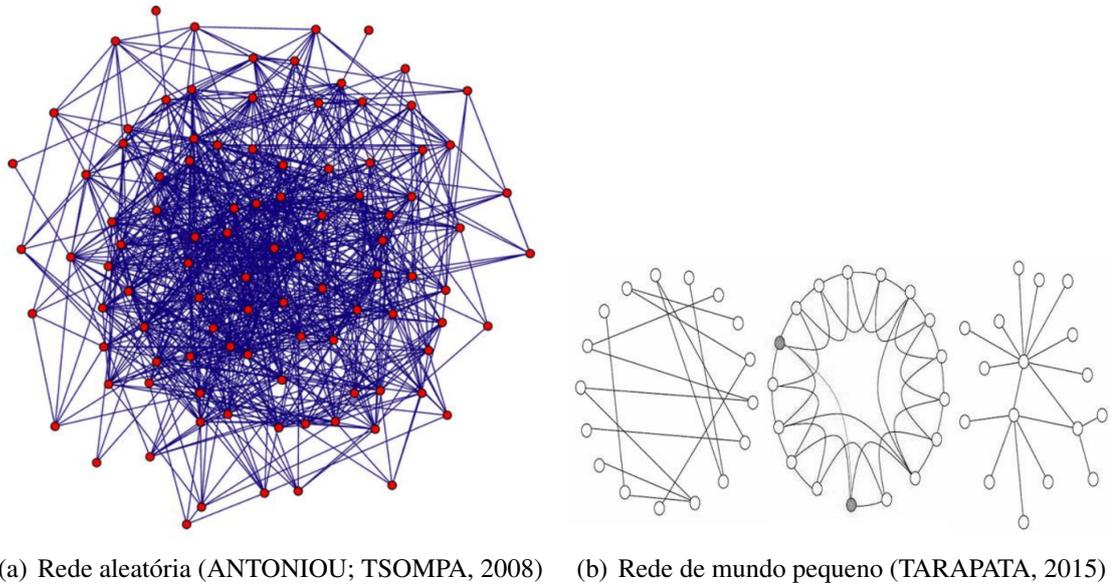


Figura 3: Modelos de redes complexas

- **Outros exemplos:** conectividade entre neurônios, classificação de topologias de redes complexas (TAKAHASHI et al., 2012), indexação de estruturas hierárquicas (Shokoufandeh et al., 2005), entre outras (DOROGOVTSEV et al., 2003).

Cada modelo de redes complexas é caracterizada ter alta ou baixa conectividade desta rede ou suas sub-redes. Esta necessidade de avaliar a conectividade é um dos fatores motivantes deste trabalho.

2.5 APLICAÇÃO PARA MOMENTOS ESPECTRAIS DE GRAFOS GRANDES NÃO DIRIGIDOS COM PESO

Os grafos grandes não dirigidos com pesos são muito utilizados em neurociência. Nesta área, grafos deste tipo são modelados para representar conexões no cérebro. Este modelo é chamado de **conectome**, com grafos com milhões de nós e arestas com pesos, representando as densidades da fibra (GASTNER; ÓDOR, 2016; BETZEL et al., 2016).

O trabalho de (Bonner et al., 2016) descreve uma aplicação, que consiste em classificar topologias de grafos. Em determinadas situações, faz-se necessário acolher uma amostra de rede complexa e, a partir desta, estimar a qual topologia mais se aproxima, classificando-a.

Outra aplicação bastante útil é descrita por (ZHOU et al., 2007). Consiste em medir a energia de um grafo. São várias as utilidades para se medir a energia de um grafo, tais como: medir a energia entre as moléculas; força magnética entre corpos; e tensão mecânica interna nos sólidos.

O trabalho de (BALASUBRAMANIAN, 1990) descreve uma aplicação com cadeias carbônicas (ligações com átomos de carbono). Em determinadas aplicações, faz-se necessário descobrir o formato das moléculas de cadeia carbônica. Com passeios nos átomos e com momentos espectrais, é possível estimar uma classificação da sua topologia.

Portanto, em algumas ocasiões faz-se necessário identificar alguns padrões no formato destas redes. O presente trabalho se faz útil para ser aplicado nestes casos.

2.6 ALGORITMO PARA CÁLCULO DE MOMENTOS ESPECTRAIS DE GRAFOS GRANDES

Como descrito na Seção 2.1.5, uma das formas de se calcular os momentos espectrais de um grafo é calculando o traço de M^l . Por se tratar de multiplicações de matrizes, a sua complexidade é $O(n^3)$, tornando ineficiente para o processamento para grafos grandes.

Para buscar uma solução para este problema, alguns trabalhos relacionados propõem diferentes soluções, porém com suas respectivas restrições.

O trabalho de (Liu; Dong; Wang, 2017) descreve uma dedução de uma expressão para o cálculo dos **quatro primeiros momentos**, baseada em uma aproximação proposta por (Preciado et al., 2010). Sua complexidade é $O(n)$.

Como descrito na Seção 2.1.5, a outra maneira de calcular os momentos espectrais é calculando os autovalores da matriz laplaciana normalizada. Entretanto, como demonstrado na Seção 2.7, é uma operação que demanda multiplicação de matrizes, portanto sua complexidade também é $O(n^3)$.

Para buscar uma solução para otimizar o desempenho, a Seção 2.7 descreve alguns trabalhos relacionados. Os mesmos descrevem diferentes algoritmos para cálculos de autovalores, com suas respectivas complexidades algorítmicas e restrições. No entanto, como é observável, no melhor caso, a sua complexidade é reduzida apenas para quadrática $O(n^2)$. Para um grafo com 1 milhão de nós, levaria $(10^6)^2 = 10^{12}$ operações.

Considerando estas limitações que cada solução tem com relação a restrições e desempenho. O presente trabalho tem este fato como motivação para implementar um algoritmo baseado em (COHEN-STEINER et al., 2018), com complexidade sublinear e possibilitando calcular uma quantidade determinada de primeiros momentos.

2.7 ALGORITMOS NUMÉRICOS PARA CÁLCULO DE AUTOVALORES DE UMA MATRIZ

Seja uma matriz $A \in \mathbb{C}^{N \times N}$, ou seja, uma matriz quadrada de números reais ou complexos de tamanho N . Seus **autovalores** são todos os possíveis valores reais ou complexos para λ , com seus respectivos **autovetores** v , que satisfazem a Equação 13. Considerando também que $\lambda \in \mathbb{C}$ e $v \in \mathbb{C}^N$ (v é um vetor de tamanho N)(JOHNSON; SAIAGO, 2018).

$$Av = \lambda v \quad (13)$$

Se a matriz A for simétrica, ou seja, for igual à sua transposta, seus autovalores serão todos reais. (Equação 14).

$$(A = A^t \iff a_{ij} = a_{ji}) \implies \lambda \in \mathbb{R} \quad (14)$$

O método clássico de cálculo de autovalores, apresentado no artigo (JOHNSON; SAIAGO, 2018), que consiste no desenvolvimento das multiplicações das matrizes Av e λv na Equação 13, incorre no problema de resolução de uma equação polinomial de ordem N (tamanho da matriz), cujo algoritmo é de complexidade cúbica ($O(n^3)$) (PAIGE, 1971). Fato este que inviabiliza o cálculo de autovalores para matrizes grandes até mesmo com tamanhos na ordem de 100 por 100.

Tendo isto como motivação, foram modelados algoritmos que utilizam métodos numéricos dando resultados significativamente aproximados, tornando-se de menor complexidade. O trabalho de (AMERICO, 2013) faz uma revisão de vários destes algoritmos, alguns com restrição para matrizes simétricas.

Pelo fato deste trabalho considerar grafos não direcionados, o mesmo explora os algoritmos com restrição para matrizes simétricas.

Alguns dos algoritmos analisados como o Algoritmo QR (KRESSNER; SCHRÖDER; WATKINS, 2009), o Dividir para Conquistar (TISSEUR; DONGARRA, 1999) e o Método de Jacobi (WILKINSON, 1965), apresentam otimizações significativas em relação ao método clássico, com algoritmos de complexidade, no máximo, cúbica ($O(n^3)$), viabilizando os mesmos para cálculos de autovalores para matrizes com tamanho na ordem de aproximadamente 200.

Entretanto, o objetivo é buscar um algoritmo eficiente para cálculo de espectro de grafos grandes. Por isto, ter-se-á que trabalhar com matrizes de adjacência com tamanho na ordem de 1 milhão de nós no grafo que, para o qual sendo executado algoritmo de complexidade cúbica ficaria: $(1 \text{ milhão})^3 = (10^6)^3 = 10^{18}$, resultando em um tempo de processamento muito alto. Fato este que inviabiliza estes algoritmos para grafos grandes.

A Tabela 1 faz um comparativo de alguns algoritmos de autovalores, revisados pelo artigo (AMERICO, 2013).

Analisando este comparativo, é possível observar que, no melhor caso, a complexidade algorítmica ainda é quadrática $O(n^2)$. Neste caso, o processamento com uma matriz com tamanho de 1 milhão demandará $(10^6)^2 = 10^{12}$ operações. Portanto, terá baixo desempenho para matrizes grandes.

Além disso, observa-se que, para cada algoritmo, possui restrições para tipos de matrizes que podem ser aplicadas. Portanto, nem todos os grafos serão aplicáveis para calcular os momentos espectrais.

2.8 INTERPRETAÇÃO PARA OS AUTOVALORES DE UMA MATRIZ DE ADJACÊNCIA

Na matemática, existe o conceito de transformação linear. O mesmo consiste em uma função, cujos operandos são vetores em um espaço com n dimensões ($f : R^n \rightarrow R^n$), que pode também ser representada com uma matriz de transformação A ($A \cdot \vec{v}_{entrada} = \vec{v}_{saída}$) (CULLEN, 2012).

Seja uma matriz de transformação A aplicada a um vetor \vec{v} , cujo resultado da transformação é um vetor apontando para a mesma direção espacial, porém multiplicado seu módulo por λ . Diz-se que λ é um **autovalor** de A , cujo **autovetor** correspondente é \vec{v} ($A \cdot \vec{v} = \lambda \cdot \vec{v}$) (VLASE; MARIN; ÖCHSNER, 2018).

Em uma matriz de adjacência de um grafo, o maior autovalor (λ_1) é dado por $d_{máx} \geq \lambda_1 \geq d_{médio}$ (ROWLINSON, 2007). Enquanto o seu menor autovalor, quando o grafo é k -regular ($\forall k = 1, 2, \dots, N$) e bipartido, caracteriza-se por ser o oposto do maior autovalor ($\lambda_N = -\lambda_1$) (DESAI; RAO, 1994).

Tabela 1: Algoritmos de cálculo de autovalores de uma matriz com suas complexidades

Algoritmo	Descrição	Restrição	Complexidade
Algoritmo QR (KRESSNER; SCHRÖDER; WATKINS, 2009)	Faz sucessivas divisões da matriz em duas multiplicadas: Q e R , uma matriz ortogonal e outra triangular, respectivamente. Comuta a multiplicação entre Q e R na próxima iteração.	Matrizes de Hessenberg.	$O(n^2)$
Algoritmo de Jacobi (WILKINSON, 1965)	Faz sucessivas rotações na matriz para tentar zerar os elementos fora da diagonal principal.	Matrizes simétricas e reais.	$O(n^3)$
Método das potências (YUAN; ZHANG, 2013)	Iterativamente, aplica um vetor inicial aleatório e normaliza-o.	Retorna apenas o maior autovalor.	$O(n^2)$
Quociente de Rayleigh (SIMONCINI; ELDÉN, 2002)	Itera $(A - \mu_i \cdot I)^{-1}$, com μ_i sendo o resultado da iteração anterior.	Apenas para matrizes hermíticas, e retorna apenas o maior autovalor.	$O(n^2)$
Dividir para conquistar (TISSEUR; DONGARRA, 1999)	Divide a matriz em submatrizes, diagonaliza e recombina-a.	Matriz hermítica tridiagonal.	$O(n^2)$

2.9 ALGORITMOS DE ROLETA

Ao trabalhar com grafos com pesos, a matriz probabilística tem distribuições não uniforme nas arestas. Para realizar caminhos aleatórios neste grafo, uma das formas é utilizando um algoritmo de roleta.

O algoritmo de roleta consiste nestes passos: (i) construção de uma linha de segmento, esquematizado na Figura 4, com os pesos dos elementos w_1, w_2, \dots, w_N ; (ii) sortear aleatoriamente um número real r , com $0 \leq r < \sum_{i=0}^N$; (iii) realizar uma busca para encontrar em qual dos pesos na linha de segmento o r está. A complexidade do algoritmo é definido na etapa de busca, que pode ser sequencial (Algoritmo 1 $O(n)$) ou binária (Algoritmo 2 $O(\log_2 n)$) (LIPOWSKI; LIPOWSKA, 2012; GOLDENBERG, 1989).

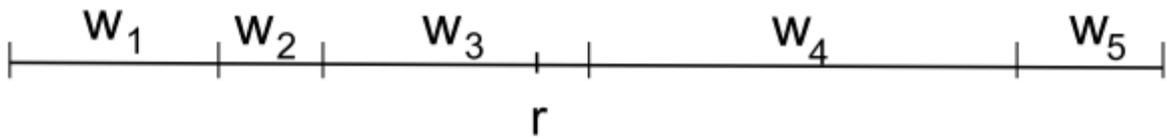


Figura 4: Pesos para seleção por roleta (LIPOWSKI; LIPOWSKA, 2012)

Algoritmo 1: Algoritmo de roleta com busca sequencial (LIPOWSKI; LIPOWSKA, 2012)

Entrada: Lista de probabilidades $P \in (p_1, p_2, \dots, p_n)$

Saída: Elemento sorteado $j \in (1, 2, \dots, n)$

início

```

1 |  $t \leftarrow$  Um número aleatório  $0 \leq t < \sum_P p_i$ 
2 | para cada  $k \in (1, 2, \dots, n)$  faça
3 | | se  $t < p_k$  então
4 | | | retorna  $k$ 
5 | | | senão
   | | |  $t \leftarrow t - p_k$ 
   | | fim
   | fim
  | fim

```

Algoritmo 2: Algoritmo de roleta com busca binária (LIPOWSKI; LIPOWSKA, 2012)

Entrada: Lista de probabilidades $P \in (p_1, p_2, \dots, p_n)$ e $p_0 = 0$

Saída: Elemento sorteado $j \in (1, 2, \dots, n)$

início

```

1  |   $t \leftarrow$  Um número aleatório  $0 \leq t < \sum_P p_i$ 
2  |   $l \leftarrow 1$ 
3  |   $r \leftarrow n$ 
4  |  enquanto  $l \leq r$  faça
5  |  |    $m \leftarrow \left\lfloor \frac{l+r}{2} \right\rfloor$ 
6  |  |   se  $\sum_{i=0}^{m-1} p_i \leq t < \sum_{i=0}^m p_i$  então
7  |  |   |   retorna  $m$ 
8  |  |   |   senão
9  |  |   |   |   se  $t < \sum_{i=0}^{m-1} p_i$  então
10 |  |   |   |   |    $r \leftarrow m - 1$ 
    |  |   |   |   |   senão
    |  |   |   |   |   |    $l \leftarrow m + 1$ 
    |  |   |   |   |   |   fim
    |  |   |   |   |   fim
    |  |   |   |   fim
    |  |   |   fim
    |  |   fim
    |  fim

```

2.9.1 Algoritmo de Lipowski

Como explicado na Seção 3.1, o algoritmo de roleta será um limitador para o de Cohen-Steiner, tornando-o com complexidade $O(n)$ e não mais sublinear.

Observando a Seção 2.9, no melhor caso, a complexidade será logarítmica ($O(\log_2 n)$), sendo a escolha da busca binária uma possível solução. Porém, para otimizar ainda mais o desempenho, foi utilizado o algoritmo de Lipowski (LIPOWSKI; LIPOWSKA, 2012), esquematizado no Algoritmo 3, cuja complexidade é constante $O(1)$, não havendo necessidade de percorrer todas as probabilidades.

Algoritmo 3: Algoritmo de roleta de (LIPOWSKI; LIPOWSKA, 2012)

Entrada: Nó atual i , número de iterações σ , matriz estocástica $M \in [0; 1]^{N \times N}$

Saída: Próximo nó j

início

```

1  |   para cada  $k \in (1, 2, \dots, \sigma)$  faça
2  |       |    $j \leftarrow$  Sorteado uniformemente, com  $j \in (1, 2, \dots, N)$ 
3  |       |    $r \leftarrow$  Número real sorteado, com  $0 \leq r < 1$ 
4  |       |   se  $r < m_{ij}$  então
5  |       |       |   retorna  $j$ 
6  |       |       |   fim
   |       |   fim
   |   retorna  $j$ 
   fim

```

Porém, o algoritmo possui uma limitação. Quando há alguns pesos com uma grande diferença em relação aos demais, o mesmo apresentará lentidão. O algoritmo é mais eficiente para quando os pesos estão mais uniformemente distribuídos (LIPOWSKI; LIPOWSKA, 2012).

O número médio de tentativas do algoritmo de Lipowski ao sair do nó y ao nó j é de $\tau = d_y \times \frac{x}{\sum_j m_{ij}} = d_y \times x$. No pior caso, há algumas arestas com probabilidade $x \sim 1$, portanto o número médio de tentativas será de $\tau \sim d_y$.

2.10 ERROS E DISPERSÃO EM CÁLCULO NUMÉRICO

Nesta seção, serão explicados alguns conceitos sobre erros em cálculo numérico, que são muito utilizados em experimentos científicos. Estes conceitos são relevantes para o entendimento dos resultados no Capítulo 4.

Em experimentos científicos, para a comprovação da precisão dos valores e validação, são feitas comparações entre os valores reais e os obtidos no experimento. Algumas das medidas utilizadas para o mesmo são o **erro absoluto** e o **erro relativo**, conceituados por (MALIK, 2003).

O **erro absoluto** ϵ_x é o módulo (sem sinal) da diferença entre o valor real x e o valor do experimento \bar{x} (Equação 15).

O **erro relativo** δ_x é a taxa de erro absoluto em relação ao valor real (Equação 16).

$$\epsilon_x = |x - \bar{x}| \quad (15)$$

$$\delta_x = \frac{\epsilon_x}{|x|} \times 100\% \quad (16)$$

Quando um experimento é realizado com vários conjuntos de amostras, haverá um erro absoluto para cada. Neste caso, uma das medidas utilizadas é o **erro médio absoluto**, que é a média aritmética dos erros absolutos dos conjuntos de amostras (WILLMOTT; MATSUURA, 2005).

Muitas vezes, é também necessário avaliar a sua dispersão entre os valores do experimento. Duas destas medidas é o **desvio padrão**, que é um valor na mesma unidade de medida dos dados que mede o quanto os valores se dispersaram (AGARWAL, 2006); e o **coeficiente de variação**, que corresponde ao valor percentual, que é dado pela taxa do desvio padrão s em relação à média μ (Equação 17) (KOOPMANS; OWEN; ROSENBLATT, 1964).

$$C_V = \frac{s}{\mu} \times 100\% \quad (17)$$

3 ADAPTAÇÃO E IMPLEMENTAÇÃO DO ALGORITMO COHEN-STEINER PARA GRAFOS NÃO DIRIGIDOS E COM PESOS

Nesta seção, são apresentados os algoritmos, técnicas e modelos utilizados, com o objetivo de ter um algoritmo computacionalmente eficiente para grafos grandes com pesos. O objetivo é adaptar o algoritmo apresentado em (COHEN-STEINER et al., 2018), e desenvolver uma implementação na linguagem de programação C. Em seguida, será apresentado como o algoritmo foi avaliado considerando a acurácia e o tempo empírico de execução para grafos de diferentes de tamanhos.

3.1 CÁLCULO DOS L PRIMEIROS MOMENTOS ESPECTRAIS

A matriz estocástica M possui seus autovalores compreendidos no intervalo entre -1 e 1 e ordenados em ordem crescente, ou seja, $-1 \leq \lambda_1 \leq \dots \leq \lambda_n \leq 1$ (FIEBERG; ELLNER, 2001).

O l -ésimo momento do espectro de M é dado pela média aritmética dos seus autovalores elevados a l , ou seja, $\frac{1}{n} \sum_{k=1}^n \lambda_k^l$, pois M^l representa as probabilidades de sair de um nó de origem i chegar em um destino j em l passos.

O **traço** (soma dos elementos da diagonal principal) de M^i é igual à soma dos seus autovalores elevados a i . Ou seja, $\text{Tr}(M^i) = \sum_{k=1}^n \lambda_k^i$.

Ou seja, o traço de M^i dividido pelo número de nós do grafo é igual ao i -ésimo momento espectral, ou seja, $l_i = \frac{\text{Tr}(M^i)}{n}$.

A matriz estocástica, como explicado na Seção 2.1.4, representa cada probabilidade de sair de um nó e ir para o outro. A mesma multiplicada k vezes, ou seja, M^k , representa esta mesma probabilidade, porém em k passos. Isto é, o elemento na linha i e na coluna j de M^k representa a probabilidade de sair do nó j e parar no nó i em k passos dados no caminho do grafo.

Baseado nestas premissas, o i -ésimo elemento na diagonal principal de M^k representa a probabilidade de, em k passos, sair do nó i e voltar nele mesmo. Considerando que a soma destas probabilidades de cada nó é igual a $\text{Tr}(M^k)$, a média aritmética é $\frac{\text{Tr}(M^k)}{n}$. Logo a média das probabilidades de cada nó de sair dele e voltar nele mesmo em l passos é igual ao l -ésimo

momento espectral.

O Algoritmo 4, o original de Cohen-Steiner que calcula apenas o l -ésimo momento, se baseia em fazer caminhos aleatórios no grafo. Em cada iteração, sorteia-se um dos nós do grafo e, a partir deste nó, percorre os nós através das arestas conectadas até percorrer l passos; a escolha do próximo nó a ir é aleatória, com uma probabilidade de sair do nó que está para ir para cada nó fornecida na matriz estocástica M , que representa a distribuição de probabilidade de caminhos nos nós no grafo (Seção 2.1.4). Após fazer s iterações, o algoritmo retorna a porcentagem (entre 0 e 1) das iterações cujo último nó do caminho retornou ao nó inicial do mesmo, que corresponde ao valor do l -ésimo momento; quanto maior número de iterações s , mais preciso será o resultado; o grafo tem n nós.

Algoritmo 4: Momento espectral aproximado (COHEN-STEINER et al., 2018)

Entrada: $M \in [0; 1]^{n \times n}$, l , s

Saída: O l -ésimo momento espectral

início

```

1  |   $x \leftarrow 0$ 
2  |  para cada  $i \in (1, 2, \dots, s)$  faça
3  |  |   $j \leftarrow$  Um número inteiro  $1 \leq j \leq n$  sorteado aleatoriamente
4  |  |   $w \leftarrow j$ 
5  |  |  para cada  $k \in (1, 2, \dots, l)$  faça
6  |  |  |   $w \leftarrow t$ , com  $t \in \mathbb{N}$  sorteado aleatoriamente  $1 \leq t \leq n$ , com
   |  |  |  probabilidade  $P(t) = m_{tw}$ 
   |  |  fim
7  |  |  se  $w = j$  então
8  |  |  |   $x \leftarrow x + 1$ 
   |  |  fim
   |  fim
9  |  retorna  $\frac{x}{s}$ 
   |  fim

```

Para otimizar o desempenho, foi feita uma adaptação no algoritmo que, ao invés de retornar apenas o l -ésimo momento, retorna todos os primeiros l momentos. No algoritmo original é feito um caminho no grafo em l passos, para depois verificar se retornou ao nó de origem. No algoritmo adaptado, a cada passo é contabilizada a quantidade de vezes que retornou ao nó de origem, retornando um vetor de porcentagem de vezes com l elementos, sendo este os l primeiros momentos. O Algoritmo 5 esquematiza a adaptação feita.

Algoritmo 5: Momento espectral aproximado (COHEN-STEINER et al., 2018)

Entrada: $M \in [0; 1]^{n \times n}$, l , s
Saída: Os l primeiros momentos espectrais

início

```

1   |   para cada  $k \in (1, 2, \dots, l)$  faça
    |   |    $x[k] \leftarrow 0$ 
    |   |   fim
2   |   para cada  $i \in (1, 2, \dots, s)$  faça
3   |   |    $j \leftarrow$  Um número inteiro  $1 \leq j \leq n$  sorteado aleatoriamente
4   |   |    $w \leftarrow j$ 
5   |   |   para cada  $k \in (1, 2, \dots, l)$  faça
6   |   |   |    $w \leftarrow t$ , com  $t \in \mathbb{N}$  sorteado aleatoriamente  $1 \leq t \leq n$ , com
    |   |   |   probabilidade  $P(t) = m_{tw}$ 
7   |   |   |   se  $w = j$  então
8   |   |   |   |    $x[k] \leftarrow x[k] + 1$ 
    |   |   |   |   fim
    |   |   |   fim
    |   |   fim
9   |   retorna  $(\frac{x[k]}{s} | k = 1, 2, \dots, l)$ 
    |   fim

```

Sem esta adaptação, ao computar os l primeiros momentos, irá levar um tempo de processamento l vezes maior (complexidade $O(l)$). Visto que teria que chamar o algoritmo l vezes.

Entretanto, há um limitador de desempenho no algoritmo com e sem a adaptação, pois ao sortear um nó para o qual andar em cada iteração k , é necessário conhecer as probabilidades de todos os nós, portanto, no pior caso, ele percorrerá todos os n nós, fato este que leva o algoritmo a ter uma complexidade $O(s.l.n)$ que, em um grafo grande com 1 milhão de nós e com um valor significativo de $s = 20mil$ e um $l = 10$, o número de processamentos será $20mil \times 10 \times 1milhão = 20.10^3 \times 10 \times 1.10^6 = 20.10^{10}$, o que inviabilizará o algoritmo para grafos grandes. Em resumo, o sorteio do próximo nó demanda utilizar o algoritmo de roleta, descrito na Seção 2.9, onde é descrito que sua complexidade é $O(n)$.

Uma possível solução é escolher a busca binária, tornando o algoritmo $O(\log_2 n)$. Porém, para otimizar ainda mais o desempenho, foi utilizado o algoritmo de Lipowski (Seção 2.9.1) para fazer o sorteio do próximo nó, tornando-o com complexidade constante $O(1)$.

Porém, como descrito na Seção 2.9.1, o algoritmo de Lipowski ficará lento quando houver alguns pesos com uma grande diferença em relação aos demais. O algoritmo é mais eficiente para quando os pesos estão mais uniformemente distribuídos.

4 EXPERIMENTOS

Neste capítulo, são apresentados os experimentos e análises dos resultados. Foram feitos três experimentos, o primeiro para verificar se o algoritmo devolve o resultado com uma aproximação em relação aos valores reais. O segundo para analisar experimentalmente o tempo de execução do algoritmo com instâncias geradas por três modelos teóricos de redes complexas. No terceiro experimento, foi executado o algoritmo com um grafo grande extraído do mundo real, validando seu desempenho para um grafo grande.

4.1 RECURSOS UTILIZADOS

Para a realização dos experimentos para a avaliação da eficácia e eficiência do algoritmo e validação do mesmo, foram desenvolvidos *scripts* nas linguagens de programação *Python* e *Shell script*. Os mesmos realizam chamadas ao programa, escrito em linguagem *C*, que realiza o processamento do algoritmo, tendo o grafo como entrada e os momentos e tempo de execução como saídas.

Foram desenvolvidos programas em *Python* que: geram grafos aleatórios de entrada, utilizando a biblioteca *networkx*; calcula os valores reais dos momentos, realizando multiplicações de matrizes; relatórios dos valores dos momentos, para avaliação da acurácia; relatórios tempos de processamento; e geração de gráficos de visualização dos dados, utilizando a biblioteca *matplotlib*.

Como o gerador aleatório de grafos gera apenas grafos sem pesos, foi estabelecido um critério para adicionar pesos às arestas. O mesmo foi a seleção de um valor aleatório, incluindo casas decimais, compreendido entre 1 e 6, com distribuição probabilística uniforme.

O gerador foi desenvolvido para gerar três possíveis topologias de redes complexas: redes livre de escala (BA); redes aleatórias (ER); e para redes de mundo pequeno (WS). Para cada um, foram estabelecidos alguns parâmetros. O número de nós do grafo a ser gerado é especificado na entrada do gerador.

Para o gerador de BA, foi estabelecido um parâmetro, que consiste em 5 arestas para cada novo nó conectado a um nó já existente. Para o gerador de ER, foi estabelecida uma probabilidade de 60% de haver arestas entre dois nós, sendo este valor escolhido para ajustar

melhor o equilíbrio da densidade dos grafos. Para o gerador de WS, foi estabelecido que os nós se conectassem com os 5 vizinhos mais próximos, e com 20% de probabilidade haver aresta entre dois nós; sendo estes valores ajustados para ter um melhor equilíbrio na densidade do grafo gerado.

Ao longo do desenvolvimento do algoritmo, foram avaliados também outros valores para os algoritmos de geração. Os resultados foram similares com relação ao apresentado neste experimento.

Também foi escrito um *script* na linguagem *Shell script*, utilizada com comandos do sistema operacional *Linux*, que auxilia na geração do relatório de tempo de execução.

Para a realização dos experimentos, foi utilizado um computador doméstico da marca HP, com processador Intel i5 e 8 GB de memória RAM.

4.2 AVALIAÇÃO DA ACURÁCIA DO ALGORITMO

Foram feitos experimentos com três amostras de grafos gerados por três modelos teóricos de redes complexas: uma rede livre de escalas (BA), uma rede aleatória (ER) e uma de rede de mundo pequeno (WS) (descrições dos modelos teóricos de redes na Seção 2.3).

Para a geração das amostras, foram estabelecidos um tamanho fixo de 1 mil nós e um valor de s igual a 50 mil iterações. O valor escolhido para o número de nós busca um maior valor possível, devido ao objetivo de trabalhar com grandes grafos, dentro dos limites dos recursos de hardware disponíveis para realização das gerações. O valor escolhido para s foi um ajuste realizado, baseado no experimento da seção 6, um valor com uma melhor acurácia, mas sendo menor possível devido ao objetivo de obter desempenho.

Apesar do objetivo ser trabalhar com grafos grandes, a razão pela qual foram testadas com um número de nós de um grafo que não é grande (não chegando a milhões de nós), se deve à necessidade de fazer o processamento dos cálculos dos valores reais dos momentos espectrais.

Para cada topologia, foi gerada uma amostra. Os resultados estão esquematizados nas Tabelas 2, 3 e 4, onde faz um comparativo entre os calculados pelo algoritmo e seus respectivos valores reais (calculado como $\frac{\text{Tr } M^l}{N}$, conforme descreve a Seção 2.1.5), para as amostras geradas de redes complexas de BA, ER e WS, respectivamente. Foram extraídas 10 amostras de valores calculados, com os 10 primeiros momentos espectrais. Nas tabelas, o valor calculado considerado para cada momento foi uma média aritmética entre os valores do respectivo momento das amostras geradas, executando o algoritmo. Para melhor análise da acurácia do algoritmo, foram também esquematizados o desvio padrão entre as amostras, o erro absoluto e o erro relativo (Seção 2.10).

Também foi calculado o erro médio absoluto (Seção 2.10) para cada grafo gerado. Os

Tabela 2: Comparação do algoritmo de momentos espectrais de (COHEN-STEINER et al., 2018) com os valores reais. Para uma amostra de rede complexa de BA

Momento espectral	Valor exato	Valor médio calculado	Desvio padrão	Erro absoluto	Erro relativo
1	0,000000	0,000000	0,000000	0,000000	0,000%
2	0,104516	0,104012	0,001602	0,000504	0,482%
3	0,000665	0,000570	0,000117	0,000095	14,268%
4	0,022037	0,021532	0,000750	0,000505	2,292%
5	0,000888	0,000816	0,000107	0,000072	8,123%
6	0,006248	0,006070	0,000389	0,000178	2,846%
7	0,000963	0,000892	0,000128	0,000071	7,403%
8	0,002452	0,002388	0,000223	0,000064	2,627%
9	0,000988	0,001020	0,000157	0,000032	3,234%
10	0,001428	0,001470	0,000170	0,000042	2,913%

Tabela 3: Comparação do algoritmo de momentos espectrais de (COHEN-STEINER et al., 2018) com os valores reais. Para uma amostra de rede complexa de ER

Momento espectral	Valor exato	Valor médio calculado	Desvio padrão	Erro absoluto	Erro relativo
1	0,000000	0,000000	0,000000	0,000000	0,000%
2	0,001949	0,001716	0,000177	0,000233	11,969%
3	0,000997	0,000952	0,000072	0,000045	4,526%
4	0,001002	0,001010	0,000071	0,000008	0,818%
5	0,001000	0,001010	0,000170	0,000010	1,001%
6	0,001000	0,001048	0,000089	0,000048	4,800%
7	0,001000	0,000968	0,000089	0,000032	3,200%
8	0,001000	0,001020	0,000172	0,000020	2,000%
9	0,001000	0,001048	0,000082	0,000048	4,800%
10	0,001000	0,000952	0,000146	0,000048	4,800%

Tabela 4: Comparação do algoritmo de momentos espectrais de (COHEN-STEINER et al., 2018) com os valores reais. Para uma amostra de rede complexa de WS

Momento espectral	Valor exato	Valor médio calculado	Desvio padrão	Erro absoluto	Erro relativo
1	0,000000	0,000000	0,000000	0,000000	0,000%
2	0,231245	0,231512	0,002226	0,000267	0,115%
3	0,051995	0,051594	0,001157	0,000401	0,771%
4	0,109328	0,110128	0,000876	0,000800	0,731%
5	0,050566	0,049858	0,001116	0,000708	1,399%
6	0,065887	0,066116	0,001451	0,000229	0,347%
7	0,041984	0,042040	0,000831	0,000056	0,133%
8	0,045087	0,044994	0,001320	0,000093	0,207%
9	0,033788	0,033880	0,001031	0,000092	0,272%
10	0,033226	0,032922	0,000874	0,000304	0,915%

Tabela 5: Erros médios absolutos dos grafos

Grafo	Erro médio absoluto
Barabási-Albert	0,000156
Erdős-Renyí	0,000049
Watts-Strogatz	0,000295

mesmos estão ilustrados na Tabela 5.

Analisando os resultados de assertividade do algoritmo nas Tabelas 2, 3 e 4, é possível, através das análises, fazer a validação do algoritmo.

Observando todas as tabelas desta seção, observa-se que, no pior caso, o erro absoluto (Seção 2.10) foi de aproximadamente 0,000800. É possível observar que a diferença de aproximação dos valores dos momentos está na quarta casa decimal.

4.3 TESTES DE ERROS ABSOLUTOS PARA DIFERENTES VALORES DE s , NÚMEROS DE NÓS E DE ARESTAS

Para uma análise mais profunda da complexidade e acurácia do algoritmo, foram feitos três experimentos. O objetivo é fazer um relatório dos erros médios absolutos (Seção 2.10) entre os valores dos momentos para diferentes valores de s , número de nós e número de arestas.

Para cada registro, foram executados os seguintes passos: (i) execução do algoritmo uma única vez, computando os 10 primeiros momentos espectrais; (ii) foram calculados seus respectivos valores reais; (iii) calculados os erros absolutos (Seção 2.10) para cada l -ésimo momento; (iv) calculado o erro médio absoluto.

4.3.1 Relação com o valor de s

No primeiro, foi executado o algoritmo para diferentes s em uma sequência de valores entre 10 mil e 100 mil, com um intervalo de 5 mil. Para executar, foi gerada uma amostra de uma rede complexa de WS com 500 nós. Os resultados estão esquematizados na Figura 5(a), e para maiores detalhes veja a Tabela 6 do Apêndice A.

Como é possível observar, conforme for escolhendo um s maior, a acurácia do algoritmo cresce, convergindo para um valor mais baixo de erro médio absoluto. O que comprova que a escolha de um valor de s mais alto aumenta a precisão do algoritmo.

4.3.2 Relação com o número de nós

No segundo, foram geradas aleatoriamente amostras de redes complexas de mundo pequeno (WS) para diferentes tamanhos (números de nós), com uma sequência de 100 até 1000 com um

intervalo de 50 nós. Para a execução, foi fixado um valor para a constante s igual a 10 mil. A Figura 5(b) e a Tabela 7 do Apêndice A esquematizam o experimento.

Como é observável, o algoritmo mantém a consistência, conforme o crescimento do tamanho do grafo.

4.3.3 Relação com o número de arestas

No terceiro, foram geradas aleatoriamente redes complexas aleatórias (ER), com diferentes números de arestas, em uma sequência de 1 mil até 10 mil, com 500 arestas de intervalo entre cada uma. Estes valores foram escolhidos considerando as limitações do hardware do computador utilizado para fazer os experimentos, porém ter um *dataset* maior para ter mais precisão na análise dos resultados. A topologia da rede escolhida se deve ao fato que a mesma facilita o controle do número de arestas ao gerar o grafo. O experimento está esquematizado na Figura 5(c), e para maiores detalhes veja a Tabela 8 do Apêndice A.

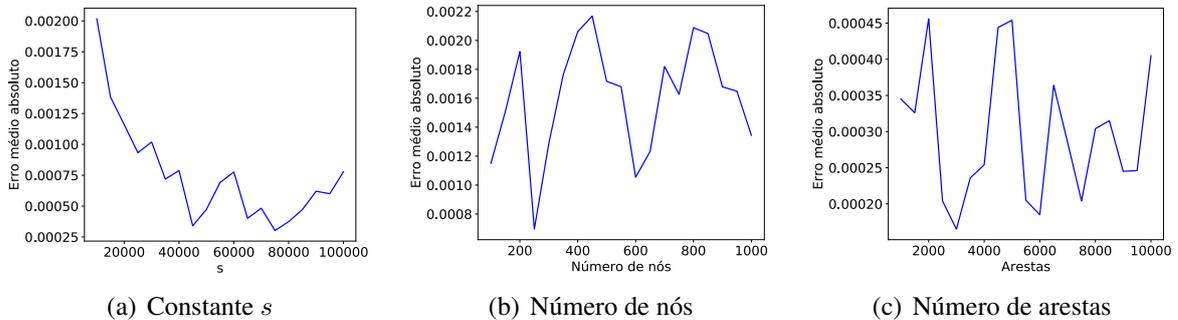


Figura 5: Relação com o erro médio absoluto

Observando os resultados, é notório que não houve crescimento nem diminuição da sua acurácia à medida que cresce do número de arestas.

4.4 TESTES DE DESEMPENHO DO ALGORITMO

Para testar o desempenho algoritmo implementado da Seção 3.1, foram feitos dois experimentos para testar o tempo de processamento, onde deseja-se verificar a sua eficiência e sua complexidade, conforme o crescimento do tamanho da entrada, sendo o número de nós no primeiro experimento, o número de arestas no segundo.

4.4.1 Relação com o número de nós

No primeiro experimento, foi executado o algoritmo para cada três conjuntos amostras de redes complexas, com diferentes topologias: redes livre de escala (BA), redes aleatórias (ER) e redes

de mundo pequeno (WS). Para cada conjunto, foi gerado um gráfico e uma tabela, relacionando o tempo de execução com o número de nós.

Para geração de grafos para as redes de BA e ER com alto número de nós, iniciando em 100 mil, a quantidade de memória RAM necessária foi insuficiente. Portanto, para a realização dos experimentos desta seção, foi necessário utilizar um servidor com 80 GB de memória RAM, e processador Xeon 2407 2,20 GHz com 4 núcleos e 4 threads.

A Seção 4.1 descreve os parâmetros para a geração aleatória das amostras de grafos. Para este experimento, houve uma alteração em um parâmetro para as redes ER. O valor da probabilidade de conectar dois nós por uma aresta foi alterado para 0,01%. Esta adaptação se deve ao fato de memória RAM insuficiente no computador para gerar uma rede com 500 mil nós desta topologia com o valor parâmetro inicialmente estabelecido.

Os registros foram gerados com uma sequência de tamanhos de grafos, começando em 100 mil e terminando em 500 mil, com um intervalo de 100 mil nós entre cada grafo. O valor escolhido para a constante s foi 10 mil.

Para cada amostra, foi executado o algoritmo 10 vezes, computando a média, o desvio padrão e o coeficiente de variação (Seção 2.10) entre as amostras.

Com o desvio padrão e o coeficiente de variação, é possível observar que houve baixa variação no tempo de processamento do algoritmo, avaliando dispersão entre os valores das amostras.

Os resultados para as topologias BA, ER e WS foram respectivamente ilustrados nas Figuras 6(a), (b) e (c). Maiores detalhes estão disponíveis nas Tabelas 9, 10 e 11 do Apêndice A. Nas figuras, estão esquematizados os tempos de execução com as respectivas barras de erro (do desvio padrão). Nas tabelas, estão esquematizados o tempo médio de execução, o desvio padrão e o coeficiente de variação (Seção 2.10) para cada número de nós.

Através da análise do experimento feito para avaliar o tempo de processamento e complexidade do algoritmo, é possível avaliar a eficiência e o desempenho do mesmo. Porém, na rede de WS, o tempo de execução se manteve constante, na rede de BA e na de ER ocorreu um aumento no tempo. Assim, o tempo de execução do algoritmo irá depender também da topologia do grafo.

Observando no experimento que, no pior caso, o algoritmo leva 1353 milissegundos para processar um grafo com 500 mil nós, é possível comprovar sua eficiência para grafos grandes.

4.4.2 Relação com o número de arestas

Para comprovação da consistência da eficiência do algoritmo conforme a variação do número de arestas do grafo, foi realizado um experimento que relaciona o tempo de execução com o número de arestas das respectivas amostras.

Para cada número de arestas, foi gerada aleatoriamente uma amostra de rede complexa com topologia aleatória (ER). As mesmas foram geradas em uma sequência de número de arestas de 1 mil até 10 mil, com um intervalo de 500. Foi fixado um tamanho de 500 nós para cada grafo gerado. Para cada amostra, foi executado o algoritmo, capturando seu tempo de execução.

O resultado do experimento está esquematizado na Figura 6(d) e na Tabela 12 do Apêndice A.

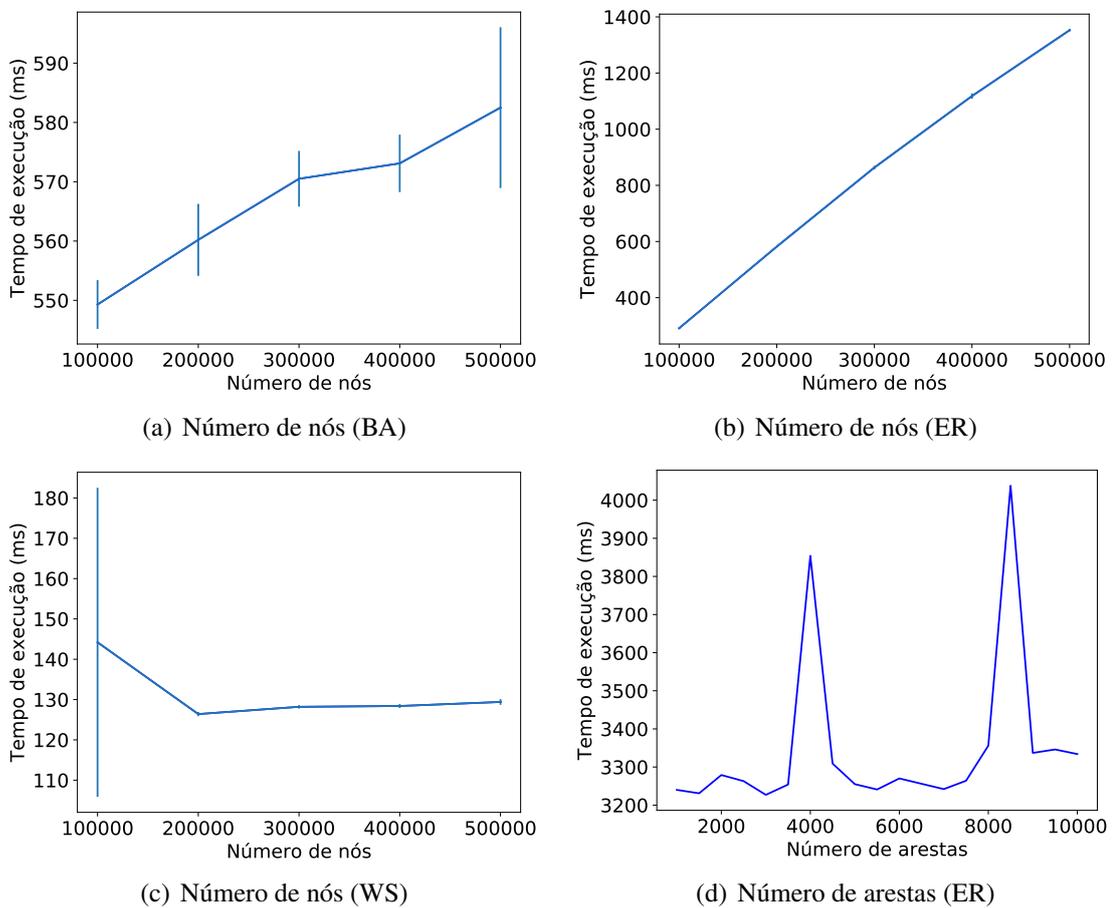


Figura 6: Relação com o tempo de execução

Como é observável nos resultados, o tempo de execução manteve-se sem crescimento e decrescimento à medida que aumenta o número de arestas do grafo.

Alguns picos no tempo de execução podem ser observados na Figura 6(d). Este comportamento, como descreve a Seção 3.1 deve-se à limitação do algoritmo de Lipowski, que se

torna lento quando há uma baixa uniformidade na distribuição das probabilidades de ir de um nó para outro nas amostras geradas cujo tempo de execução foi mais alto do que a média.

4.4.3 Executando o algoritmo em uma rede real grande

Para uma nova validação do algoritmo, foi testado executando para um grafo grande extraída do mundo real, calculando os 10 primeiros momentos espectrais.

O grafo, extraído de (BURNS et al., 2013), é uma amostra de **connectome**, um modelo de grafo que representa um mapa de conexões entre neurônios em uma rede neural de um cérebro (BURNS et al., 2013). A amostra escolhida é um grafo não dirigido com pesos, com 841097 nós e 48522314 arestas.

O algoritmo foi executado em 2313 milissegundos, usando um computador da marca HP, com processador Intel i5 e 8 GB de memória RAM.

Analisando este experimento, é observável que para um grafo com mais de 800 mil nós, quase 1 milhão, o algoritmo foi executado em um tempo suficiente para que um computador comum consiga realizar, em menos de 3 segundos. O computador no qual foi executado possui configurações de hardware de um computador doméstico comum.

5 CONCLUSÕES

Observando as validações da eficácia e eficiência nos resultados da Seção 4, este trabalho oportuniza futuros trabalhos e pesquisadores utilizar o algoritmo para desenvolver soluções para aplicações do mundo real e novas pesquisas. Trabalhos futuros também poderão desenvolver melhorias e adaptações deste algoritmo.

Esta dissertação trabalha com um algoritmo para calcular o espectro do grafo, onde os momentos espectrais é apenas a primeira etapa do mesmo. Trabalhos futuros poderão desenvolver a segunda etapa do algoritmo, que calcula o espectro final. Para implementar o mesmo, é necessário calcular a distribuição probabilística a partir dos momentos. Há uma longa discussão dos pesquisadores matemáticos para este problema, pois não é uma tarefa trivial. Um dos motivos é que podem existir múltiplas distribuições para um mesmo arranjo momentos.

A TABELAS DOS RESULTADOS DOS EXPERIMENTOS

Tabela 6: Relação entre s e erro médio absoluto

Constante s	Erro médio absoluto
10 mil	0.002017
15 mil	0.001383
20 mil	0.001160
25 mil	0.000932
30 mil	0.001019
35 mil	0.000719
40 mil	0.000788
45 mil	0.000340
50 mil	0.000473
55 mil	0.000692
60 mil	0.000776
65 mil	0.000401
70 mil	0.000483
75 mil	0.000302
80 mil	0.000374
85 mil	0.000472
90 mil	0.000620
95 mil	0.000600
100 mil	0.000779

Tabela 7: Relação entre o número de nós e o erro médio absoluto

Número de nós	Erro médio absoluto
100	0.001151
150	0.001510
200	0.001923
250	0.000696
300	0.001287
350	0.001763
400	0.002060
450	0.002168
500	0.001717
550	0.001680
600	0.001054
650	0.001232
700	0.001820
750	0.001627
800	0.002088
850	0.002047
900	0.001679
950	0.001649
1000	0.001344

Tabela 8: Relação entre o número de arestas e o erro médio absoluto

Número de arestas	Erro médio absoluto
1000	0.000345
1500	0.000326
2000	0.000456
2500	0.000204
3000	0.000165
3500	0.000236
4000	0.000254
4500	0.000444
5000	0.000454
5500	0.000205
6000	0.000185
6500	0.000364
7000	0.000285
7500	0.000204
8000	0.000304
8500	0.000315
9000	0.000245
9500	0.000246
10000	0.000405

Tabela 9: Tempo de execução do algoritmo (BA)

Tamanho do Grafo	Tempo médio de execução	Desvio padrão	Coefficiente de variação
100 mil	549,300000	4,110961	0,748%
200 mil	560,200000	6,069962	1,084%
300 mil	570,500000	4,696334	0,823%
400 mil	573,100000	4,840799	0,845%
500 mil	582,500000	13,566708	2,329%

Tabela 10: Tempo de execução do algoritmo (ER)

Tamanho do Grafo	Tempo médio de execução	Desvio padrão	Coefficiente de variação
100 mil	291,000000	2,538591	0,872%
200 mil	581,700000	1,828782	0,314%
300 mil	863,300000	6,037844	0,699%
400 mil	1118,000000	9,580072	0,857%
500 mil	1352,600000	3,893014	0,288%

Tabela 11: Tempo de execução do algoritmo (WS)

Tamanho do Grafo	Tempo médio de execução	Desvio padrão	Coefficiente de variação
100 mil	144,200000	38,342897	26,590%
200 mil	126,400000	0,516398	0,409%
300 mil	128,200000	0,421637	0,329%
400 mil	128,400000	0,516398	0,402%
500 mil	129,400000	0,699206	0,540%

Tabela 12: Relação entre o número de arestas e o tempo de execução

Número de arestas	Tempo de execução (milissegundos)
1000	3240
1500	3231
2000	3279
2500	3263
3000	3227
3500	3254
4000	3854
4500	3309
5000	3255
5500	3241
6000	3270
6500	3256
7000	3242
7500	3264
8000	3356
8500	4038
9000	3337
9500	3346
10000	3334

REFERÊNCIAS

- AGARWAL, B. *Basic Statistics*. New Age International, 2006. ISBN 9788122418149. Disponível em: <<https://books.google.com.br/books?id=qT2srNLLxB0C>>.
- AMERICO, H. V. Análise numérica para busca de autovalores. Universidade Federal de Minas Gerais, 2013.
- ANTONIOU, I.; TSOMPA, E. Statistical analysis of weighted networks. *Discrete Dynamics in Nature and Society*, v. 2008, 02 2008.
- BALASUBRAMANIAN, K. Spectral moments and walks for large carbon cage clusters. *Chemical Physics Letters*, v. 175, n. 3, p. 273 – 278, 1990. ISSN 0009-2614. Disponível em: <<http://www.sciencedirect.com/science/article/pii/000926149085555Q>>.
- BANERJEE, A. *The spectrum of the graph Laplacian as a tool for analyzing structure and evolution of networks*. Tese (Doutorado) — Verlag nicht ermittelbar, 2008.
- BARABÁSI, A.-L.; ALBERT, R. Emergence of scaling in random networks. *science*, American Association for the Advancement of Science, v. 286, n. 5439, p. 509–512, 1999.
- BARABÁSI, A.-L.; BONABEAU, E. *Scale-Free Networks*. [S.l.: s.n.], 2003.
- Barabási, A.-L.; Bonabeau, E. Scale-free networks. *Scientific American*, v. 288, n. 5, p. 60–69, 2003.
- BETZEL, R. F. et al. Generative models of the human connectome. *Neuroimage*, Elsevier, v. 124, p. 1054–1064, 2016.
- Bezdek, J. C.; Tsao, E. C. ; Pal, N. R. Fuzzy kohonen clustering networks. In: *[1992 Proceedings] IEEE International Conference on Fuzzy Systems*. [S.l.: s.n.], 1992. p. 1035–1043. ISSN null.
- Bonner, S. et al. Deep topology classification: A new approach for massive graph classification. In: *2016 IEEE International Conference on Big Data (Big Data)*. [S.l.: s.n.], 2016. p. 3290–3297.
- Bottou, L.; Bengio, Y.; Le Cun, Y. Global training of document processing systems using graph transformer networks. In: *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 1997. p. 489–494.
- BURNS, R. et al. The open connectome project data cluster: Scalable analysis and vision for high-throughput neuroscience. In: *Proceedings of the 25th International Conference on Scientific and Statistical Database Management*. New York, NY, USA: Association for Computing Machinery, 2013. (SSDBM). ISBN 9781450319218. Disponível em: <<https://doi.org/10.1145/2484838.2484870>>.

CHUNG, F. *Spectral Graph Theory*. Conference Board of the Mathematical Sciences, 1992. (CBMS Regional Conference Series, N° 92). ISBN 9780821889367. Disponível em: <https://books.google.com.br/books?id=YUc38_MCuhAC>.

COHEN-STEINER, D. et al. Approximating the spectrum of a graph. In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. New York, NY, USA: ACM, 2018. (KDD '18), p. 1263–1271. ISBN 978-1-4503-5552-0. Disponível em: <<http://doi.acm.org/10.1145/3219819.3220119>>.

COSTA, L. da F. et al. Characterization of complex networks: A survey of measurements. *Advances in Physics*, Taylor Francis, v. 56, n. 1, p. 167–242, 2007. Disponível em: <<https://doi.org/10.1080/00018730601170527>>.

CULLEN, C. *Matrices and Linear Transformations: Second Edition*. Dover Publications, 2012. (Dover Books on Mathematics). ISBN 9780486132419. Disponível em: <<https://books.google.com.br/books?id=czDdnZPBFHcC>>.

CVETKOVIC, D. M.; GUTMAN, I. *Applications of graph spectra*. [S.l.]: Matematički institut SANU, 2009.

CVETKOVIC, D.; SIMIĆ, S. *Graph spectra in computer science*. 2010.

DENG, J. et al. Large-scale object classification using label relation graphs. In: SPRINGER. *European conference on computer vision*. [S.l.], 2014. p. 48–64.

DESAI, M.; RAO, V. A characterization of the smallest eigenvalue of a graph. *Journal of Graph Theory*, Wiley Online Library, v. 18, n. 2, p. 181–194, 1994.

Dias de Oliveira, G.; Kashiwabara, A. Y. A probabilistic algorithm to estimate the spectral moments of large undirected weighted graphs. In: *2019 8th Brazilian Conference on Intelligent Systems (BRACIS)*. [S.l.: s.n.], 2019. p. 783–787.

DORIGO, M.; GAMBARDELLA, L. M. Ant colonies for the travelling salesman problem. *Biosystems*, v. 43, n. 2, p. 73 – 81, 1997. ISSN 0303-2647. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0303264797017085>>.

DOROGOVTSEV, S. N. et al. Spectra of complex networks. *Phys. Rev. E*, American Physical Society, v. 68, p. 046109, Oct 2003. Disponível em: <<https://link.aps.org/doi/10.1103/PhysRevE.68.046109>>.

ERDŐS, P.; RÉNYI, A. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci.*, v. 5, n. 1, p. 17–60, 1960.

ERMAN, J.; ARLITT, M.; MAHANTI, A. Traffic classification using clustering algorithms. In: *Proceedings of the 2006 SIGCOMM Workshop on Mining Network Data*. New York, NY, USA: Association for Computing Machinery, 2006. (MineNet '06), p. 281–286. ISBN 159593569X. Disponível em: <<https://doi.org/10.1145/1162678.1162679>>.

ESTRADA, E. Spectral moments of the edge adjacency matrix in molecular graphs. 1. definition and applications to the prediction of physical properties of alkanes. *Journal of chemical information and computer sciences*, ACS Publications, v. 36, n. 4, p. 844–849, 1996.

- ESTRADA, E. Spectral moments of the edge adjacency matrix in molecular graphs. 3. molecules containing cycles. *Journal of chemical information and computer sciences*, ACS Publications, v. 38, n. 1, p. 23–27, 1998.
- FIEBERG, J.; ELLNER, S. P. Stochastic matrix models for conservation and management: a comparative review of methods. *Ecology letters*, Wiley Online Library, v. 4, n. 3, p. 244–266, 2001.
- FLAMMER, G. H.; KALKWARF, R. L. *Method for routing packets in a packet communication network*. [S.l.]: Google Patents, jul. 3 1990. US Patent 4,939,726.
- FOGGIA, P. et al. A graph-based clustering method and its applications. In: . [S.l.: s.n.], 2007. p. 277–287.
- GASTNER, M. T.; ÓDOR, G. The topology of large open connectome networks for the human brain. *Scientific reports*, Nature Publishing Group, v. 6, p. 27249, 2016.
- Gkantsidis, C.; Mihail, M.; Zegura, E. Spectral analysis of internet topologies. In: *IEEE INFO-COM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No.03CH37428)*. [S.l.: s.n.], 2003. v. 1, p. 364–374 vol.1.
- GOLDENBERG, D. E. *Genetic algorithms in search, optimization and machine learning*. [S.l.]: Addison Wesley, Reading: MA, 1989.
- Heinzelman, W. B.; Chandrakasan, A. P.; Balakrishnan, H. An application-specific protocol architecture for wireless microsensor networks. *IEEE Transactions on Wireless Communications*, v. 1, n. 4, p. 660–670, Oct 2002. ISSN 1558-2248.
- HOLME, P. Core-periphery organization of complex networks. *Phys. Rev. E*, American Physical Society, v. 72, p. 046111, Oct 2005. Disponível em: <<https://link.aps.org/doi/10.1103/PhysRevE.72.046111>>.
- JOHNSON, C.; SAIAGO, C. *Eigenvalues, Multiplicities and Graphs*. Cambridge University Press, 2018. (Cambridge Tracts in Mathematics). ISBN 9781107095458. Disponível em: <<https://books.google.com.br/books?id=wDhFDwAAQBAJ>>.
- JUNGNICKEL, D. *Graphs, Networks and Algorithms*. [S.l.: s.n.], 2008.
- KILGOUR, D. M.; HIPEL, K. W. The graph model for conflict resolution: past, present, and future. *Group Decision and Negotiation*, Springer, v. 14, n. 6, p. 441–460, 2005.
- KOOPMANS, L. H.; OWEN, D. B.; ROSENBLATT, J. I. Confidence intervals for the coefficient of variation for the normal and log normal distributions. *Biometrika*, v. 51, n. 1-2, p. 25–32, 06 1964. ISSN 0006-3444. Disponível em: <<https://doi.org/10.1093/biomet/51.1-2.25>>.
- KRESSNER, D.; SCHRÖDER, C.; WATKINS, D. S. Implicit qr algorithms for palindromic and even eigenvalue problems. *Numerical Algorithms*, Springer, v. 51, n. 2, p. 209–238, 2009.
- LIPOWSKI, A.; LIPOWSKA, D. Roulette-wheel selection via stochastic acceptance. *Physica A: Statistical Mechanics and its Applications*, Elsevier, v. 391, n. 6, p. 2193–2196, 2012.
- Liu, Q.; Dong, Z.; Wang, E. Moment-based spectral analysis of large-scale generalized random graphs. *IEEE Access*, v. 5, p. 9453–9463, 2017.

- LURIE, J. *Review of Spectral Graph Theory: By Fan R. K. Chung*. New York, NY, USA: ACM, 1999. 14–16 p. ISSN 0163-5700. Disponível em: <<http://doi.acm.org/10.1145/568547.568553>>.
- MALIK, G. *Calculus of Finite Difference & Numerical Analysis*. Krishna Prakashan Media, 2003. (214-41). ISBN 9788182833319. Disponível em: <<https://books.google.com.br/books?id=l6t4BgAAQBAJ>>.
- MENEZES, R. et al. *Complex Networks: Results of the 1st International Workshop on Complex Networks (CompleNet 2009)*. Springer Berlin Heidelberg, 2009. (Studies in Computational Intelligence). ISBN 9783642012051. Disponível em: <<https://books.google.com.br/books?id=XhvLxguc6uoC>>.
- MESBAHI, M.; EGERSTEDT, M. *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press, 2010. (Princeton Series in Applied Mathematics). ISBN 9781400835355. Disponível em: <<https://books.google.com.br/books?id=GlrqRwumdDcC>>.
- MIEGHEM, P. V. *Graph spectra for complex networks*. [S.l.]: Cambridge University Press, 2010.
- NEWMAN, M. E. J. *Properties of highly clustered networks*. 2003.
- NOH, J. D.; RIEGER, H. *Random walks on complex networks*. 2004.
- PAIGE, C. C. *The computation of eigenvalues and eigenvectors of very large sparse matrices*. Tese (Doutorado) — University of London, 1971.
- PAPADOPOULOS, S. et al. Community detection in social media. *Data Min. Knowl. Discov.*, v. 24, p. 515–554, 05 2012.
- PRECIADO, V. M.; JADBABAIE, A. Moment-based spectral analysis of large-scale networks using local structural information. *IEEE/ACM Transactions on Networking*, IEEE, v. 21, n. 2, p. 373–382, 2012.
- Preciado, V. M. et al. Distributed control of the laplacian spectral moments of a network. In: *Proceedings of the 2010 American Control Conference*. [S.l.: s.n.], 2010. p. 4462–4467.
- ROWLINSON, P. The main eigenvalues of a graph: a survey. *Applicable Analysis and Discrete Mathematics*, JSTOR, p. 455–471, 2007.
- SEIER, E.; BONETT, D. Two families of kurtosis measures. *Metrika*, Springer, v. 58, n. 1, p. 59–70, 2003.
- Shokoufandeh, A. et al. Indexing hierarchical structures using graph spectra. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 27, n. 7, p. 1125–1140, 2005.
- SILVA, T. C.; ZHAO, L. *Machine learning in complex networks*. [S.l.]: Springer, 2016.
- SIMONCINI, V.; ELDÉN, L. Inexact rayleigh quotient-type methods for eigenvalue computations. *BIT Numerical Mathematics*, Springer, v. 42, n. 1, p. 159–182, 2002.
- SPIELMAN, D. A. Spectral graph theory and its applications. In: *IEEE. 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*. [S.l.], 2007. p. 29–38.

- STROGATZ, S. H. Exploring complex networks. *Nature*, v. 410, n. 6825, p. 268–276, Mar 2001. ISSN 1476-4687. Disponível em: <<https://doi.org/10.1038/35065725>>.
- TAKAHASHI, D. Y. et al. Discriminating different classes of biological networks by analyzing the graphs spectra distribution. *PLOS ONE*, Public Library of Science, v. 7, n. 12, p. 1–12, 12 2012. Disponível em: <<https://doi.org/10.1371/journal.pone.0049949>>.
- TARAPATA, Z. Modelling and analysis of transportation networks using complex networks: Poland case study. *Archives of Transport*, v. 36, p. 55–65, 01 2015.
- THULASIRAMAN, K.; SWAMY, M. N. S. *Graphs: Theory and Algorithms*. [S.l.: s.n.], 1992.
- TISSEUR, F.; DONGARRA, J. A parallel divide and conquer algorithm for the symmetric eigenvalue problem on distributed memory architectures. *SIAM Journal on Scientific Computing*, SIAM, v. 20, n. 6, p. 2223–2236, 1999.
- TOLEDO, K. *Software compares more than two complex networks simultaneously — AGÊNCIA FAPESP*. 9 2019. <http://agencia.fapesp.br/software-compares-more-than-two-complex-networks-simultaneously/31482/>. Acesso em 11 Mar. 2020.
- VANMARCKE, E. H. Properties of spectral moments with applications to random vibration. *Journal of the Engineering Mechanics Division*, ASCE, v. 98, n. 2, p. 425–446, 1972.
- VERSLUIS, C.; STRAETMANS, S. Skewness measures for the weibull distribution. *Available at SSRN 2590356*, 2015.
- VLASE, S.; MARIN, M.; ÖCHSNER, A. *Eigenvalue and Eigenvector Problems in Applied Mechanics*. Springer International Publishing, 2018. (Advanced Structured Materials). ISBN 9783030009915. Disponível em: <<https://books.google.com.br/books?id=0rx1DwAAQBAJ>>.
- WANG, X. F.; CHEN, G. *Complex networks: small-world, scale-free and beyond*. 2003.
- WATTS, D. J.; STROGATZ, S. H. Collective dynamics of ‘small-world’ networks. *nature*, Nature Publishing Group, v. 393, n. 6684, p. 440–442, 1998.
- WEST, D. B. et al. *Introduction to graph theory*. [S.l.]: Prentice hall Upper Saddle River, 2001.
- WILKINSON, J. H. *The algebraic eigenvalue problem*. [S.l.]: Oxford Clarendon, 1965.
- WILLMOTT, C. J.; MATSUURA, K. Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. *Climate research*, v. 30, n. 1, p. 79–82, 2005.
- YUAN, X.-T.; ZHANG, T. Truncated power method for sparse eigenvalue problems. *Journal of Machine Learning Research*, v. 14, n. Apr, p. 899–925, 2013.
- ZAVLANOS, M. M.; PRECIADO, V. M.; JADBABAIE, A. Spectral control of mobile robot networks. In: IEEE. *Proceedings of the 2011 American Control Conference*. [S.l.], 2011. p. 3245–3250.
- ZHOU, B. et al. On spectral moments and energy of graphs. *MATCH Commun. Math. Comput. Chem*, v. 57, p. 183–191, 2007.