UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Câmpus de São José do Rio Preto

JOSÉ CARLOS DE FREITAS

Otimização em algoritmos de extração de bibliométricas de redes de colaboração científica

São José do Rio Preto
2020

JOSÉ CARLOS DE FREITAS

Otimização em algoritmos de extração de bibliométricas de redes de colaboração científica

Dissertação apresentada como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação, junto ao Programa de Pós-Graduação em Ciência da Computação, do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista "Júlio de Mesquita Filho", Câmpus de São José do Rio Preto.

Financiadora: CAPES

Prof. Dr. Carlos Roberto Valêncio
Orientador

São José do Rio Preto
2020

F866o

Freitas, José Carlos de
Otimização em algoritmos de extração de bibliométricas de redes de colaboração científica / José Carlos de Freitas. -- São José do Rio Preto, 2020
76 p. : il., tabs.

Dissertação (mestrado) - Universidade Estadual Paulista (Unesp), Instituto de Biociências Letras e Ciências Exatas, São José do Rio Preto
Orientador: Carlos Roberto Valêncio

1. Bibliometria. 2. Big Data. 3. Banco de dados não-relacionais. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca do Instituto de Biociências Letras e Ciências Exatas, São José do Rio Preto. Dados fornecidos pelo autor(a).

Essa ficha não pode ser modificada.

JOSÉ CARLOS DE FREITAS

Otimização em algoritmos de extração de bibliométricas de redes de colaboração científica

Dissertação apresentada como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação, junto ao Programa de Pós-Graduação em Ciência da Computação, do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista "Júlio de Mesquita Filho", Câmpus de São José do Rio Preto.

Financiadora: CAPES

Comissão Examinadora

Prof. Dr. Carlos Roberto Valêncio
UNESP – São José do Rio Preto
Orientador

Prof. Dr. Geraldo Francisco Donega Zafalon
UNESP – São José do Rio Preto

Prof. Dr. Angelo Cesar Colombini
Universidade Federal Fluminense – Niterói - RJ

São José do Rio Preto
28 de agosto de 2020

Ao meu Pai, minha Mãe, meu irmão e minha irmã, que sempre me deram suporte emocional e financeiro.

Agradecimentos

Agradeço primeiramente à minha família, pelo suporte emocional e financeiro. Aos meus colegas de faculdade, que são agora amigos para a vida, aos colegas de Grupo de Banco de Dados (GBD) que me ajudaram a evoluir tecnicamente e, em especial, ao colega de grupo William Tenório e ao professor Carlos Roberto Valêncio, por me orientarem e me ajudarem a evoluir não apenas meus trabalhos acadêmicos, mas também como pessoa.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

"*O sucesso é ir de fracasso em fracasso sem perder entusiasmo*"

Winston Churchill (1959, p. 109)

Resumo

Com o processo natural de evolução da ciência, a produção de informações nesta área passou a ganhar dimensões importantes, como é o caso quanto as produções científicas e as respectivas colaborações dos pesquisadores. Isso deu origem a grandes redes de colaboração científica, as quais podem ser extraídas de plataformas de armazenamento de informações acadêmicas. Nesse contexto, tem-se a bibliometria com o objetivo de extrair conhecimento quantitativo dessas redes de colaboração científica por meio de métricas, denominadas bibliométricas. Entretanto, os algoritmos de extração de bibliométricas não são escaláveis e, portanto, não suportam grandes redes de colaboração. Neste sentido, faz-se necessário o desenvolvimento de algoritmos otimizados por meio de distribuição de dados, que utilizam os recursos de forma mais eficiente. Assim, a contribuição científica desse trabalho é a proposição de algoritmos de extração de bibliométricas com desempenho superior aos semelhantes encontrados na literatura para grandes redes de colaboração. Verificou-se por meio dos testes que o algoritmo de extração da bibliométrica de transitividade desenvolvido tem crescimento de tempo de processamento 12,76 vezes menor que o tempo de processamento do algoritmo paralelo proposto na literatura, quando o número de pesquisadores tende ao infinito. Como subproduto, foi desenvolvida uma Ferramenta de Extração de Indicadores Bibliométricos com o objetivo de facilitar o uso dos algoritmos desenvolvidos para extração de conhecimento de redes de colaboração científica.

Palavras-chave: Bibliometria. Big Data. Data Mining. NoSQL. Paralelização e Distribuição de Algoritmos. Rede de Colaboração Científica.

Abstract

Due to the natural process of science evolution, information production in this area has been reaching important dimensions, such as scientific productions and their respective collaborations of researchers. As a result, large scientific collaboration networks have arisen, which can be extracted from academic information storage platforms. In this context, Bibliometry aims at extracting quantitative knowledge from these scientific collaboration networks through metrics, called bibliometrics. However, the extraction algorithms of bibliometrics are not scalable and, consequently, do not support large collaboration networks. Considering this, the development of an optimized algorithm becomes necessary through data distribution that uses resources more efficiently. Therefore, the scientific contribution of this work is to implement the transitivity algorithm for extracting bibliometrics developed through the Apache Spark framework with superior performance to those found in the literature for large collaboration networks. Tests have revealed that the developed algorithm has a processing time growth 12.76 times smaller than the processing time of the parallel algorithm proposed in the literature, where the number of researchers tends to infinity. As a by-product, a Tool for Extracting Bibliometric Indicators was designed to enable the use of algorithms developed to extract knowledge from scientific collaboration networks.

Keywords: Bibliometric. Big Data. Data Mining. NoSQL. Algorithm Parallelization and Distribution. Co-authorship Network.

Lista de Ilustrações

Figura 1 – rede de colaboração exemplificada 18
Figura 2 – exibição comparativa de duas redes hipotéticas, a de cor azul e a outra, de cor laranja 47
Figura 3 – rede de colaboração criada para o teste de mesa 51
Figura 4 – rede de colaboração para teste de corretude de comparação de resultados 51
Figura 5 – teste de desempenho do algoritmo de Transitividade 57
Figura 6 – teste de desempenho do algoritmo de número total de trabalhos em coautoria 59
Figura 7 – teste de desempenho do algoritmo de número total de coautorias 60
Figura 8 – teste de desempenho do algoritmo de número total de coautores 61
Figura 9 – teste de desempenho do algoritmo de assortatividade 62
Figura 10 – teste de escalabilidade do algoritmo de Transitividade 64
Figura 11 – gráfico trienal de total de pesquisadores 74
Figura 12 – gráfico trienal de total de trabalhos em coautoria 74
Figura 13 – gráfico trienal do total de coautorias 74
Figura 14 – gráfico trienal do total de coautores 74
Figura 15 – gráfico trienal da média de trabalhos em coautoria 75
Figura 16 – gráfico trienal da densidade das coautorias 75
Figura 17 – gráfico trienal da Assortatividade 75
Figura 18 – gráfico trienal da Transitividade 75
Figura 19 – gráfico trienal da distância média entre os pesquisadores 76
Figura 20 – gráfico trienal do Diâmetro da rede de colaboração 76
Figura 21 – gráfico trienal do tamanho do maior grupo de coautores 76
Figura 22 – gráfico trienal da porcentagem do maior grupo de coautores 76

Lista de Tabelas

Tabela 1 – conhecimento extraído por cada bibliométricas 23
Tabela 2 – comparação entre os trabalhos correlatos 34
Tabela 3 – estrutura do trabalho 36
Tabela 4 – propriedades utilizadas para implementação de cada bibliométrica 37
Tabela 5 – valores extraídos no teste de mesa 52
Tabela 6 – valores extraídos nas duas ferramentas 52
Tabela 7 – bibliométricas da rede de colaboração da UNESP 54
Tabela 8 – bibliométricas trienais da rede de colaboração da UNESP 55
Tabela 9 – testes de desempenho do algoritmo de transitividade 56
Tabela 10 – teste de desempenho do algoritmo de Transitividade 57
Tabela 11 – teste de desempenho do algoritmo de número total de trabalhos em coautoria 58
Tabela 12 – teste de desempenho do algoritmo de número total de coautorias 59
Tabela 13 – teste de desempenho do algoritmo de número total de coautores 60
Tabela 14 – teste de desempenho do algoritmo de assortatividade 61
Tabela 15 – teste de desempenho do algoritmo de Caminho médio e Diâmetro 63
Tabela 16 – teste de escalabilidade do algoritmo de Transitividade 63
Tabela 17 – comparação entre os trabalhos correlatos e esse trabalho. 67

Lista de Abreviaturas e Siglas

ACID – Atomicidade, Consistência, Isolamento e Durabilidade

API – Application Programming Interface

APSP – All-Pairs Shortest-Paths

BFS – Breadth-first search

BRICS – Brazil, Russia, India, China and South Africa

CSS – Cascading Style Sheets

CSV – Comma-separated values

FEIB – Ferramenta de Extração de Indicadores Bibliométricos

HDFS – Hadoop Distributed File System

HTML – HyperText Markup Language

JSON – JavaScript Object Notation

NoSQL – Not Only Structured Query Language

RDD – Resilient Distributed Datasets

SGBD – Sistema Gerenciadores de Banco de Dados

SIMD – Single Instruction Multiple Data

UNESP – Universidade Estadual Paulista “Júlio de Mesquita Filho”

WoS – Web of Science

XML – Extensible Markup Language

YARN – Yet Another Resource Negotiator

Sumário

1. Introdução 13
1.1. Motivação e Escopo 14
1.2. Objetivos 14
1.3. Contribuições científicas 15
1.4. Organização da monografia 15
2. Fundamentação Teórica 16
2.1. Cienciometria e Bibliometria 16
2.2. Bibliométricas 17
2.2.1. Número total de pesquisadores 18
2.2.2. Número total de coautorias 18
2.2.3. Número total de trabalhos em coautorias 19
2.2.4. Número total de coautores 19
2.2.5. Média de trabalhos em coautoria 19
2.2.6. Densidade das coautorias 20
2.2.7. Transitividade 20
2.2.8. Assortatividade 21
2.2.9. Distância média entre os pesquisadores 21
2.2.10. Diâmetro 22
2.2.11. Tamanho do maior grupo de coautores 22
2.2.12. Porcentagem do maior grupo de coautores 22
2.2.13. Considerações finais 23
2.3. Paralelização e Distribuição de algoritmos 23
2.3.1. MapReduce 25
2.3.2. Apache Hadoop 26
2.3.3. Apache Spark 26
2.3.4. Apache GraphX 27
2.3.5. Pregel API 28
2.4. NoSQL 28
2.5. Fundamentos em Bibliometria e Processamento Paralelo 29
2.5.1. Estudos bibliométricos de redes de colaboração científica 29
2.5.2. Ferramentas e algoritmos para análise de redes de colaboração científica 32
2.5.3. Comparação entre os trabalhos correlatos 33
2.6. Considerações finais 34
3. Desenvolvimento do projeto 35
3.1. Material e métodos 35
3.2. Distribuição das bibliométricas 36

3.2.1. Bibliométricas com baixa complexidade de implementação 37
3.2.2. Assortatividade 39
3.2.3. Transitividade 41
3.2.4. Caminho médio e diâmetro 43
3.3. Ferramenta de Extração de Indicadores Bibliométricos (FEIB) 44
3.3.1. Importador 45
3.3.2. Banco de dados orientado a grafos 46
3.3.3. Extração de bibliométricas 46
3.3.4. Exibição das bibliométricas 46
3.4. Considerações finais 48
4. Avaliação Experimental 49
4.1. Configurações da plataforma de testes 49
4.2. Teste de corretude 50
4.2.1. Teste de mesa 51
4.2.2. Comparação de resultados 52
4.3. Caso de uso 53
4.3.1. Unesp geral 53
4.3.2. Unesp trienal 54
4.4. Teste de desempenho 55
4.4.1. Transitividade 56
4.4.2. Número total de trabalhos em coautoria, número total de coautorias, número total de coautores e assortatividade 58
4.4.3. Caminho médio e diâmetro 62
4.5. Teste de escalabilidade 63
4.6. Considerações finais 64
5. Conclusão 66
5.1. Contribuições científicas 67
5.2. Trabalhos futuros 67
Referências 69
Apêndice A – gráficos comparativos das bibliométricas extraídas no estudo trienal 74

1. Introdução

O desenvolvimento de pesquisa científica pode ser realizado de forma individual ou colaborativa. A forma colaborativa apresenta vantagens em relação à individual, tais como: facilidade no compartilhamento de ideias, nos resultados das pesquisas, na dimensão do conhecimento, nos equipamentos científicos e recursos disponíveis, expansão da visibilidade dos artigos e criação de conexões entre as academias (MENA-CHALCO et al., 2014; VALENCIO et al., 2017; VALENCIO et al., 2020).

Essas vantagens, atreladas à evolução da ciência, dos meios de comunicação e dos meios de transporte, contribuem para o aumento no desenvolvimento de trabalhos científicos desenvolvidos em coautoria, o que, consequentemente, contribui para a criação de complexas redes de colaboração científica, formadas por pesquisadores e suas colaborações (DA SILVA; BARBOSA; DUARTE, 2012).

Essas redes de colaboração científica podem ser extraídas de plataformas que armazenam informações científicas (VALENCIO et al., 2017; VALENCIO et al., 2020), como a Plataforma Lattes (CNPQ, 2019) e a *The Digital Bibliography and Library Project* (DBLP, 2019). Por meio do seu estudo, é possível caracterizar e verificar a evolução da ciência ao longo do tempo, representar os interesses em comum dos pesquisadores e fornecer uma ampla visão sobre o relacionamento existente entre eles (MENA-CHALCO et al., 2014; VALENCIO et al., 2017; VALENCIO et al., 2020). Dessa forma, o estudo das redes de colaborações científicas pode se tornar uma ótima ferramenta para monitorar a evolução das pesquisas científicas (MOED, 1985).

Assim, motivado pela importância e complexidade da extração de conhecimento das redes de colaboração científica e com objetivo de possibilitar seu estudo quantitativo, foram

desenvolvidas métricas baseadas na teoria dos grafos, denominadas bibliométricas (MENA-CHALCO et al., 2014; VALENCIO et al., 2017; VALENCIO et al., 2020). Essas bibliométricas foram fundamentadas pela Cienciometria e a Bibliometria, que são os estudos dos aspectos quantitativos da ciência e das produções científicas, respectivamente (HOOD; WILSON, 2001).

1.1. Motivação e Escopo

As bibliométricas foram desenvolvidas para possibilitar o estudo de redes de colaboração científica. Entretanto, os algoritmos para extração de algumas delas não são escaláveis, como, por exemplo, os de transitividade, caminho médio e diâmetro (VALENCIO et al., 2017; VALENCIO et al., 2020). Isso dificulta a extração de conhecimento conforme cresce o número de pesquisadores e coautorias (VALENCIO et al., 2017; VALENCIO et al., 2020).

Dessa forma, o estudo de grandes redes de colaboração científica por meio das bibliométricas pode se tornar impraticável (MENA-CHALCO et al., 2014; VALENCIO et al., 2017; VALENCIO et al., 2020). Assim, tendo em vista a importância de sua caracterização, verifica-se que esforços no sentido de facilitar e tornar possível a análise de trabalhos científicos no contexto atual de grandes redes de colaboração científica são tarefas importantes (MENA-CHALCO et al., 2014; VALENCIO et al., 2017; VALENCIO et al., 2020).

Para isso, faz-se necessário a otimização dos algoritmos de extração de bibliométricas. Uma das formas é projetá-los para utilizar todos os recursos disponibilizados por plataformas de computação de alto desempenho de forma eficiente (WU et al., 2013). Um dos meios para isso é utilizar os modelos de programação paralela e distribuída (WU et al., 2013).

1.2. Objetivos

Tendo em vista a importância e as dificuldades do estudo e caracterização de redes de colaboração científica, o objetivo desse trabalho é o desenvolvimento de algoritmos de extração de bibliométricas, otimizados por meio de distribuição de dados, com desempenho superior aos semelhantes encontrados na literatura, para grandes redes de colaboração científica.

Ademais, como subproduto, foi desenvolvida uma Ferramenta de Extração de Indicadores Bibliométricos composta com os seguintes itens principais:

a) importador de informações científicas;
b) banco de dados unificado orientado a grafos;
c) extração de bibliométricas com os algoritmos otimizados.

As otimizações realizadas por esse trabalho possibilitam a extração de bibliométricas de redes de colaboração científica de forma escalável, e, por consequência, possibilita o estudo de produções científicas nos diversos âmbitos, seja nos limites locais ou envolvendo demais países, e em tempo hábil (VALENCIO et al., 2017; VALENCIO et al., 2020).

1.3. Contribuições científicas

As contribuições científicas desse trabalho são: o desenvolvimento de algoritmos de extração de bibliométricas, distribuídos e escaláveis, com desempenho superior aos encontrados na literatura para grandes redes de colaboração científica e, como subproduto, o desenvolvimento de uma ferramenta para extração de indicadores bibliométricos para facilitar a extração de conhecimento de redes de colaboração científica.

1.4. Organização da monografia

O restante da monografia está organizado em outras quatro seções. No capítulo 2 é apresentada a fundamentação teórica com os conceitos relevantes para entendimento do trabalho, tais como: redes de colaboração, Cienciometria e Bibliometria, Big Data, paralelização e distribuição de algoritmos, programação funcional e bancos de dados não relacionais (NoSQL), além de uma revisão bibliográfica sobre a evolução das pesquisas em extração de conhecimento sobre redes de colaboração científicas e trabalhos correlatos. No capítulo 3 é exposto de forma detalhada o desenvolvimento dos algoritmos otimizados de extração de bibliométricas. No capítulo 4 é retratado o resultado dos experimentos e testes realizados sobre os algoritmos desenvolvidos, os quais foram subdivididos em quatro categorias: testes de corretude, casos de uso, testes de desempenho e testes de escalabilidade. Por fim, no capítulo 5 são apresentadas as conclusões, as contribuições científicas e as propostas de trabalhos futuros.

2. Fundamentação Teórica

Neste capítulo são apresentados os conceitos de Cienciometria e Bibliometria, as principais métricas da bibliometria, o conceito de paralelização e distribuição de algoritmos, a quarta geração de banco de dados, uma revisão bibliográfica da literatura focada nos estudos bibliométricos e nas ferramentas e algoritmos desenvolvidos e, por fim, a comparação entre os trabalhos correlatos.

2.1. Cienciometria e Bibliometria

Os trabalhos da literatura, em sua maioria, representam as redes de colaboração científica como grafos G (V, A), no qual os pesquisadores são representados pelos vértices V e os trabalhos em coautoria são representados pelas arestas A (MENA-CHALCO et al., 2014; CHEN et al., 2017). Além disso, dois pesquisadores são considerados conectados caso tenham um ou mais trabalhos desenvolvidos em coautoria (NEWMAN, 2001a, 2001b).

Essa forma de representação possibilitou o desenvolvimento de métricas baseadas na teoria dos grafos, denominadas bibliométricas, que buscam facilitar a extração de conhecimento das redes de colaboração científica. Essas bibliométricas foram fundamentadas pela Cienciometria e pela Bibliometria (MENA-CHALCO, DIGIAMPIETRI, CESAR-JR, 2012; MENA-CHALCO et al., 2014; VALENCIO et al., 2017; VALENCIO et al., 2020).

A Cienciometria é o estudo dos aspectos quantitativos da ciência como uma disciplina ou atividade econômica. Seus tópicos mais relevantes são o levantamento de bibliométricas, dados científicos, visualização e modelagem de relações, eficiência e tendências

de coautorias entre os pesquisadores (MENA-CHALCO et al., 2014; VALENCIO et al., 2017; VALENCIO et al., 2020).

A Bibliometria, por sua vez, é o estudo dos aspectos quantitativos das produções escritas e, caso utilizado para produções científicas, torna-se um dos tópicos da Cienciometria. Seus itens mais relevantes são o uso de padrões, modelos matemáticos e teoria dos grafos para extração de métricas que apoiam as tomadas de decisões e a elaboração de previsões (TAGUE-SUTCLIFFE, 1992; VALENCIO et al., 2017; VALENCIO et al., 2020).

Alguns esforços têm sido despendidos com o objetivo de caracterizar a ciência, a saber: Olawumi e Chan (2018), por exemplo, analisaram 2096 trabalhos científicos relacionados aos temas sustentabilidade e desenvolvimento sustentável, a fim de permitir melhor entendimento das pesquisas e poder guiar trabalhos futuros por meio de tendências e padrões encontrados; Heilg e Voß (2014), por sua vez, estudaram a evolução das pesquisas em computação em nuvem, a fim de incentivar o compartilhamento de conhecimento e a colaboração entre pesquisadores.

Esses trabalhos evidenciam a importância da extração de conhecimento de redes de colaboração científica a fim de caracterizar a ciência e avaliar sua evolução. Para realizar essa caracterização, pode-se extrair indicadores quantitativos por meio das bibliométricas e analisá-los.

2.2. Bibliométricas

Dentre as bibliométricas baseadas na teoria dos grafos para extração de conhecimento e análise das redes de colaboração, destacam-se (MENA-CHALCO, DIGIAMPIETRI E CESAR-JR, 2012; MENA-CHALCO et al., 2014; VALENCIO et al., 2017; VALENCIO et al., 2020):

a) número total de pesquisadores;
b) número total de coautorias;
c) número total de trabalhos em coautorias;
d) número total de coautores;
e) média de trabalhos em coautoria;
f) densidade das coautorias;
g) transitividade;
h) assortatividade;
i) distância média entre os pesquisadores;

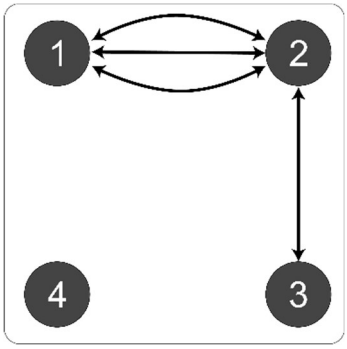
j) diâmetro;

k) tamanho do maior grupo de coautores;

l) porcentagem do maior grupo de coautores.

Esse conjunto de bibliométricas, quando extraídos e analisados, permitem extrair características das redes de colaboração. Devido a essa importância, são explicadas com um pouco mais de detalhes nos textos que seguem. Além disso, a extração das bibliométricas da rede de colaboração ilustrada na Figura 1 será executada para exemplificar o processo.

Figura 1 – rede de colaboração exemplificada



Fonte: Elaborado pelo autor

2.2.1. Número total de pesquisadores

O número total de pesquisadores é uma medida simples que dimensiona a rede de colaboração científica. Seu cálculo é executado por meio da quantidade total de vértices, ou seja, por |V| (MENA-CHALCO et al., 2014; VALENCIO et al., 2017; VALENCIO et al., 2020). Seu valor para a rede de colaboração da Figura 1 é de 4 pesquisadores, pois seu grafo tem quatro vértices.

2.2.2. Número total de coautorias

O número total de coautorias entre os pesquisadores reflete a conectividade entre eles. Além disso, é possível também avaliar a proximidade entre os pesquisadores ao longo do tempo. Seu cálculo é realizado mediante a incidência de arestas entre dois pesquisadores, ou seja, se houver uma ou mais arestas entre dois pesquisadores, soma-se um ao total de coautorias (MENA-CHALCO et al., 2014; VALENCIO et al., 2017; VALENCIO et al., 2020). Seu valor para a rede de colaboração da Figura 1 é de 2 coautorias, pois há dois pares de vértices adjacentes, são eles: (1, 2) e (2, 3).

2.2.3. Número total de trabalhos em coautorias

O número total de trabalhos desenvolvidos em coautoria entre os pesquisadores demonstra o aumento de produtividade entre eles ao longo do tempo. Seu cálculo é efetuado por meio da quantidade total de arestas do grafo, ou seja, por |A| (MENA-CHALCO et al., 2014; VALENCIO et al., 2017; VALENCIO et al., 2020). Seu valor para a rede de colaboração da Figura 1 é de 4 trabalhos, pois seu grafo tem quatro arestas.

2.2.4. Número total de coautores

O número total de pesquisadores com ao menos um trabalho em coautoria retrata a quantidade de pesquisadores que se aproveitaram das vantagens do desenvolvimento de pesquisas científicas em colaboração. Dado que o grau de um vértice indica o número total de arestas que incidem sobre ele, o cálculo dessa bibliométrica é executado por intermédio da quantidade de vértices de grau maior que zero (MENA-CHALCO et al., 2014; VALENCIO et al., 2017; VALENCIO et al., 2020). Seu valor para a rede de colaboração da Figura 1 é de 3 pesquisadores, pois seu grafo tem três vértices com grau maior que zero, são eles: 1, 2 e 3.

2.2.5. Média de trabalhos em coautoria

O número médio de trabalhos realizados em coautoria representa a produtividade média dos pesquisadores. O cálculo dessa bibliométrica é efetuado por meio da Equação 1, que é a razão entre a soma dos graus dos vértices e a quantidade de pesquisadores (MENA-CHALCO et al., 2014; VALENCIO et al., 2017; VALENCIO et al., 2020).

$$M.T.C. = \frac{\sum_i deg(v_i)}{|V|} \tag{1}$$

Seu valor para a rede de colaboração da Figura 1 é de 2 trabalhos por pesquisador e seu cálculo é exemplificado na Equação 2.

$$M.T.C. = \frac{3+4+1+0}{4} = \frac{8}{4} = 2 \tag{2}$$

2.2.6. Densidade das coautorias

A densidade das coautorias é a relação entre a quantidade de coautorias e a maior quantidade de coautorias possível entre os pesquisadores. Dado que um grafo completo é um grafo simples em que todo vértice é adjacente a todos os outros vértices, essa bibliométrica é calculada por meio da razão entre a quantidade de arestas do grafo G (V, A) e a quantidade de arestas de um grafo completo de mesmo número de vértices G' (V', A'). O valor da densidade varia de 0 a 1 e pode ser calculada por meio da Equação 3 (MENA-CHALCO et al., 2014; VALENCIO et al., 2017; VALENCIO et al., 2020).

$$Densidade = \frac{|A|}{\frac{|V| * (|V| - 1)}{2}} \quad (3)$$

Seu valor para a rede de colaboração da Figura 1 é $\frac{2}{6}$, pois seu grafo tem duas arestas e um grafo completo de mesmo número de vértices tem seis arestas.

2.2.7. Transitividade

A transitividade é a probabilidade de dois coautores de um pesquisador também serem coautores entre si e representa a compactação da rede. É calculada por meio do coeficiente de *clustering*: probabilidade de dois vértices adjacentes a um vértice *i* também serem adjacentes entre si, ou seja, a razão entre o número de adjacências existentes entre os vértices vizinhos a *i* e o número máximo de adjacências possíveis entre eles. Os valores possíveis da transitividade variam de 0 a 1 e seu valor, para cada vértice, é calculado por meio da Equação 4, na qual AEV representa a quantidade de adjacência entre os vizinhos (coautorias) e QVA representa a quantidade de vizinhos adjacentes. A transitividade do grafo é a média da transitividade de todos os vértices que tenham dois ou mais vértice adjacente (MENA-CHALCO et al., 2014; VALENCIO et al., 2017; VALENCIO et al., 2020).

$$T.do\ vértice = \frac{AEV}{\frac{QVA * (QVA - 1)}{2}} \quad (4)$$

Seu valor para a rede de colaboração da Figura 1 é 0, pois o único vértice com dois ou mais vértices adjacentes é o 2 e o valor da sua transitividade é 0, como demonstrado na Equação 5.

$$T.do\ vértice\ 2 = \frac{0}{\frac{2 * (2-1)}{2}} = 0 \quad (5)$$

2.2.8. Assortatividade

A assortatividade é a probabilidade de um pesquisador ter a mesma quantidade de trabalhos em coautoria que seus coautores e representa a homogeneidade da rede. Essa bibliométrica é calculada por meio do coeficiente de Pearson, que representa a preferência dos vértices serem adjacentes a outros vértices de grau similar. Os valores possíveis da assortatividade variam de -1 a 1, no qual valores positivos indicam preferência de adjacência entre vértices de graus similares e valores negativos indicam preferência de adjacência entre os vértices de graus diferentes. O cálculo do coeficiente de Pearson é ilustrado na Equação 6, na qual M representa a quantidade de arestas e j_i e k_i são os graus dos vértices incidentes à i-ésima aresta, com *i* = 1, ..., M (NEWMAN, 2002).

$$r = \frac{M^{-1}\sum_i j_i k_i - \left[M^{-1}\sum_i \frac{1}{2}(j_i + k_i)\right]^2}{M^{-1}\sum_i \frac{1}{2}(j_i^2 + k_i^2) - \left[M^{-1}\sum_i \frac{1}{2}(j_i + k_i)\right]^2} \quad (6)$$

Seu valor para a rede de colaboração da Figura 1 é $-0{,}6$, seu cálculo é exemplificado na Equação 7.

$$r = \frac{\frac{1}{4} * 40 - \left[\frac{1}{4} * \frac{26}{2}\right]^2}{\frac{1}{4} * \frac{92}{2} - \left[\frac{1}{4} * \frac{26}{2}\right]^2} = \frac{10 - 10{,}56}{11{,}5 - 10{,}56} = -\frac{0{,}56}{0{,}94} = -0{,}6 \quad (7)$$

2.2.9. Distância média entre os pesquisadores

A média das distâncias mínimas entre todos os pesquisadores representa a proximidade entre eles. Seu cálculo é efetuado por meio da média de todos os pares de caminhos mínimos calculados entre todos os vértices do grafo. Dado que um caminho em um grafo consiste em uma sequência finita alternada de vértices e arestas, que começa e termina com vértices e em que cada aresta é incidente ao vértice que a precede e a sucede, sem repetição de vértices, o caminho mínimo é o menor caminho entre um par de vértices (MENA-CHALCO et

al., 2014; VALENCIO et al., 2017; VALENCIO et al., 2020). Seu valor para a rede de colaboração da Figura 1 é de $\frac{4}{3}$ arestas, pois há 3 caminhos mínimos entre os pares de vértices (1, 2), (2, 3) e (1, 3) com os valores 1, 1 e 2, respectivamente.

2.2.10. Diâmetro

O diâmetro representa a maior distância mínima entre dois pesquisadores de toda a rede. O diâmetro é o maior caminho mínimo dentre todos os caminhos mínimos do grafo. Os valores possíveis do diâmetro variam de 1 até $|V| - 1$, em que V é o número de vértices do grafo (WASSERMAN; FAUST, 1994). Seu valor para a rede de colaboração da Figura 1 é 2 arestas, valor do caminho mínimo entre o par de vértices (1, 3).

2.2.11. Tamanho do maior grupo de coautores

O tamanho do maior grupo de coautores representa a quantidade de pesquisadores do maior grupo de coautores conectados da rede de colaboração. Dado que um grafo é conexo caso existir um caminho entre qualquer par de vértices, o cálculo dessa bibliométrica efetuado por meio do número de vértice do maior subgrafo G'' (V'', A'') conexo do grafo, ou seja, por |V''| (MENA-CHALCO et al., 2014; VALENCIO et al., 2017; VALENCIO et al., 2020). Seu valor para a rede de colaboração da Figura 1 é 3 pesquisadores, representados pelos vértices 1, 2 e 3.

2.2.12. Porcentagem do maior grupo de coautores

A porcentagem do maior grupo de coautores em relação à rede de coautoria possibilita dimensioná-lo. Pode haver mais que um grupo de coautores, pois há grupos que trabalham de forma isolada de outros grupos. O estudo do maior grupo de coautores é importante, pois ele representa a rede de colaboração em si. Seu cálculo é executado mediante à razão entre o tamanho do maior grupo de coautores e o número total de pesquisadores e seus possíveis valores variam entre 0 a 1. (MENA-CHALCO et al., 2014; VALENCIO et al., 2017; VALENCIO et al., 2020). Seu valor para a rede de colaboração da Figura 1 é 75%.

2.2.13. Considerações finais

Nesta seção foram apresentadas as principais bibliométricas utilizadas para a extração de conhecimento de redes de colaboração científica. Por meio de cada uma delas, é possível extrair um conhecimento diferente e, em conjunto, é possível analisar a rede de colaboração científica de forma mais completa. Um resumo com o conhecimento que cada bibliométrica fornece sobre a rede de colaboração científica é apresentado na Tabela 1.

Tabela 1 – conhecimento extraído por cada bibliométricas

Bibliométrica	**Conhecimento extraído**
Número total de pesquisadores	Dimensiona a rede de colaboração
Número total de coautorias	Conectividade entre os pesquisadores
Número total de trabalhos em coautorias	Produtividade dos pesquisadores
Número total de coautores	Quantidade de pesquisadores que utilizaram a colaboração para desenvolvimento de trabalhos
Média de trabalhos em coautoria	Produtividade média dos pesquisadores
Densidade das coautorias	Grupo de pesquisadores denso ou esparso
Transitividade	Compactação da rede de colaboração
Assortatividade	Homogeneidade da rede de colaboração
Distância média entre os pesquisadores	Proximidade entre os pesquisadores
Diâmetro	Maior distância mínima entre dois pesquisadores
Tamanho do maior grupo de pesquisadores	O maior grupo interligado de pesquisadores
Porcentagem do maior grupo de pesquisadores	Relação entre o tamanho do maior grupo de pesquisadores e o tamanho rede de colaboração

Fonte: Elaborado pelo autor

2.3. Paralelização e Distribuição de algoritmos

Uma vez que alguns dos algoritmos para extração de bibliométrica foram apresentados, deve-se considerar que alguns deles não são escaláveis, como, por exemplo, os de transitividade, caminho médio e diâmetro, pois tem o desempenho degradado conforme cresce a quantidade de dados a ser processada, como apresentado por Valencio et al. (2017) e Valencio et al. (2020). Isso dificulta a extração de conhecimento conforme cresce o número de pesquisadores e coautorias (MENA-CHALCO et al., 2014; VALENCIO et al., 2017; VALENCIO et al., 2020).

Não só as redes de colaboração científica cresceram, mas também, com a evolução da computação, o advento da Internet e da internet das coisas, houve uma geração massiva de dados em diversos âmbitos, o que introduz desafios em relação ao armazenamento,

processamento e extração de conhecimento sobre esses dados (GANTZ; REINSEL, 2011; BHADANI; JOTHIMANI, 2016). Esse fenômeno originou o Big Data, o qual se fundamenta em um modelo de 5 V's (CHEN; MAO; LIU, 2014; FAHAD et al., 2014; STOREY; SONG, 2017), a saber:

a) volume – grande quantidade de dados gerados em alta velocidade e armazenados em escalas de terabytes, petabytes e exabytes;
b) velocidade – para computação de respostas em tempo hábil é necessária alta velocidade de criação, coleta, extração, processamento e *streaming* de dados;
c) variedade – dados armazenados em diferentes formas, como estruturados, semiestruturados, não estruturados, arquivos binários, textos, entre outras;
d) valor – dados com baixa densidade, mas com informações relevantes para o negócio. É um dos desafios mais complexos do Big Data, pois sua extração depende da lógica de negócios ou do domínio do problema;
e) veracidade – assim como em pequenas bases de dados, a precisão das informações é imprescindível. É por meio desse V que se trata os desafios de inconsistência e qualidade dos dados no universo Big Data (STOREY; SONG, 2017).

Adicionalmente, há também o modelo de 7 V's, no qual são acrescentados os desafios de validade e volatilidade. Por fim, há ainda outros V's, tais como visualização, variabilidade, vulnerabilidade, visibilidade, entre outros (STOREY; SONG, 2017).

Para lidar com esses desafios de manipulação dos dados introduzidos pelo Big Data, torna-se necessário utilizar plataformas de computação de alto desempenho, tais como clusters de computadores ou computação em nuvem. De modo a extrair todo o potencial de processamento dessas plataformas, faz-se necessário o desenvolvimento de algoritmos com suporte ao processamento paralelo e distribuído (WU et al., 2013).

Há duas formas mais utilizadas na literatura para paralelização de algoritmos: paralelismo de tarefas e paralelismo de dados (N'TAKPÉ; SUTER; CASANOVA, 2007; NA et al., 2014). No paralelismo de tarefas, a aplicação é dividida em uma série de tarefas que são executadas de forma simultânea de acordo com uma ordem pré-estabelecida. Por outro lado, no paralelismo de dados, normalmente utilizado em estruturas de repetição, é utilizado o conceito de *Single Instruction Multiple Data* (SIMD), ou seja, a mesma instrução é executada simultaneamente para o processamento de múltiplos dados. De modo a obter-se maior

escalabilidade, ambas as formas podem ser utilizadas em conjunto, tal abordagem origina o termo *mixed parallelism* (N'TAKPÉ; SUTER; CASANOVA, 2007; NA et al., 2014).

Para a distribuição dos algoritmos, problemas complexos são introduzidos, tais como: balanceamento de carga, performance da rede e tolerância a falhas. Dessa forma, *frameworks* foram propostos (DA SILVA MORAIS, 2015) para abstrair esses problemas e facilitar o desenvolvimento: um deles é o MapReduce, que se destaca por sua complexidade moderada, flexibilidade e tolerância a falhas (JUKIC; SUBASI, 2017; STOREY; SONG, 2017). Ele e outras importantes ferramentas para distribuição de algoritmos são detalhadas a seguir.

2.3.1. MapReduce

De modo a facilitar o desenvolvimento de algoritmos com processamento distribuído, foi desenvolvido pela Google um modelo de programação chamado MapReduce, que executa algoritmos automaticamente de forma paralela caso tenha as funções de *map* e *reduce* implementadas (DEAN; GHEMAWAT, 2008). Essas funções são a essência de duas fases distintas para a execução do MapReduce (BHADANI; JOTHIMANI, 2016):

a) *map* – fase em que a carga de trabalho é dividida em cargas menores, e origina uma lista ordenada de pares (chave, valor);
b) *reduce* – fase em que a lista ordenada produzida na fase anterior é analisada e combinada para produzir o resultado do algoritmo.

Além disso, o uso do MapReduce apresenta algumas vantagens, como por exemplo (LEE, 2012):

a) complexidade moderada – requer que se implemente as funções *Map* e *Reduce*;
b) flexível – não há modelo e esquema de dados, ou seja, o programador tem maior facilidade para lidar com dados não estruturados e irregulares;
c) tolerante a falhas – pode continuar o processamento mesmo tendo em média 1,2 falhas por execução, como avaliado pelo Google.

Dentre as diversas implementações do MapReduce, destaca-se a do Apache Hadoop, que é um software de código aberto que lida com grandes quantidades de dados em tempo real por meio da distribuição do processamento dos dados entre *clusters* de máquinas (NASSER; TARIQ, 2015).

2.3.2. Apache Hadoop

Uma das formas de utilizar o MapReduce é por meio do Apache Hadoop, uma infraestrutura que tem como objetivo armazenar e processar grandes volumes de dados de forma paralela e distribuída e que, para atingir todo o seu potencial, deve ter a disposição um cluster de computadores. Além disso, tem duas funcionalidades fundamentais (DA SILVA MORAIS, 2015):

a) YARN (*Yet Another Resource Negotiator*) – gerencia e atribui os recursos disponíveis às aplicações e processos executados. As primeiras versões do Hadoop suportavam apenas as aplicações desenvolvidas com o MapReduce (DA SILVA MORAIS, 2015);
b) HDFS (*Hadoop Distributed File System*) – um sistema de arquivos distribuídos que abrange todos os nós do cluster e interliga todos os sistemas de arquivos individuais de cada máquina para formar um único e grande sistema de arquivos (DA SILVA MORAIS, 2015). Além disso, é responsável por oferecer alta disponibilidade e tolerância a falhas (SINGH; REDDY, 2015).

Entretanto, o framework MapReduce tem como uma de suas principais desvantagens a ineficiência na execução de algoritmos iterativos, no qual há uma alta demanda de operações de entrada e saída no disco que sobrecarregam o HDFS e degradam o desempenho do sistema (SINGH; REDDY, 2015).

2.3.3. Apache Spark

Com o objetivo de superar as limitações de entrada e saída de disco do MapReduce, foi desenvolvido o Apache Spark, uma plataforma para o tratamento de grandes volumes de dados, na ordem de gigabytes (SINGH; REDDY, 2015). Seu processamento é similar, porém mais eficiente que o do Hadoop MapReduce (DA SILVA MORAIS, 2015; SHI et al., 2015), podendo ser cerca de cem vezes mais rápido em memória e dez vezes em disco (SINGH; REDDY, 2015; STOREY; SONG, 2017).

Para isso, o Spark utiliza coleções de objetos somente leitura particionadas e distribuídas entre as máquinas do cluster e denominadas *Resilient Distributed Datasets* (RDDs). Dentre as vantagens dessas coleções encontra-se a alta tolerância a falha, visto que podem ser reconstruídas caso uma partição seja perdida. Essas coleções são imutáveis e podem, também,

serem armazenadas em sistemas de armazenamentos externos, tais como o HDFS (DA SILVA MORAIS, 2015; SINGH; REDDY, 2015).

Os principais componentes do Spark são descritos a seguir (DA SILVA MORAIS, 2015; SINGH; REDDY, 2015):

a) shark SQL – ferramenta para execução de consultas SQL e processamento de dados estruturados;
b) spark *streaming* – ferramenta para processamento de dados em tempo real;
c) MLib – ferramenta para aprendizado de máquina;
d) GraphX – ferramenta para processamento de grafos.

Uma vez que as redes de colaboração são representadas por meio de grafos, o componente do Apache Spark ideal para manipulá-las é o Apache GraphX.

2.3.4. Apache GraphX

Para realizar o processamento distribuído e especializado de grafos, foi desenvolvido sobre o Apache Spark a *Application Programming Interface* (API) Apache GraphX (GRAPHX, 2020), que disponibiliza o suporte à operadores básicos, como funções de mapeamento e junção, para o desenvolvimento de algoritmos para o processamento dos grafos. Além disso, os algoritmos desenvolvidos sobre ele contam com suporte ao processamento distribuídos dos dados e tolerância a falhas (GONZALEZ *et al*, 2014).

Apresenta, também, uma abstração que permite que os dados sejam representados como um grafo ou como uma tabela, sem precisar modificá-los ou duplicá-los para isso. Ademais, em adição aos operadores básicos, tais como o de mapeamento e redução, são disponibilizados operadores especializados em grafos, tais como subgraph e mrTriplets, responsáveis por processar os grafos de forma distribuída. Uma lista completa e detalhada das operações pode ser encontrada em Apache Spark (2019). Destaca-se que, todas essas operações, quando realizadas sobre os RDD, são automaticamente processadas de forma distribuída (APACHE SPARK, 2019).

Para disponibilizar esses operadores que abstraem a complexidade de distribuição do processamento, é utilizado o paradigma de programação funcional, no qual o processamento é realizado principalmente por meio de funções. Para isso, as funções são tratadas de forma que podem ser passadas como parâmetro para outras funções. Além disso, a função pode ser atribuída a uma variável, ou até ser o valor de retorno de outra função (DAVIS, 2016).

Por fim, as operações são separadas em duas categorias: operações de transformação, responsáveis por criar conjuntos de dados a partir conjuntos de dados existentes, ou seja, são transformados; e operações de ação, normalmente encadeadas após a sequência de operações de transformação, responsáveis por recuperar os dados. Esses operadores são simples, mas importante o suficiente para implementar abstrações como o Pregel API (XIN *et al*, 2014).

2.3.5. Pregel API

Um dos operadores capazes de implementar algoritmos como o *Breadth-first search* (BFS) ou busca em largura, utilizado para implementação do caminho médio e diâmetro (VALENCIO et al., 2017; VALENCIO et al., 2020) é o Pregel API, um operador que gerencia o envio de mensagens síncronas em massa restringidas à topologia do grafo. Seu funcionamento se baseia na execução iterativa de *super steps*. Um *super step* é uma das iterações do Pregel API, composta por três passos:

a) recebimento de mensagens – os vértices recebem e fundem mensagens de seus vizinhos;
b) cálculo de novo valor – um novo valor é calculado para o vértice, esse cálculo é baseado nas mensagens recebidas;
c) novas mensagens – preparam novas mensagens a serem enviadas para seus vizinhos no próximo *super step*.

Para realizar seu processamento, o Pregel recebe três funções como parâmetro, invocadas para processar os três passos citados anteriormente. Por fim, os vértices que não receberem mensagens no *super step* não são processados e, quando não houver mais mensagens, o Pregel é finalizado e retorna o novo grafo processado (GRAPHX, 2020).

2.4. NoSQL

Os desafios do Big Data incidem também sobre o armazenamento e recuperação de informações, dado que os Sistemas Gerenciadores de Banco de Dados (SGBDs) relacionais tradicionais não foram concebidos para suportar a demanda apresentada pelo Big Data (DAVOUDIAN; CHEN; LIU, 2018). Estes requisitos do Big Data determinaram a quarta geração de Banco de Dados, denominada NoSQL (do inglês, *Not Only Structured Query Language*) ou bancos de dados não relacionais, que têm como objetivo atender a alta

disponibilidade e escalabilidade necessários para as grandes aplicações. As principais características dessa nova geração são: modelo de dados flexível, transações de persistência de dados mais amenas, duplicação do armazenamento de dados para otimizar sua recuperação, suporte a distribuição de dados e interfaces simples para consulta dos dados (HAN et al., 2011; DAVOUDIAN; CHEN; LIU, 2018).

Os SGBDs NoSQL são categorizados de acordo com seu modelo de dados, ou seja, em como as entidades do mundo real são representadas. As principais categorias são (HAN et al., 2011; DAVOUDIAN; CHEN; LIU, 2018):

a) chave-valor;
b) colunas;
c) documentos;
d) grafos.

Os trabalhos da literatura, em sua maioria, utilizam-se de redes de colaboração científica como grafos (MENA-CHALCO et al., 2014; CHEN et al., 2017), deste modo, torna-se adequado o uso de um SGBD orientado a grafos para armazenar e recuperar as redes de colaboração. Isto porque esses SGBDs são voltados para redes sociais e tratam as relações entre os objetos com tanta importância quanto os próprios objetos (BATRA; TYAGI, 2012).

Além disso, em comparação aos SGBDs relacionais, os orientados a grafos são muito mais eficientes se levado em consideração dados conectados (JAISWAL; AGRAWAL, 2013). Dentre os motivos para isso, destacam-se alguns pontos dos SGBDs relacionais que degradam seu desempenho quando considerados grandes volumes de dados conectados, são eles: a grande quantidade de junções entre as múltiplas relações e a dependência de um rígido esquema de tabelas e relações (JAISWAL; AGRAWAL, 2013).

2.5. Fundamentos em Bibliometria e Processamento Paralelo

De forma a apresentar a evolução e o estado atual das pesquisas sobre estudos bibliométricos de redes de colaboração científica, nas próximas seções é apresentada a revisão bibliográfica da literatura.

2.5.1. Estudos bibliométricos de redes de colaboração científica

Conforme ocorreu o aumento do número de trabalhos científicos desenvolvidos em coautoria e a formação das primeiras redes de colaboração, notou-se um crescente número de

pesquisas direcionadas para seu estudo e caracterização. Por esse motivo, Subramanyam (1983) fez um levantamento bibliográfico e resumiu o estado da arte dos primeiros estudos bibliométricos realizados até o ano de 1982. Além disso, destacou a importância de métodos bibliométricos nos estudos das colaborações existentes nas pesquisas científicas.

Na década de 90, Van Raan (1996) demonstrou os potenciais e as limitações dos métodos bibliométricos para a avaliação dos pontos fortes e fracos das pesquisas científicas desenvolvidas em coautoria. Além disso, compara dois diferentes métodos bibliométricos e demonstra como a combinação de ambos pode resultar em uma nova, poderosa e avançada metodologia para observar o papel dos autores nos avanços científicos.

Posteriormente, Katz e Hicks (1997) utilizaram um modelo bibliométrico para verificar como a colaboração entre os autores aumenta o número de citações e o impacto dos artigos científicos. Uma das conclusões foi que a colaboração entre pesquisadores de mesma universidade/região aumenta o número de citações em 0,75, enquanto a colaboração entre pesquisadores de instituições estrangeiras as aumenta em 1,6.

Em continuidade aos estudos, Ramos-Rodríguez e Ruíz-Navarro (2004) extraíram bibliométricas sobre os trabalhos de muitos autores para analisarem a evolução das pesquisas na área de estratégias de gerenciamento. Por fim, os autores reforçaram que estudos quantitativos são apenas um complemento, e não um substituto, aos estudos qualitativos tradicionais de revisões bibliográficas.

Também, Archambault et al. (2009) extraíram e compararam as bibliométricas de redes de colaboração de dois grandes repositórios científicos, o Web of Science (WoS) e o Scopus. O resultado foi que a correlação entre as métricas obtidas entre ambos os repositórios para o número de trabalhos e o número de citações agrupadas por países foram extremamente altas. Tais resultados evidenciaram que as bibliométricas podem ser utilizadas para a caracterização da ciência em nível nacional.

Em seguida, Mena-Chalco, Digiampietri e Cesar-Jr (2012) desenvolveram um algoritmo para identificação automática e rápida de coautorias em produções bibliográficas para facilitar a extração de redes de coautorias de repositórios de armazenamento como a Plataforma Lattes. Além disso, realizaram a caracterização topológica das redes de coautorias de grupos de pesquisadores cadastrados na Plataforma Lattes identificados por meio do algoritmo. Na avaliação topológica, foram utilizadas bibliométricas baseadas na teoria dos grafos.

Posteriormente, as pesquisas culminaram no trabalho de Mena-Chalco et al. (2014), no qual foi identificada e caracterizada a rede de colaboração científica brasileira. Os passos para isso foram:

a) estratégias para importar os currículos da base de dados da Plataforma Lattes para um banco de dados unificado;
b) algoritmos para identificação automática de coautorias por meio das referências bibliográficas;
c) dez métricas topológicas baseadas na teoria dos grafos (bibliométricas) para identificação quantitativa das interações entre os pesquisadores.

No estudo foram avaliadas as informações de muitos pesquisadores associados com as oito grandes áreas do conhecimento e de forma trienal. Com isso, obteve-se um profundo conhecimento sobre a estrutura e o comportamento social dos pesquisadores das redes de colaboração científica brasileiras.

Mais tarde, Bornmann, Wagner e Leydesdorff (2015) realizaram o estudo da evolução do número de citações recebidas pelos trabalhos realizados pelos pesquisadores dos países integrantes do BRICS (acrônimo dado aos cinco países emergentes que demonstraram grande crescimento econômico nos últimos anos, são eles: Brasil, Rússia, Índia, China e África do Sul). Nessa análise, foi verificado que as pesquisas desenvolvidas nos países do BRICS, com exceção da Rússia, tiveram uma taxa anual de crescimento do número de citações maior do que as dos países com maior número de citações.

Por fim, foram reproduzidas redes de colaboração entre os autores das pesquisas mais citadas dos anos de 1995, 2000, 2005 e 2010 e foi constatado que todos os países integrantes do BRICS tiveram pesquisadores participantes nelas.

Em continuidade aos estudos, Finardi e Buratti (2016) estudaram a rede de colaboração existente entre os cinco países do BRICS e 65 países com mais coautorias com eles. Os resultados mostraram que as colaborações entre os cinco países do BRICS são frágeis, o que indica a necessidade de estimular a pesquisa científica entre eles. A melhora das colaborações científicas entre eles pode resultar em efeitos positivos no desenvolvimento social e econômico.

Por fim, Zhang et al. (2018), realizaram um estudo sistemático das colaborações científicas, para isso, consideraram as características dos pesquisadores, tais como: produtividade, número de citações, interesses de pesquisa e gênero. Algumas conclusões sobre os resultados do estudo foram as seguintes:

a) pesquisadores com mais trabalhos em coautorias tem maior probabilidade em participar de novas coautorias, enquanto pesquisadores com números similares de publicações de autoria única tem menor preferência;
b) o número de citações não influencia as preferências de coautoria entre os pesquisadores;
c) interesses de pesquisa e gêneros semelhantes influenciam fortemente à formação de novas colaborações;
d) a transitividade entre os autores, ou seja, a probabilidade de dois coautores de um pesquisador também serem coautores entre si, tem um dos maiores impactos na formação das redes de colaboração.

2.5.2. Ferramentas e algoritmos para análise de redes de colaboração científica

Dada a importância das redes de colaboração, Van Eck e Waltman (2010) desenvolveram uma ferramenta focada em sua representação gráfica. O trabalho foi dividido em três seções, são elas:

a) funcionalidades – as funcionalidades da ferramenta (VOSviewer) foram resumidas. As funcionalidades variam desde a exibição do grafo da rede de colaboração até mapas de calor;
b) implementação – os detalhes de implementação da ferramenta foram discutidos;
c) demonstração – a capacidade da ferramenta foi demonstrada com a exibição de um mapa de coautorias de 5000 jornais científicos.

Posteriormente, no trabalho de Valencio et al. (2017), foi desenvolvida uma ferramenta para extração de indicadores bibliométricos de redes de colaboração genéricas que contava com:

a) importador de currículos da Plataforma Lattes, baseado no SASD/Lattes (GBD, 2019);
b) versão simplificada de um importador de currículos de arquivos CSV e de bancos de dados relacionais limitado ao SGBD PostgreSQL (POSTGRESQL, 2019);
c) extrator de dez bibliométricas e paralelização por meio de *multithreading* de três delas: transitividade, distância média e diâmetro.

Por fim, analisaram a rede de colaboração da Universidade Estadual Paulista "Júlio de Mesquita Filho" (UNESP) das seguintes formas:

a) geral – extração e análise das bibliométricas de todos os tempos e todos os pesquisadores da faculdade;
b) trienal – extração e análise da evolução das bibliométricas ao longo do tempo, nos quais foram considerados todos os pesquisadores e períodos de três anos;
c) unidades – extração e análise das bibliométricas de cada unidade da universidade.

A ferramenta construída visa facilitar a extração de bibliométricas de redes de colaboração genéricas em tempo hábil. Entretanto, por ter os algoritmos mais custosos computacionalmente apenas paralelizados, sua escalabilidade se dá de forma vertical, o que a torna mais cara e limitada. Para resolver esse problema é necessário distribuir os algoritmos, para a escalabilidade ser de forma horizontal.

Mais tarde, Schoeneman e Zola (2019) propuseram, implementaram e analisaram diferentes estratégias para o algoritmo de extração de todos os caminhos mínimos – All-Pairs Shortest-Paths (APSP), algoritmo base utilizado para a implementação das bibliométricas de caminho médio e diâmetro. Os estudos demonstraram que, por meio da implementação básica do Apache Spark, sem ferramentas adicionais, torna-se impraticável o desenvolvimento do algoritmo de extração de todos os caminhos mínimos (APSP) eficiente e escalável.

Por fim, o trabalho de Valencio et al. (2017) foi revisado e expandido no trabalho de Valencio et al. (2020), no qual foram apresentados novos testes e resultados, além de mais detalhes sobre a implementação dos algoritmos. O trabalho demonstrou a evolução da ciência em uma universidade entre os triénios de 1990~1992 até 2014~2016 por meio do estudo de uma rede de colaboração de mais de 3000 pesquisadores.

2.5.3. Comparação entre os trabalhos correlatos

Nesta seção, foi apresentada uma revisão bibliográfica da literatura, na qual foram apresentados trabalhos que explicitam a importância das bibliométricas na caracterização das redes de colaboração científica e, consequentemente, na caracterização da ciência em um determinado escopo. Dentre eles, os trabalhos de Mena-Chalco et al. (2014) e Valencio et al. (2017) foram escolhidos como correlatos e comparados a fim de verificar os pontos que cada um aborda, o resultado é exibido na Tabela 2. Ao analisar os resultados, nota-se que nenhum dos trabalhos correlatos aborda a distribuição dos algoritmos das bibliométricas ou a importação de redes de colaboração de qualquer SGBD relacional e dos formatos de arquivo XML e JSON. Dessa forma, valida-se a necessidade de uma ferramenta genérica e otimizada para essa tarefa.

Tabela 2 – comparação entre os trabalhos correlatos

Características	Mena-Chalco et al. (2014)	Valencio et al. (2017)
Visualização comparativa das bibliométricas	✓	✗
Extração de bibliométricas	✓	✓
Análise de bibliométricas	✓	✓
Armazenamento em banco de dados orientado a grafos	✓	✓
Ferramenta para extração de rede de colaboração genérica	✗	✓
Paralelização das métricas mais custosas	✗	✓
Filtros para a rede de colaboração	✗	✓
Distribuição das bibliométricas	✗	✗
Importador de redes de colaboração de SGBD relacional PostgreSql, Plataforma Lattes e o formato de arquivo CSV	✗	✓
Importador de redes de colaboração de qualquer SGBD relacional e os formatos de arquivo XML e JSON	✗	✗

Fonte: Elaborado pelo auto

2.6. Considerações finais

Nesse capítulo foram apresentados conceitos relevantes sobre redes de colaboração, Cienciometria e Bibliometria, Big Data, Paralelização e Distribuição de algoritmos e seus frameworks e, por fim, NoSQL. O estudo desse conteúdo é relevante para viabilizar o desenvolvimento desse trabalho.

Além disso, foi apresentada uma revisão bibliográfica sobre os trabalhos com temas correlacionados a redes de colaboração científica, por meio da qual é possível verificar a evolução dos estudos sobre redes de colaboração científica. Por fim, os trabalhos correlatos foram identificados e comparados a fim de identificar pontos de melhoria.

3. Desenvolvimento do projeto

Neste capítulo são apresentados os algoritmos bibliométricos otimizados para suportarem o processamento distribuído e a ferramenta de extração de indicadores bibliométricos. Inicialmente, os dados são importados, após isso, as tecnologias utilizadas são definidas, posteriormente, os algoritmos bibliométricos são adaptados para suportarem o processamento distribuído e, por fim, a ferramenta é desenvolvida com suporte aos algoritmos desenvolvidos e a filtros para refinar a extração de conhecimento.

3.1. Material e métodos

Nesta seção, é apresentada a sequência de passos realizados para desenvolvimento do trabalho. De forma a facilitar a compreensão, os principais componentes e tecnologias utilizadas são exibidos na Tabela 3 e explicados com mais detalhes a seguir.

Inicialmente, o importador construído em Valencio et al. (2017) foi melhorado de modo a contemplar também a importação de currículos de arquivos XML e JSON, além de outros formatos compatíveis com SGBDs relacionais. Com o importador finalizado, os currículos dos pesquisadores da Universidade Estadual Paulista “Júlio de Mesquita Filho” (UNESP) foram importados da Plataforma Lattes para posteriormente serem utilizados nos testes.

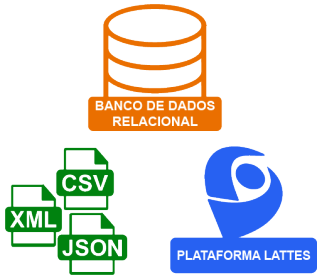
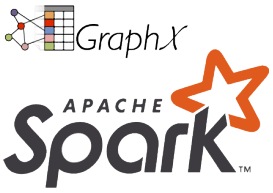
O SGBD orientado a grafos utilizado é o Neo4j, um SGBD *Not Only SQL* (NoSQL), distribuído e escalável. Sua escolha foi baseada em pesquisas que o destacam em relação a outros SGBDs da mesma categoria, por sua alta performance, robustez e escalabilidade (BATRA; TYAGI, 2012; JAISWAL; AGRAWAL, 2013).

Com o objetivo de auxiliar na validação, todos os algoritmos das bibliométricas desenvolvidas em Valencio et al. (2017) foram reproduzidos.

Posteriormente, os algoritmos de extração de bibliométricas foram distribuídos por meio do framework Apache Spark e da API de acesso a grafos Apache GraphX. Com o objetivo de obter o melhor desempenho, todos os algoritmos foram desenvolvidos em Scala, linguagem de programação nativa da plataforma Apache Spark.

Por fim, os filtros desenvolvidos em Valencio et al. (2017) foram atualizados para serem compatíveis com os algoritmos distribuídos e a exibição das bibliométricas foi desenvolvida por meio da biblioteca JavaScript D3 (D3JS, 2019). A interface do usuário foi desenvolvida por meio de HTML5, CSS3 e JavaScript ES6.

Tabela 3 – estrutura do trabalho

1 – Importador	2 – Banco de dados unificado	3 – Algoritmos otimizados
		

Fonte: Elaborado pelo autor

3.2. Distribuição das bibliométricas

Com o objetivo de suportar a extração de conhecimento de grandes redes de colaboração científica, todos os doze algoritmos de extração de bibliométricas descritos na seção 2.2 foram desenvolvidos para suportar o processamento distribuído de dados por meio da plataforma Apache Spark e a API Apache GraphX.

Para isso, as redes de colaboração científica, normalmente representadas por grafos não dirigidos, foram adaptadas para um grafo dirigido, por conta do suporte da API apenas a esse formato. Para esse fim, foram adicionadas duas arestas dirigidas para representar uma aresta não dirigida, sendo que ambas representam o mesmo trabalho e incidem sobre os mesmos nós, ou seja, os pesquisadores, mas em direções opostas. Dessa forma, o grafo sempre terá o dobro de arestas que o grafo da rede de colaboração original, o que deve ser considerado para o cálculo de todas as bibliométricas.

Os passos para desenvolvimento das bibliométricas são detalhados nesta seção. Destaca-se que foram utilizadas as operações explicadas na seção 2.3.4, por conta da maneira automática com a qual seu processamento distribuído é realizado.

3.2.1. Bibliométricas com baixa complexidade de implementação

Das doze bibliométricas apresentadas na seção 2.2 e resumidas na Tabela 1, oito delas contêm algoritmos de extração menos complexos que serão detalhados nesta seção, a saber: total de pesquisadores, total de coautorias, total de trabalhos em coautoria, total de coautores, média de trabalhos em coautorias, densidade das coautorias, tamanho do maior grupo de coautores e porcentagem do maior grupo de coautores. Para o desenvolvimento, foram utilizadas propriedades e funções do grafo, gerenciado pelo Apache GraphX. As propriedades utilizadas para cada bibliométrica são apresentadas na Tabela 4 e detalhadas a seguir:

Tabela 4 – propriedades utilizadas para implementação de cada bibliométrica

Bibliométrica	Implementação
Total de pesquisadores	numVertices
Total de coautorias	collectNeighborIds(EdgeDirection.Either).map().sum() / 2
Total de trabalhos em coautoria	numEdges / 2
Total de coautores	Degrees.filter().count()
Média de trabalhos em coautoria	Total de trabalhos em coautoria * 2 / Total de pesquisadores
Densidade	Total de trabalhos em coautoria / Máx. de arestas possível
Maior grupo de coautores	Max(connectedComponents().countByValue())
Porcentagem do maior grupo	100 * Maior grupo de coautores / Total de pesquisadores

Fonte: Elaborado pelo autor

Para o desenvolvimento do algoritmo de extração do total de pesquisadores foi utilizado uma propriedade do grafo, denominada *numVertices*, que armazena a quantidade total de vértices no grafo e, por consequência, o número total de pesquisadores da rede de colaboração.

Para o desenvolvimento do algoritmo de extração do total de coautorias foi extraído do grafo um mapa chave-valor do tipo (<vértice>, <vértices vizinhos>). Uma vez calculado esse mapa, é criada uma lista com a quantidade de vizinhos distintos de cada vértice, ou seja, a quantidade de coautorias de cada pesquisador. Por fim, é calculada a somatória total desses valores, o valor obtido é divido por 2 e resulta no valor dessa bibliométrica.

Para o desenvolvimento do algoritmo de extração do total de trabalhos em coautoria foi utilizado uma propriedade do grafo, denominada *numEdges*, que armazena a quantidade total de arestas no grafo. O valor armazenado por essa propriedade foi dividido por 2, por conta

da adaptação realizada no grafo para representar a rede de colaboração, detalhada na seção 3.2. O resultado dessa divisão é o valor dessa bibliométrica.

Para o desenvolvimento do algoritmo de extração do total de coautores é extraído do grafo um mapa chave-valor do tipo (<vértice>, <grau do vértice>). Esse mapa é filtrado para manter apenas os itens com grau maior que zero. Por fim, é calculada a quantidade de itens no mapa, que é também o valor dessa bibliométrica.

Para o desenvolvimento do algoritmo de extração da média de trabalhos em coautoria foi utilizado o número total de trabalhos em coautorias multiplicado por dois e divido pelo número de pesquisadores, que resulta no valor dessa bibliométrica.

Para o desenvolvimento do algoritmo de extração da densidade das coautorias, foi utilizada o valor do número de coautorias. Além disso, foi utilizado o valor do número máximo de coautorias possível para a quantidade de pesquisadores na rede de colaboração científica, calculado por meio da Equação 8, em que V é o número de pesquisadores da rede de colaboração. Após obtido ambos os valores, o número de coautorias foi dividido pela quantidade máxima de pesquisadores possíveis, resultando no valor dessa bibliométrica.

$$E = \frac{|V| * (|V| - 1)}{2} \tag{8}$$

Para o desenvolvimento do algoritmo de extração do tamanho do maior grupo de coautores é extraído do grafo um mapa chave-valor do tipo (<vértice>, <código do componente>), em que o código do componente representa o grupo que o pesquisador está. Após isso, é calculado um mapa chave-valor do tipo (<código do componente>, <quantidade>), ou seja, a quantidade de pesquisadores por grupo. Por fim, extrai-se o componente com a maior quantidade de pesquisadores, que representa o tamanho do maior grupo de coautores.

Para o desenvolvimento do algoritmo de extração da porcentagem do maior grupo de coautores, o valor do tamanho do maior grupo de coautores é dividido pelo número total de pesquisadores. Após isso, o resultado da divisão é multiplicado por 100 para obtenção do valor em porcentagem dessa bibliométrica.

Nas próximas seções serão detalhadas as outras quatro bibliométricas.

3.2.2. Assortatividade

O algoritmo para a extração da assortatividade desenvolvido pode ser visualizado no Algoritmo 1 e é explicado de maneira detalhada a seguir.

Algoritmo 1 – Algoritmo para processamento distribuído do cálculo da assortatividade

```
1.  Declara: mapaGrauDosNos, variaveisAssortatividade, m, variaveisAssortatividade3,
    assortatividade
2.  Função calculaAssortatividade(grafo)
3.      mapaGrauDosNos = grafo.degrees.collectAsMap()
4.      variaveisAssortatividade = graph.triplets.map(mapeiaTriplets).reduce(reduzTriplets)
5.      m = Math.pow(grafo.numEdges, -1)
6.      variaveisAssortatividade3 = Math.pow(m * variaveisAssortatividade._3, 2)
7.      val assortatividade = ((m * variaveisAssortatividade._1) - variaveisAssortatividade3)
    / ((m * variaveisAssortatividade._2) - variaveisAssortatividade3)
8.      Retorna assortatividade
9.  Fim da função

10. Função mapeiaTriplets(triplet)
11.     grauVerticeOrigem = mapaGrauDosNos.get(triplet.srcId).get
12.     grauVerticeDestino = mapaGrauDosNos.get(triplet.srcId).get
13.     Retorna (
            (grauVerticeOrigem * grauVerticeDestino),
            (Math.pow(grauVerticeOrigem, 2) + Math.pow(grauVerticeDestino, 2)) / 2,
            (grauVerticeOrigem + grauVerticeDestino) / 2
        )
14. Fim da função

15. Função reduzTriplets(tripletsTupleA, tripletsTupleB)
16.     Retorna (
            tripletsTupleA._1 + tripletsTupleB._1,
            tripletsTupleA._2 + tripletsTupleB._2,
            tripletsTupleA._3 + tripletsTupleB._3,
        )
17. Fim da função
```

Inicialmente, na linha 1, são declaradas as variáveis utilizadas no decorrer do algoritmo. Posteriormente, na linha 2, é declarada a função para cálculo da assortatividade, que recebe como parâmetro o grafo da rede de colaboração científica, objeto do estudo quantitativo. Adiante, na linha 3, é utilizada a propriedade *degrees* para extrair um mapa chave-valor do tipo (<vértice>, <grau do vértice>) que contém o grau de todos os vértices do grafo.

Em seguida, na linha 4, os *triplets* do grafo, ou seja, todos os trios formados por vértices de origem, aresta e vértice de destino são extraídos do grafo, mapeados por meio da

função *map(f)*, que recebe como parâmetro a função *mapeiaTriplets* e retorna um conjunto de tuplas, e reduzido por meio da função *reduce(f)* que recebe como parâmetro a função *reduzTriplets* e retorna a somatória do desse conjunto de tuplas item a item, ou seja, o primeiro item da primeira tupla é acumulado ao primeiro item da segunda tupla, o segundo item da primeira tupla é acumulado ao item da segunda tupla, e assim por adiante até finalizar o conjunto de tuplas. A tupla retornada é armazenada na variável *variaveisAssortatividade*.

Para melhor entendimento, a explicação sequencial linha a linha do algoritmo será interrompida de forma a priorizar a explicação das funções *mapeiaTriplets* e *reduzTriplets*, com o objetivo de facilitar o entendimento.

A função *mapeiaTriplets* é declarada na linha 10 e recebe como parâmetro a *triplet* a ser mapeada. Posteriormente, nas linhas 11 e 12, o grau dos vértices de origem e destino são extraídos e armazenados, respectivamente, nas variáveis *grauVerticeOrigem* e *grauVerticeDestino*. Adiante, na linha 13, é retornada uma tupla com três colunas, que representam os três cálculos realizados pelos itens das somatórias contidas na equação de extração da assortatividade, exibida na Equação 6. Dessa forma, o primeiro item da tupla representa $j * k$, o segundo item representa $\frac{1}{2} * (j^2 + k^2)$ e o terceiro e último item representa $\frac{1}{2} * (j + k)$.

A função *reduzTriplets* é declarada na linha 15 e recebe como parâmetro duas tuplas, idênticas às retornadas pela função *mapeiaTriplets*. Posteriormente, retorna uma tupla, também idêntica ao retorno da função *mapeiaTriplets*, mas com a soma de ambas as tuplas recebidas. Essa função, quando passada para a função *reduce(f)*, realiza o somatório de todas as tuplas item a item.

Explicadas as funções auxiliares, será retomada a explicação linha a linha do algoritmo. Na linha 5, é calculado o valor de *m*, que é quantidade de arestas do grafo elevado a -1, que representa o item M^{-1} da Equação 6. Em seguida, o terceiro item da tupla é multiplicado por m e elevado ao quadrado. Por fim, tendo a disposição todos os itens utilizados no cálculo da assortatividade, seu valor é calculado na linha 7 e retornado na linha 8. Destaca-se que as funções de *map* e *reduce* são executadas automaticamente de forma distribuída.

Após a explicação detalhada do algoritmo de assortatividade, será explicado agora o algoritmo de transitividade.

3.2.3. Transitividade

O algoritmo para a extração da transitividade desenvolvido pode ser visualizado no Algoritmo 2 e é explicado de maneira detalhada a seguir.

Algoritmo 2 – Algoritmo para processamento distribuído do cálculo da transitividade

```
1.  Declara: vizinhos, triângulos, vizinhosETriangulos, transitividade
2.  Função calculaTransitividade (grafo)
3.      vizinhos = grafo.collectNeighbors()
4.      triangulos = grafo.triangleCount().vertices
5.      vizinhosETriangulos = triangulos.join.vizinhos
6.      transitividade = vizinhosETriangulos
            .flatMap(calculaTransitividadeDoVertice)
            .mean()
7.  Fim da função

8.  Função calculaTransitividadeDoVertice(vizinhosETriangulos)
9.      quantidadeDeVizinhos = vizinhosETriangulos._1.distinct.length
10.     Se (quantidadeDeVizinhos > 1)  Retorna Some(vizinhosETriangulos._2 /
    nMaxCoautoriasPossivel(quantidadeDeVizinhos))
11.     Senão Retorna None;
12. Fim da função

13. Função nMaxCoautoriasPossivel(qtddDeVertices)
14.     Retorna qtddDeVertices * (qtddDeVertices-1) / 2
15. Fim da função
```

Inicialmente, na linha 1, são declaradas as variáveis utilizadas no decorrer do algoritmo. Posteriormente, na linha 2, é declarada a função para cálculo da transitividade, que recebe como parâmetro o grafo da rede de colaboração científica, objeto do estudo quantitativo. Adiante, na linha 3, é utilizada a função do grafo *collectNeighbors()* para extrair um mapa chave-valor do tipo (<vértice>, <vértices vizinhos>) que contém uma coleção com todos os vizinhos de cada vértice do grafo. Em seguida, na linha 4, é utilizada a função do grafo *triangleCount()* para extrair um mapa chave-valor do tipo (<vértice>, <quantidade de triângulos>) que contém a quantidade de triângulos de cada vértice, ou seja, a quantidade de vizinhos dois a dois que também são ligados entre si. Ambos os mapas são atribuídos às variáveis *vizinhos* e *triangulos*, respectivamente.

Subsequentemente, na linha 5, os dois mapas extraídos anteriormente são agrupados com base em suas chaves do tipo <vértice>, para isso, a função *join(mapa)* do mapa é utilizada, recebendo como parâmetro o mapa a ser agrupado e retorna um mapa do tipo

(<vértice>, <(vértices vizinhos, quantidade de triângulos)>). O valor retornado é atribuído à variável *vizinhosETringulos*.

Sucessivamente, na linha 6, a transitividade é calculada, para isso, o mapa atribuído à variável *vizinhosETringulos* é mapeado por meio da função *flatMap(f)*, que recebe como parâmetro a função *calculaTransitividadeDoVertice* e retorna o valor da transitividade de cada vértice. Por conta de ser um *flatMap*, apenas os valores válidos para a transitividade são mapeados, ou seja, apenas os vértices com mais que 1 vizinho tem seus valores de transitividade mapeados. Por fim, calcula-se a média das transitividades válidas de cada vértice, que resulta no valor da transitividade.

Posteriormente, na linha 8, é declarada a função *calculaTransitividadeDoVertice*, que recebe como parâmetro uma tupla do tipo (vértices vizinhos, quantidade de triângulos) e retorna uma *Option*, que pode ser de dois tipos: *Some* e *None*. Se o retorno for do tipo *Some*, o valor retornado é adicionado na nova coleção retornada pelo *flatMap*, por outro lado, se for do tipo *None*, o valor é descartado. Adiante, na linha 9, é extraída a quantidade de vértices vizinhos distintos contidos na coleção do primeiro item da tupla recebida como parâmetro (vértices vizinhos) e esse valor é atribuída à variável *quantidadeDeVizinhos*.

Em seguida, na linha 10, é verificado se a quantidade de vértices vizinhos, é maior que 1, se for, retorna um *Option* do tipo *Some*, com o valor da transitividade do vértice, se não, na linha 11, é retornado um *Option* do tipo *None*. Para realizar o cálculo da transitividade, ainda na linha 10, o valor do segundo item da tupla recebida como parâmetro, que representa a quantidade de triângulos, é dividido pela quantidade máxima de coautoria possível, calculada pela função *nMaxCoautoriasPossivel(quantidadeDeVizinhos)* que recebe como parâmetro a quantidade de vizinhos.

Por fim, na linha 13 é declarada a função *nMaxCoautoriasPossivel*, que recebe como parâmetro a quantidade de vértices do grafo e retorna o cálculo da quantidade máxima de coautorias possíveis em um grafo com a quantidade de vértices recebida. Seu cálculo é efetuado na linha 14, por meio da Equação 9.

$$C = \frac{quantidadeDeVertices * (quantidadeDeVertices - 1)}{2} \qquad (9)$$

Após explicação detalhada do algoritmo de transitividade, será explicado agora o algoritmo para cálculo do caminho médio e diâmetro.

3.2.4. Caminho médio e diâmetro

Para a implementação dessa bibliométrica foi utilizado o Pregel API, explicado na seção 2.3.5, para realizar o cálculo de todos os caminhos mínimos, utilizados para o cálculo do caminho médio e diâmetro. Para invocar o Pregel API é necessário passar como parâmetro três funções, responsáveis por: atualizar valor do vértice, enviar mensagens para os vértices vizinhos e mesclar mensagens recebidas. A implementação das funções utilizadas para invocar o Pregel API para realizar esse cálculo, é exibida no Algoritmo 3, com um algoritmo de mais alto nível para facilitar o entendimento, e detalhada a seguir. Destaca-se que o valor do vértice é inicializado com um mapa chave-valor do tipo (<vértice>, <distância mínima>), com o valor de distância mínima igual a 0 para ele mesmo.

Algoritmo 3 – Caminho médio e diâmetro com o Pregel API

```
1.  Função atualizarVertice (vertice, valorAtualMapaDoVertice, novosValoresMapa)
2.      mapa = Map[vertice, distanciaMinima]()
3.      Se for o primeiro super step
4.          mapa = valorAtualMapaDoVertice
5.      Senão
6.          Para cada chave do novosValoresMapa que não existir ou tiver valor menor que o
        valor da mesma chave no valorAtualMapaDoVertice, adiciona a chave com seu valor
        acrescido de 1 ao valorAtualMapaDoVertice e ao mapa.
7.      Fim Se
8.      Retorna o mapa
9.  Fim da função

10. Função enviarMensagem(triplets)
11.     Se o mapa não estiver vazio, o envia para seus vizinhos
12. Fim da função

13. Função mesclarMensagens(a, b)
14.     Se receber mais que um mapa, mantém as chaves iguais com menores valores
15. Fim da função
```

Inicialmente, na linha 1 é declarada a função *atualizarVertice*, responsável por atualizar o valor do vértice, que recebe três parâmetros, a saber: o vértice que terá o valor atualizado, o *valorAtualMapaDoVertice*, que é um mapa chave-valor do tipo (<vértice>, <distância mínima>) que representa o valor do vértice no *super step* e que será atualizado e o *novosValoresMapa*, que são os valores atualizados no *super step* anterior.

Posteriormente, na linha 2, é instanciado um mapa do tipo (<vértice>, <distância mínima>), que armazenará os dados atualizados no *super step*. Em seguida, na linha 3, é verificado se é o primeiro *super step* e, caso seja o primeiro *super step*, será executada a linha 4, em que a variável *mapa* será atribuída com o valor da variável *valorAtualMapaDoVertice*. Por outro lado, se não for o primeiro *super step*, será executada a linha 6, em que para cada vértice contido na variável *novosValoresMapa*, será verificado se também está contido na variável *valorAtualMapaDoVertice*, ou, caso esteja contido, se o valor do <caminho mínimo> contido no valor da chave da variável *novosValoresMapa* é menor que o valor da mesma chave contido na variável *valorAtualMapaDoVertice*, se for adiciona a chave com seu valor acrescido de 1 ao *valorAtualMapaDoVertice* e ao *mapa*. Por fim, retorna a variável *mapa* para ser utilizada pela função *enviaMensagem*.

Em seguida, na linha 10, é declarada a função *enviarMensagem*, que recebe como parâmetro uma *triplet*, ou seja, os trios formados por vértices de origem, aresta e vértice de destino, semelhante à utilizada no cálculo da assortatividade, em que o mapa retornado pela função *atualizarVertice*, se não estiver vazio, é enviado do vértice de origem para o vértice de destino.

Por fim, na linha 13, é declarada a função *mesclarMensagem*, que recebe como parâmetro duas mensagens enviadas pela função *enviarMensagem* para o mesmo vértice de destino e as mescla, priorizando chaves com valores menores.

A execução do Pregel invocado com as três funções exibidas no Algoritmo 3 enviadas com parâmetro, retorna um mapa chave-valor do tipo (<vértice>, (<vértice>,<distância mínima>)) que contém um mapa para todos os vértices atingíveis pelo vértice e a respectiva distância mínima. De posse desses dados, é calculada a média de todos os caminhos médios para obtenção do valor do caminho médio da rede de colaboração e extraído o maior caminho médio para obtenção do valor do diâmetro.

Dessa forma, todas as otimizações realizadas nas bibliométricas foram detalhadas. Na próxima seção será explicitado detalhes a respeito da ferramenta obtida como subproduto desse trabalho.

3.3. Ferramenta de Extração de Indicadores Bibliométricos (FEIB)

Como subproduto, foi desenvolvida uma ferramenta denominada Ferramenta de Extração de Indicadores Bibliométricos (FEIB), para extração de conhecimentos bibliométricos

de redes de colaboração científicas. A ferramenta é composta pelos itens apresentados anteriormente na Tabela 3 e detalhados a seguir:

1) **importador de redes de colaboração científicas** – extrator de redes de colaboração científicas de informações acadêmicas com suporte para SGBD relacional, Plataforma Lattes e os formatos de arquivo: CSV, XML e JSON. Será avaliada a possibilidade de adição de outras fontes de informações científicas;
2) **banco de dados unificado orientado a grafos** – todas as informações acadêmicas serão armazenadas em um SGBD orientado a grafos, tendo em vista que este é o suporte mais adequado e compatível ao domínio a que pertence o problema alvo, ou seja, as redes de colaboração científicas (BATRA; TYAGI, 2012; JAISWAL; AGRAWAL, 2013);
3) **algoritmos distribuídos para extração de bibliométricas** – com o objetivo de alcançar a escalabilidade necessária para o tratamento de grandes redes de colaboração, foram utilizados os algoritmos para a extração das bibliométricas otimizados, como parte dos objetivos desse trabalho.

Além disso, o funcionamento da FEIB é o seguinte:

a) o usuário interessado em extrair os indicadores bibliométricos aplica filtros na rede de colaboração para restringi-la;
b) os algoritmos distribuídos para extração de bibliométricas processam com base na rede de colaboração filtrada;
c) por fim, a ferramenta exibe as bibliométricas e a rede de colaboração para o usuário.

3.3.1. Importador

Para desenvolvimento do importador, foi utilizado o framework Spring Data JPA (SPRING, 2019), o qual gera uma interface para comunicação com os bancos de dados relacionais e permite a alteração da implementação do banco de dados relacional sem alterações na aplicação.

O importador suporta importar os dados, desde que estruturados, de SGBD relacional e os formatos de arquivo: CSV, XML e JSON. Além disso, suporta a importação por meio da Plataforma Lattes, no qual é utilizado o importador desenvolvido no SASD/Lattes (GBD, 2019). Sua escolha se deve ao fato de toda a UNESP, inclusive a Pró-reitoria de pesquisa

da UNESP (PROPe) o utilizar como ferramenta de apoio no levantamento de informações sobre sua produção acadêmica e científica. O que configura o SASD/Lattes uma ferramenta de relevância para a aquisição de dados confiáveis e consistentes para serem utilizados nesse trabalho.

Todas as redes de colaboração importadas são armazenadas da seguinte forma: nome e grafo. Dessa forma, é possível realizar a extração de bibliométricas de mais que uma rede de colaboração, sem ser necessário realizar uma nova importação para alterá-la.

3.3.2. Banco de dados orientado a grafos

O SGBD utilizado foi o Neo4j – que consiste em um gerenciador NoSQL, distribuído e escalável e permite o armazenamento e recuperação de informações por meio do modelo de grafos. A escolha do Neo4j foi feita com base no fato das pesquisas recentes terem demonstrado sua eficiência para o tratamento de dados semelhantes aos de redes de colaboração (BATRA, TYAGI, 2012), mas outros bancos orientados a grafos também podem ser utilizados.

3.3.3. Extração de bibliométricas

Para a extração de bibliométricas, é possível escolher em qual das redes de colaboração já importadas serão extraídos os dados. Além disso, é possível aplicar filtro de período (ano inicial e ano final). Por fim, as bibliométricas são extraídas por meio da API desenvolvida em Apache Spark e exibidas para o usuário, o qual poderá comparar as duas últimas bibliométricas extraídas por meio do gráfico radial que exibe as bibliométricas.

3.3.4. Exibição das bibliométricas

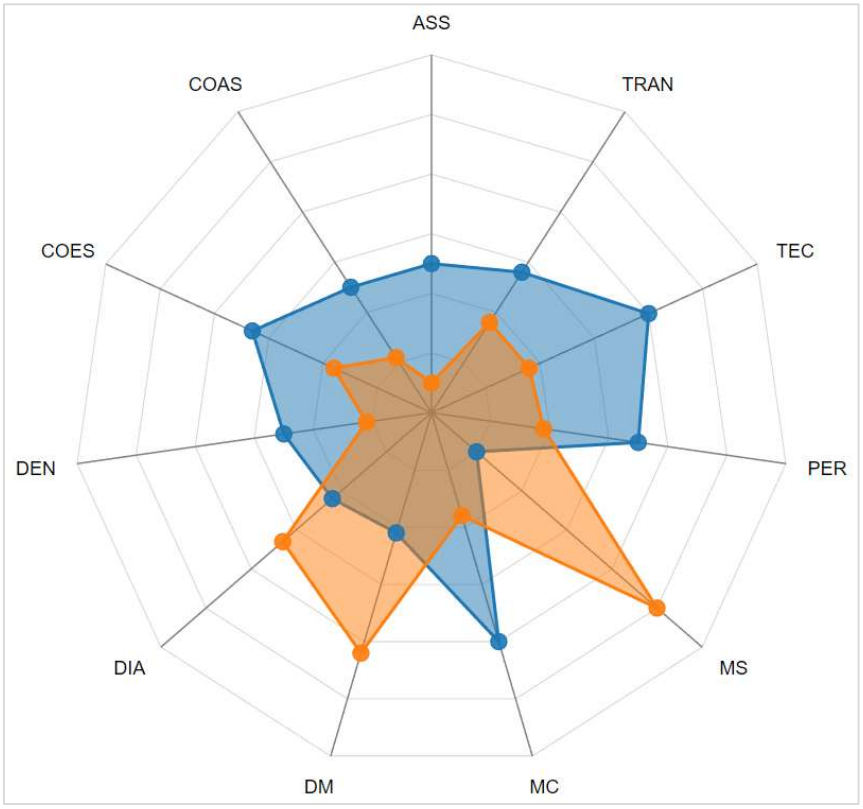
Para exibição das bibliométricas e possibilitar a comparação entre diferentes redes de colaboração, foi desenvolvido o gráfico radial demonstrado na Figura 2. Nele, é possível comparar as bibliométricas extraídas entre duas ou mais redes de colaboração com as seguintes características:

a) as extrações de bibliométricas de diferentes redes de colaboração são representadas por cores distintas;
b) o valor da bibliométrica é representado pela posição do círculo no eixo, em que quanto mais próximo do centro menor é o valor da bibliométrica e quanto mais distante do centro maior é seu valor.

c) cada eixo está vinculado a apenas uma bibliométrica.

A legenda do gráfico da Figura 2 é a seguinte: Assortatividade (ASS); Coautorias (COAS); Coautores (COES); Densidade (DEN); Diâmetro (DIA); Distância Média (DM); Média de Coautorias (MC); Maior Subgrafo (MS); Percentual do Maior Subgrafo (PER); Trabalhos em Coautoria (TEC); Transitividade (TRAN).

Figura 2 – exibição comparativa de duas redes hipotéticas, a de cor azul e a outra, de cor laranja



Fonte: Elaborado pelo autor

No exemplo, as bibliométricas assortatividade, transitividade, trabalhos em coautoria, percentual do maior subgrafo, média de coautorias, densidade, coautores e coautorias da rede de colaboração representada pela cor azul são maiores que as da representada pela cor laranja. Por outro lado, as bibliométricas maior subgrafo, distância média e diâmetro da rede de colaboração representada pela cor laranja são maiores que as da representada pela cor azul. Além disso, ao posicionar o cursor do mouse sobre cada círculo dos eixos, é possível verificar o valor exato da bibliométrica representada por ele.

3.4. Considerações finais

Nesta seção foram detalhados os algoritmos de extração das bibliométricas, otimizados por esse trabalho. Além disso, foi detalhada a ferramenta de extração de indicadores bibliométricos, cujo propósito é facilitar o uso dos algoritmos otimizados. Na próxima seção, são apresentados os testes para os algoritmos otimizados com o objetivo de confirmar as melhorias obtidas.

4. Avaliação Experimental

Neste capítulo são descritos os testes realizados, os quais foram separados em quatro categorias, a fim de avaliar diferentes características do algoritmo, são elas:

a) testes de corretude, com o objetivo de comprovar que as bibliométricas extraídas pelo algoritmo são corretas;
b) estudo de caso, com o objetivo de realizar o estudo, na prática, de uma grande comunidade científica, e verificar a aplicabilidade da ferramenta desenvolvida;
c) testes de desempenho, com o objetivo de explicitar as melhorias dos algoritmos em comparação com suas antigas versões encontradas na literatura;
d) testes de escalabilidade, com o objetivo de demonstrar que os algoritmos comportam o processamento de grandes redes de colaboração científica.

4.1. Configurações da plataforma de testes

Os testes de corretude, estudo de caso e desempenho foram realizados em um computador pessoal com as seguintes configurações de hardware:

a) placa mãe Gigabyte Z170X-Gaming 3;
b) processador Intel Core i7 – 6700k com 4.0 GHz de frequência (4.2 GHz com a tecnologia Turbo Boost) e 4 núcleos físicos;
c) dois pentes de 8 Gigabytes de memória RAM DDR4 com 2400 MHz de frequência;
d) placa gráfica Geforce GTX 970 G1 Gaming;
e) 120 Gigabytes de SSD – SANDISK SDSSDA 120G;

f) 1 Terabyte de HDD com 7200RPM – Seagate ST1000DM003.

Além disso, continha as seguintes configurações de software:

g) Windows 10;

h) Java 8;

i) Apache Tomcat 8.

Os testes de escalabilidade foram executados na Amazon EMR (AMAZON, 2019), que é uma plataforma de cluster de computadores gerenciada para a execução de estruturas de Big Data, como o Apache Hadoop e o Apache Spark, para processar e analisar grandes quantidades de dados. As instâncias dos computadores utilizados foram da categoria *m5.xlarge* com as seguintes configurações:

a) 1ª ou a 2ª geração do processador Intel Xeon Platinum série 8000 (Skylake-SP ou Cascade Lake) com até 3,1 GHz: 4 vCPU;

b) 16 GB de memória RAM;

c) Armazenamento EBS de 64 GBs.

4.2. Teste de corretude

Os testes de corretude do algoritmo têm o objetivo de verificar se os resultados processados estão corretos. Para isso, foram utilizadas duas abordagens e cada uma utilizou uma rede de colaboração científica distinta para a execução dos testes:

a) teste de mesa;

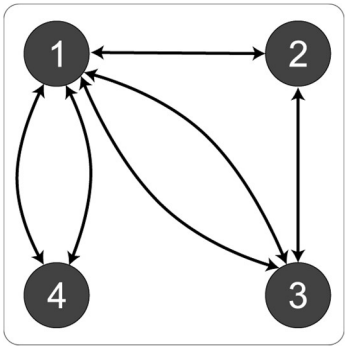
b) comparação dos resultados com trabalhos correlatos.

Para o teste de mesa, a rede de colaboração, ilustrada na Figura 3, foi processada com o auxílio da ferramenta, a fim de que sejam extraídas as respectivas bibliométricas. Após isso, os resultados foram verificados mediante a extração das bibliométricas de forma manual.

Por outro lado, para os testes de comparação de resultados com os trabalhos correlatos, foi processada a rede de colaboração, ilustrada na Figura 4, e suas bibliométricas foram extraídas tanto por meio da ferramenta desenvolvida em Valencio et al. (2017), quanto pelos algoritmos propostos por esse trabalho. Ao final, os resultados foram comparados a fim de constatar que ambos extraíram os mesmos valores para as mesmas bibliométricas.

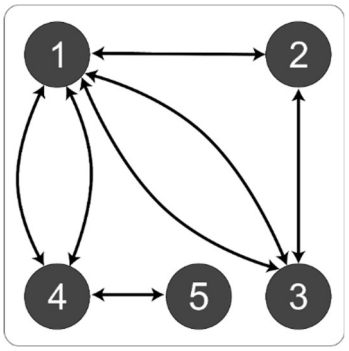
Destaca-se que o trabalho correlato da literatura e utilizado como parâmetro a fim de confrontar os testes de corretude e de desempenho, foi o escolhido em Valencio et al. (2017), pois o mesmo foi utilizado em testes comparativos com os principais trabalhos da literatura e apresentou melhor desempenho em todos as comparações efetivadas.

Figura 3 – rede de colaboração criada para o teste de mesa



Fonte: Elaborada pelo autor.

Figura 4 – rede de colaboração para teste de corretude de comparação de resultados



Fonte: Elaborada pelo autor.

4.2.1. Teste de mesa

A rede de colaboração, ilustrada na Figura 3, foi processada e suas bibliométricas foram extraídas de duas formas: manualmente e pela ferramenta. Na forma manual, as bibliométricas da rede de colaboração foram extraídas por meio de cálculos manuais utilizando as equações apresentadas na seção 2.2 e os resultados são exibidos na coluna “Extração manual” da Tabela 5. Por outro lado, na extração pela ferramenta, a rede de colaboração foi processada pela ferramenta desenvolvida e as bibliométricas extraídas são apresentadas na coluna “Ferramenta desenvolvida” da Tabela 5.

De forma a validar a corretude dos dados extraídos pela ferramenta, comparou-se os resultados extraídos de forma manual e os resultados extraídos pela ferramenta desenvolvida, e foi constatado que todos os resultados extraídos são idênticos, o que assegura a corretude dos resultados extraídos pelos algoritmos da ferramenta desenvolvida.

Tabela 5 – valores extraídos no teste de mesa

Bibliométrica	Extração manual	Ferramenta desenvolvida
Total de pesquisadores	4	4
Total de trabalhos em coautorias	6	6
Total de coautorias	4	4
Total de coautores	4	4
Média de trabalhos em coautoria	3	3
Densidade das coautorias	0,667	0,667
Assortatividade	-0,714	-0,714
Transitividade	0,778	0,778
Distância média entre os pesquisadores	1,333	1,333
Diâmetro	2	2
Tamanho do maior grupo de coautores	4	4
Porcentagem do maior grupo de coautores	100%	100%

Fonte: Elaborado pelo autor

4.2.2. Comparação de resultados

Para confirmar a corretude dos dados da ferramenta, foram extraídas as bibliométricas de uma pequena rede de colaboração, ilustrada na Figura 4, por meio da ferramenta desenvolvida em Valencio et al. (2017) e por meio da ferramenta desenvolvida nesse trabalho, para então comparar os resultados de ambas as extrações.

Os resultados extraídos de ambos os trabalhos são exibidos na Tabela 6, no qual constata-se que ambas as ferramentas extraem os mesmos valores para cada bibliométrica para conjuntos de dados de entrada iguais. Desta forma, constata-se a corretude dos algoritmos desenvolvidos por esse trabalho.

Tabela 6 – valores extraídos nas duas ferramentas

Bibliométrica	Valencio et al. (2017)	Esse trabalho
Total de pesquisadores	5	5
Total de trabalhos em coautorias	7	7
Total de coautorias	5	5
Total de coautores	5	5
Média de trabalhos em coautoria	2,8	2,8
Densidade das coautorias	0,5	0,5
Assortatividade	-0,28	-0.28
Transitividade	0,58	0,58
Distância média entre os pesquisadores	1,7	1,7
Diâmetro	3	3
Tamanho do maior grupo de coautores	5	5
Porcentagem do maior grupo de coautores	100%	100%

Fonte: Elaborado pelo autor

4.3. Caso de uso

A fim de enfatizar a importância da ferramenta, foi realizado o estudo da rede de colaboração da Unesp, de forma geral e trienal. Para isso, a rede de colaboração da Unesp foi importada da Plataforma Lattes no dia 29 de janeiro de 2019. O caso de uso foi dividido em duas abordagens:

a) um estudo geral da universidade, no qual foram considerados todos os pesquisadores ativos e trabalhos desenvolvidos por eles;
b) um estudo trienal, que mostra a evolutiva dos trabalhos nos triênios de 1990~1992 até 2017~2019. Nesse teste, os filtros de períodos foram utilizados. Além disso, apenas os pesquisadores atualmente vinculados à Unesp foram considerados.

Os resultados de ambos os estudos são exibidos a seguir.

4.3.1. Unesp geral

Os resultados extraídos da rede de colaboração da Unesp são exibidos na Tabela 7. Tais resultados revelam que os 3.764 pesquisadores produziram 335.088 trabalhos em 35.116 coautorias (sendo que esses trabalhos podem conter trabalhos repetidos), uma média de 178,05 trabalhos por pesquisador. Dentre os 3.764 pesquisadores, 3.644 participaram de pelo menos uma coautoria, o que explicita a importância da pesquisa em colaboração. A assortatividade relativamente alta de 0,56 demonstra que a rede de colaboração é homogenia, ou seja, não há pesquisadores que concentram a produtividade. A transitividade de 0,34 demonstra que a rede é pouco compacta. A distância média de 3,74 atende ao fenômeno de Milgram (1967), que diz que todas as pessoas estão conectadas a no máximo 6 passos uma da outra. O diâmetro 10 pode ser explicado pela baixa compactação da rede demonstrado pela transitividade. Por fim, dos 3.644 pesquisadores com ao menos 1 coautoria, apenas 3 não estão no maior grupo de coautores, que contém 3.641 pesquisadores, que representa 96,73% da rede de colaboração.

Tabela 7 – bibliométricas da rede de colaboração da UNESP

Bibliométrica	Valor extraído
Total de pesquisadores	3.764
Total de trabalhos em coautorias	335.088
Total de coautorias	35.116
Total de coautores	3.644
Média de trabalhos em coautoria	178,05
Densidade das coautorias	0,00496
Assortatividade	0,55545
Transitividade	0,33593
Distância média entre os pesquisadores	3,74842
Diâmetro	10
Tamanho do maior grupo de coautores	3.641
Porcentagem do maior grupo de coautores	96,73%

Fonte: Elaborado pelo autor

4.3.2. Unesp trienal

A evolução da rede de colaboração da Unesp pode ser observada pelo estudo trienal exibido na Tabela 8 e por meio dos gráficos comparativos ilustrados no Apêndice A. Por meio da tabela e dos gráficos, pode-se observar que as bibliométricas tendem a evoluir durante o tempo, entretanto, o último triênio não atende a essa característica de melhoria/aumento das bibliométricas, como pode ser verificado no gráfico de total de coautores, representado na Figura 14. Tal característica pode ser explicada por conta de o triênio de 2017~2019 ainda não ter sido finalizado até a data de importação dos dados utilizados por esse trabalho, ou seja, ainda poderão ser desenvolvidos novos trabalhos ou os trabalhos já desenvolvidos podem ainda não ter sido adicionados ao currículo Lattes dos pesquisadores, fonte em que os dados dessa rede de colaboração foram extraídos.

Tabela 8 – bibliométricas trienais da rede de colaboração da UNESP

Bibliométrica	1990 ~ 1992	1993 ~ 1995	1996 ~ 1998	1999 ~ 2001	2002 ~ 2004	2005 ~ 2007	2008 ~ 2010	2011 ~ 2013	2014 ~ 2016	2017 ~ 2019
Pesquisadores	3764	3764	3764	3764	3764	3764	3764	3764	3764	3764
Trabalhos em coautorias	2543	4913	9240	18058	30832	42118	59313	73593	65760	26690
Total de coautorias	820	1325	2221	3582	5364	7317	10023	14079	14750	8881
Coautores	666	957	1334	1781	2221	2578	2902	3222	3392	2980
Média de trabalhos em coautoria	1,351	2,611	4,91	8,595	16,38	22,38	31,52	39,10	34,94	14,18
Densidade das coautorias	1e-4	2e-4	3e-4	0,001	0,001	0,001	0,001	0,002	0,002	0,001
Assortatividade	0,676	0,655	0,784	0,404	0,525	0,538	0,609	0,669	0,626	0,641
Transitividade	0,548	0,484	0,478	0,46	0,438	0,408	0,419	0,408	0,397	0,425
Distância média	8,097	6,39	8,834	7,983	7,238	6,48	5,672	5,115	5,219	6,66
Diâmetro	23	20	27	26	24	20	15	13	14	18
Tamanho do maior subgrafo	179	301	868	1376	1901	2402	2811	3157	3334	2822
Porcentagem do maior subgrafo	5%	8%	23%	36%	51%	64%	75%	84%	89%	75%

Fonte: Elaborado pelo autor

4.4. Teste de desempenho

Para os testes de desempenho, os algoritmos de extração das bibliométricas de número total de pesquisadores, média de trabalhos em coautoria, densidade das coautorias, tamanho do maior grupo de coautores e porcentagem do maior grupo de pesquisadores não serão considerados. Isso se deve ao fato de que esses algoritmos têm tempo de execução irrelevante para as bases de dados utilizadas nesse teste.

Os testes utilizaram 7 amostras extraídas da rede de colaboração da Unesp, utilizada no teste de caso de uso. As amostras foram compostas com os seguintes valores:

a) G1 – 500 pesquisadores e 80.443 trabalhos;
b) G2 – 1.000 pesquisadores e 104.738 trabalhos;
c) G3 – 1.500 pesquisadores e 167.951 trabalhos;
d) G4 – 2.000 pesquisadores e 196.809 trabalhos;
e) G5 – 2.500 pesquisadores e 241.559 trabalhos;
f) G6 – 3.000 pesquisadores e 265.839 trabalhos;
g) G7 – 3.500 pesquisadores e 313.260 trabalhos.

Com o objetivo de obter consistência estatística, os algoritmos foram executados 5 vezes para cada amostra, seus tempos de execução foram extraídos e o tempo de execução final para a execução do algoritmo para a amostra foi a média dos tempos. Além disso, o desvio padrão foi extraído para explicitar o grau de dispersão dos tempos extraídos.

4.4.1. Transitividade

Os tempos em segundos extraídos das execuções do algoritmo de transitividade são exibidos na Tabela 9. Nela, são exibidos os valores de tempo de execução para cada amostra do algoritmo de transitividade da ferramenta desenvolvida em Valencio et al. (2017) e do algoritmo de transitividade da ferramenta desenvolvida nesse trabalho. Além disso, o desvio padrão demonstra que o grau de dispersão dos tempos extraídos é menor que 11% do valor da média dos tempos.

Tabela 9 – testes de desempenho do algoritmo de transitividade

Amostra	Ferramenta	Exec1 (s)	Exec2 (s)	Exec3 (s)	Exec4 (s)	Exec5 (s)	Média (s)	Desvio padrão
G7	Valencio et al. (2017)	13,05	11,98	11,97	11,67	11,84	12,10	0,55
	Esse trabalho	2,20	1,88	1,96	2,06	2,00	2,02	0,12
G6	Valencio et al. (2017)	11,41	10,25	10,33	10,52	10,30	10,56	0,49
	Esse trabalho	1,72	1,85	1,82	1,74	2,00	1,83	0,11
G5	Valencio et al. (2017)	10,47	9,35	9,38	9,70	9,46	9,67	0,46
	Esse trabalho	1,69	1,60	1,64	1,52	1,75	1,64	0,09
G4	Valencio et al. (2017)	8,78	7,43	7,14	7,25	7,09	7,54	0,71
	Esse trabalho	1,57	1,57	1,55	1,49	1,45	1,52	0,06
G3	Valencio et al. (2017)	6,40	6,19	5,88	6,01	6,19	6,13	0,20
	Esse trabalho	1,56	1,52	1,47	1,50	1,70	1,55	0,09
G2	Valencio et al. (2017)	3,32	2,95	2,76	2,73	2,74	2,90	0,25
	Esse trabalho	1,16	1,38	1,23	1,30	1,25	1,26	0,08
G1	Valencio et al. (2017)	2,58	2,15	2,05	2,04	2,09	2,18	0,23
	Esse trabalho	1,30	1,02	1,20	1,20	1,10	1,16	0,11

Fonte: Elaborado pelo autor

Os valores médios dos tempos em segundos das execuções para as diferentes amostras do algoritmo de transitividade são exibidos de forma simplificada na Tabela 10.

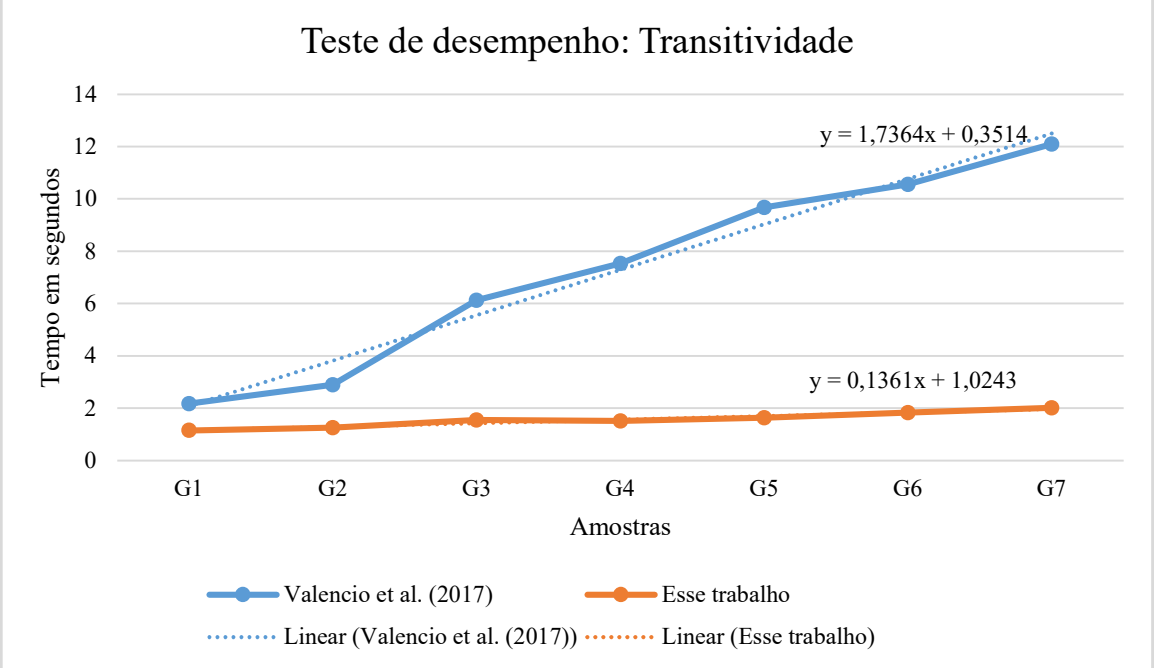
Tabela 10 – teste de desempenho do algoritmo de Transitividade

Amostra	Tempo (s)	
	Valencio et al. (2017)	Esse trabalho
G1	2,18	1,16
G2	2,9	1,26
G3	6,13	1,55
G4	7,54	1,52
G5	9,67	1,64
G6	10,56	1,83
G7	12,10	2,02

Fonte: Elaborado pelo autor

A comparação entre os testes de desempenho realizados sobre ambos os trabalhos é exibida no gráfico ilustrado na Figura 5.

Figura 5 – teste de desempenho do algoritmo de Transitividade



Fonte: Elaborado pelo autor

Por meio da variação do tempo entre as execuções do algoritmo de transitividade para diferentes amostras, extraiu-se as seguintes informações:

a) Valencio et al. (2017) – as variações de tempo para as diferentes amostras tendem para a Equação 10, com coeficiente angular constate de 1,7364;

b) esse trabalho – as variações de tempo para as diferentes amostras tendem para a Equação 11, com coeficiente angular constate de 0,1361.

$$y = 1{,}7364x + 0{,}3514 \quad (10)$$

$$y = 0{,}1361\text{x} + 1{,}0243 \quad (11)$$

Portanto, o algoritmo de transitividade implementado em Valencio et al. (2017) tem crescimento de tempo 12,76 vezes maior que o implementado nesse trabalho quando o tamanho do grafo tende ao infinito.

4.4.2. Número total de trabalhos em coautoria, número total de coautorias, número total de coautores e assortatividade

Os algoritmos para extração das bibliométricas número total de trabalhos em coautoria, número total de coautorias, número total de coautores e assortatividade também foram avaliados em relação ao desempenho.

Sobre o algoritmo de extração da bibliométrica número total de trabalhos em coautoria, os valores médios dos tempos em segundos das execuções para as diferentes amostras e os respectivos desvios padrões são exibidos na Tabela 11.

Tabela 11 – teste de desempenho do algoritmo de número total de trabalhos em coautoria

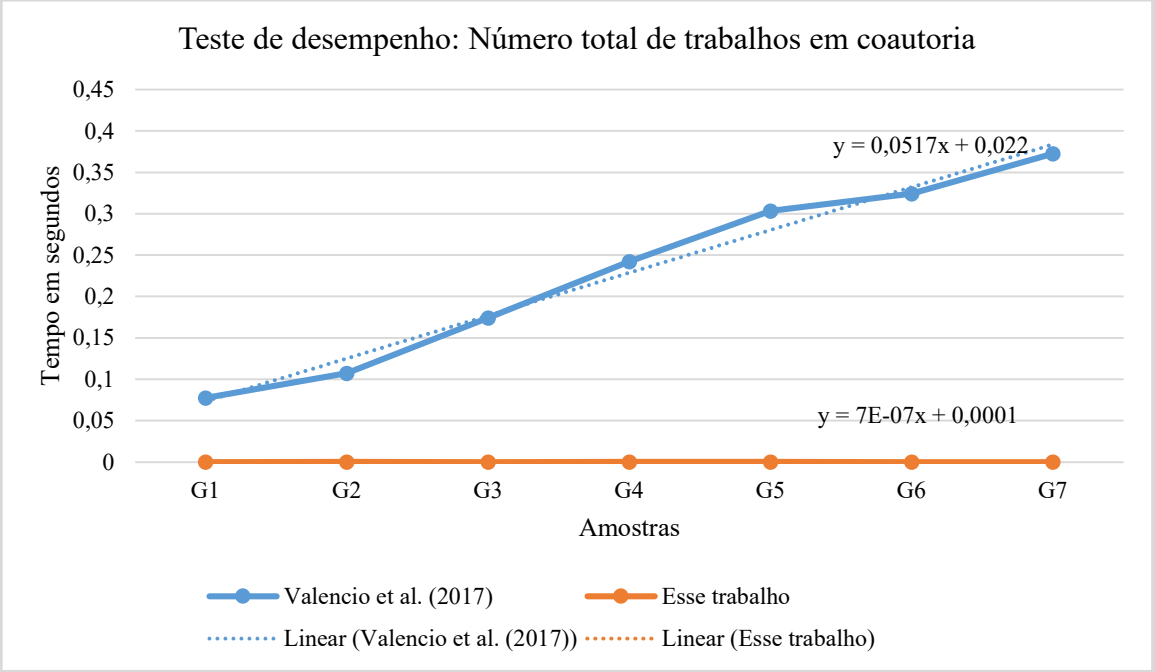
Amostra	Valencio et al. (2017)		Esse trabalho	
	Tempo médio (s)	Desvio padrão	Tempo médio (s)	Desvio padrão
G1	0,07776	0,0202	0,00013	0,00003
G2	0,10721	0,0249	0,00014	0,00002
G3	0,17441	0,0212	0,00016	0,00008
G4	0,24227	0,0498	0,00014	0,00003
G5	0,30351	0,0289	0,00014	0,00002
G6	0,32436	0,0297	0,00016	0,00006
G7	0,37261	0,0533	0,00013	0,00002

Fonte: Elaborado pelo autor

A comparação entre os testes de desempenho realizados sobre ambos os trabalhos é exibida no gráfico ilustrado na Figura 6. Por meio da variação do tempo entre as execuções do algoritmo para diferentes amostras, extraiu-se que o algoritmo de número total de trabalhos em coautoria implementado em Valencio et al. (2017) tem crescimento de tempo 73.857 vezes maior que o implementado nesse trabalho quando o tamanho do grafo tende ao infinito. Esse número pode ser explicado por conta de o valor da quantidade de arestas do grafo ser uma

constante armazenada na memória principal quando o grafo da rede de colaboração é carregado na memória principal para processamento.

Figura 6 – teste de desempenho do algoritmo de número total de trabalhos em coautoria



Fonte: Elaborado pelo autor

Sobre o algoritmo de extração da bibliométrica número total de coautorias, os valores médios dos tempos em segundos das execuções para as diferentes amostras e os respectivos desvios padrões são exibidos na Tabela 12.

Tabela 12 – teste de desempenho do algoritmo de número total de coautorias

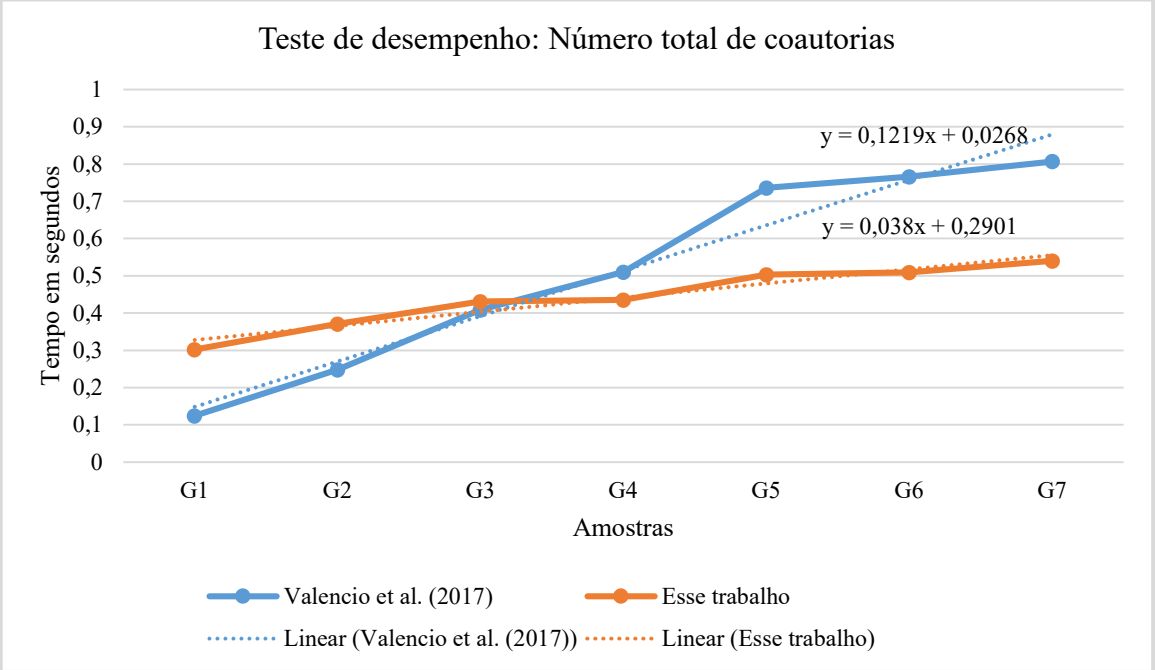
Amostra	Valencio et al. (2017)		Esse trabalho	
	Tempo médio (s)	Desvio padrão	Tempo médio (s)	Desvio padrão
G1	0,12432	0,0162	0,30252	0,0395
G2	0,24789	0,0382	0,37107	0,0319
G3	0,40836	0,0173	0,43107	0,0578
G4	0,51024	0,0326	0,43571	0,0305
G5	0,73621	0,0238	0,50294	0,0363
G6	0,76584	0,0655	0,50968	0,0273
G7	0,80721	0,0497	0,54050	0,0467

Fonte: Elaborado pelo autor

A comparação entre os testes de desempenho realizados sobre ambos os trabalhos é exibida no gráfico ilustrado na Figura 7. Por meio da variação do tempo entre as execuções do algoritmo para diferentes amostras, extraiu-se que o algoritmo de número total de coautorias

implementado em Valencio et al. (2017) tem crescimento de tempo 3,21 vezes maior que o implementado nesse trabalho quando o tamanho do grafo tende ao infinito.

Figura 7 – teste de desempenho do algoritmo de número total de coautorias



Fonte: Elaborado pelo autor

Sobre o algoritmo de extração da bibliométrica total de coautores, os valores médios dos tempos em segundos das execuções para as diferentes amostras e os respectivos desvios padrões são exibidos na Tabela 13.

Tabela 13 – teste de desempenho do algoritmo de número total de coautores

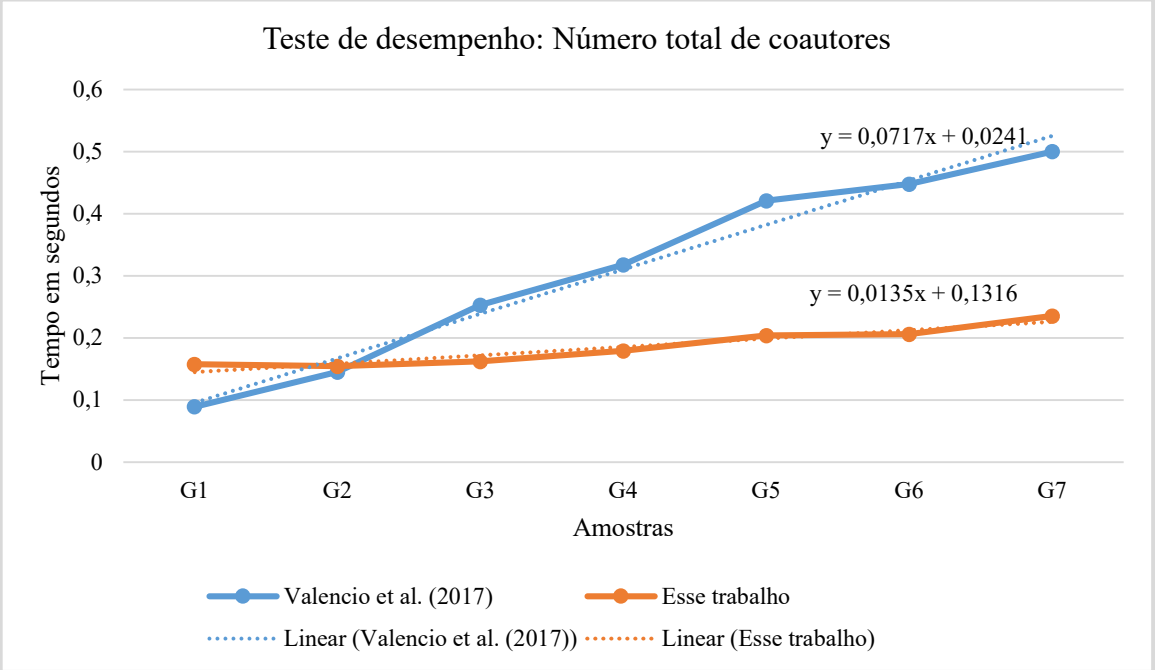
Amostra	Valencio et al. (2017)		Esse trabalho	
	Tempo médio (s)	Desvio padrão	Tempo médio (s)	Desvio padrão
G1	0,08936	0,0139	0,15785	0,0137
G2	0,14537	0,0213	0,15455	0,0113
G3	0,25302	0,0249	0,16223	0,0357
G4	0,31799	0,0464	0,17912	0,0201
G5	0,42104	0,0565	0,20422	0,0429
G6	0,44803	0,0326	0,20622	0,0197
G7	0,50034	0,0371	0,23547	0,0479

Fonte: Elaborado pelo autor

A comparação entre os testes de desempenho realizados sobre ambos os trabalhos é exibida no gráfico ilustrado na Figura 8. Por meio da variação do tempo entre as execuções do algoritmo para diferentes amostras, extraiu-se que o algoritmo de número total de coautores

implementado em Valencio et al. (2017) tem crescimento de tempo 5,31 vezes maior que o implementado nesse trabalho quando o tamanho do grafo tende ao infinito.

Figura 8 – teste de desempenho do algoritmo de número total de coautores



Fonte: Elaborado pelo autor

Sobre o algoritmo de extração da bibliométrica assortatividade, os valores médios dos tempos em segundos das execuções para as diferentes amostras e os respectivos desvios padrões são exibidos na Tabela 14.

Tabela 14 – teste de desempenho do algoritmo de assortatividade

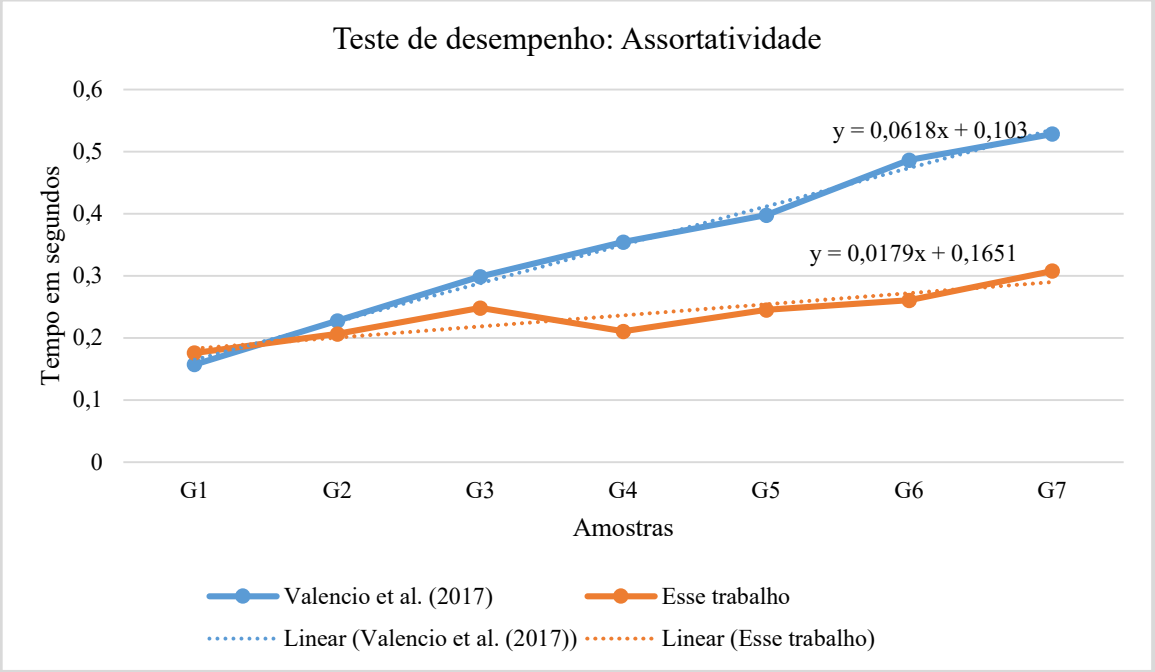
Amostra	Valencio et al. (2017)		Esse trabalho	
	Tempo médio (s)	Desvio padrão	Tempo médio (s)	Desvio padrão
G1	0,15751	0,0086	0,17613	0,0459
G2	0,22775	0,0262	0,20679	0,0663
G3	0,29881	0,0423	0,24832	0,0612
G4	0,35444	0,0438	0,21068	0,0757
G5	0,39773	0,0130	0,24522	0,0413
G6	0,48658	0,0489	0,26078	0,0959
G7	0,52873	0,0287	0,30794	0,1061

Fonte: Elaborado pelo autor

A comparação entre os testes de desempenho realizados sobre ambos os trabalhos é exibida no gráfico ilustrado na Figura 9. Por meio da variação do tempo entre as execuções do algoritmo para diferentes amostras, extraiu-se que o algoritmo de assortatividade

implementado em Valencio et al. (2017) tem crescimento de tempo 3,45 vezes maior que o implementado nesse trabalho quando o tamanho do grafo tende ao infinito.

Figura 9 – teste de desempenho do algoritmo de assortatividade



Fonte: Elaborado pelo autor

4.4.3. Caminho médio e diâmetro

Por fim, foram realizados os testes de desempenho dos algoritmos para cálculo do caminho médio e diâmetro desenvolvido nesse trabalho, o qual utiliza o algoritmo de busca em largura para realizar a extração de todos os caminhos mínimos. Entretanto, como relatado em Schoeneman e Zola (2019), por meio da implementação básica do Apache Spark, torna-se impraticável o desenvolvimento de uma implementação do algoritmo de extração de todos os caminhos mínimos eficiente e escalável.

Por esse motivo, o algoritmo desenvolvido por esse trabalho não se mostrou eficiente quando implementado por meio do Apache Spark. Nos testes de desempenho, o algoritmo se mostrou menos eficiente que o implementado em Valencio et al. (2017) nas quatro primeiras amostras (G1, G2, G3 e G4) e, a partir de então, apresentou o erro de estouro de memória e foi interrompido. Os valores dos tempos em segundos da execução para as diferentes amostras do algoritmo de caminho médio e diâmetro são exibidos na Tabela 15.

Tabela 15 – teste de desempenho do algoritmo de Caminho médio e Diâmetro

Amostra	Tempo (s)	
	Valencio et al. (2017)	Esse trabalho
G1	1,75	146,1
G2	5,22	412,2
G3	12,93	1006,6
G4	18,87	1783,6
G5	30,02	Falha
G6	40,83	Falha
G7	51,37	Falha

Fonte: Elaborado pelo autor

4.5. Teste de escalabilidade

Para os testes de escalabilidade, foi importada uma rede de colaboração disponível em Stanford (2019) nominada *LiveJournal social network*, que representa uma comunidade livre com quase 10 milhões de membros, sendo uma fração significante desses membros altamente ativos. A rede de colaboração importada foi dividida em 4 amostras, compostas pelos seguintes itens:

a) G1 – 100.000 pesquisadores e 1.659.270 trabalhos;
b) G2 – 200.000 pesquisadores e 4.454.901 trabalhos;
c) G3 – 300.000 pesquisadores e 7.305.114 trabalhos;
d) G4 – 400.000 pesquisadores e 10.125.632 trabalhos;

Os valores dos tempos em segundos da execução para as diferentes amostras do algoritmo de transitividade são exibidos na Tabela 16.

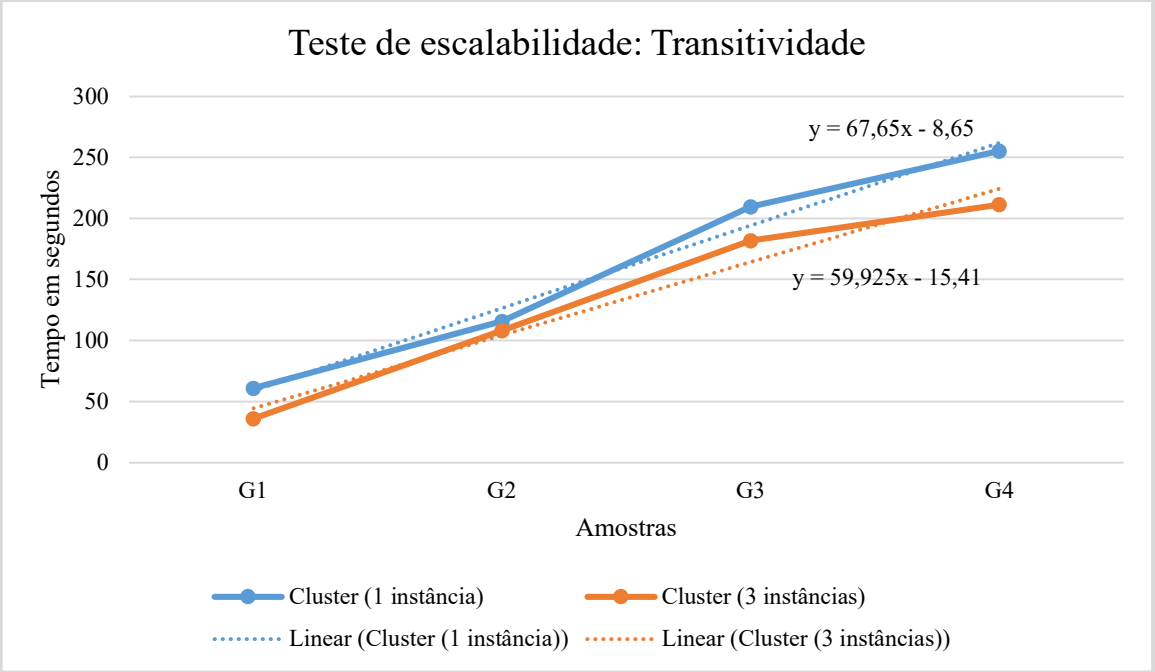
Tabela 16 – teste de escalabilidade do algoritmo de Transitividade

Amostra	Tempo (s)	
	Cluster (1 instância)	Cluster (3 instâncias)
G1	61,1	36,17
G2	115,8	108,19
G3	209,7	181,9
G4	255,3	211,35

Fonte: Elaborado pelo autor

A comparação entre os testes de escalabilidade realizados sobre ambos os clusters é exibida no gráfico ilustrado na Figura 10.

Figura 10 – teste de escalabilidade do algoritmo de Transitividade



Fonte: Elaborado pelo autor

Por meio da variação do tempo entre as execuções do algoritmo de transitividade para diferentes amostras e diferentes clusters, extraiu-se as seguintes informações:

a) cluster (1 instância) – as variações de tempo para as diferentes amostras tendem para a Equação (12), com coeficiente angular constate de 67,65;
b) cluster (3 instâncias) – as variações de tempo para as diferentes amostras tendem para a Equação (13), com coeficiente angular constate de 59,925.

$$y = 67{,}65x - 8{,}65 \tag{12}$$

$$y = 59{,}925x - 15{,}41 \tag{13}$$

Portanto, o algoritmo de transitividade executado no cluster (1 instância) tem crescimento de tempo 1,129 vezes maior que o executado no cluster (3 instâncias). Dessa forma, infere-se que o algoritmo pode ser escalado horizontalmente ao aumentar o número de instâncias dos clusters.

4.6. Considerações finais

Os testes de corretude demonstraram que os algoritmos desenvolvidos extraem corretamente as bibliométricas. Os testes de caso de uso demonstraram a utilidade da ferramenta para análise e extração de conhecimento de redes de colaboração científica. Os testes de

desempenho demonstram que, quando o número de pesquisadores tende ao infinito, o algoritmo de extração da bibliométrica de transitividade desenvolvido tem crescimento de tempo de processamento 12,76 vezes menor que o tempo de processamento do algoritmo paralelo proposto na literatura. Por fim, os testes de escalabilidade demonstram que o algoritmo de transitividade escala horizontalmente conforme é aumentada a quantidade de instâncias do cluster.

5. Conclusão

Com o avanço da tecnologia e o aumento de trabalhos científicos desenvolvidos em coautoria, surgiram grandes redes de colaboração, das quais a extração e análise de bibliométricas podem contribuir amplamente para a caracterização da ciência em geral. Entretanto, os algoritmos bibliométricos não são escaláveis e precisam utilizar todos os recursos computacionais de forma ótima para extraírem bibliométricas dessas redes de colaboração científica.

Portanto, nesse trabalho foram desenvolvidos algoritmos de extração de bibliométricas distribuídos e escaláveis, suportados por uma ferramenta de extração de indicadores bibliométricos de redes de colaboração genéricas, a qual contempla algoritmos para importação dos dados, filtros e os algoritmos de extração de bibliométricas otimizados.

Em relação ao desempenho, o algoritmo de transitividade foi melhorado e teve tempo de execução em média 12,76 vezes menor que o algoritmo desenvolvido em Valencio et al. (2017). Além disso, foi demonstrado por meio dos testes de escalabilidade que o algoritmo de transitividade escala horizontalmente conforme é aumentada a quantidade de instâncias do cluster utilizado para seu processamento.

Com as melhorias realizadas nesse trabalho, o estudo bibliométrico em tempo hábil de qualquer rede de colaboração cujo dados estejam acessíveis ao usuário será facilitado e, assim, possibilitará a caracterização de redes de colaboração à nível continental ou até global.

5.1. Contribuições científicas

Por fim, as contribuições científicas desse trabalho podem ser visualizadas com o auxílio da Tabela 17, que compara esse trabalho aos seus principais correlatos.

As contribuições científicas desse trabalho são: a otimização dos algoritmos de extração de bibliométricas por meio de distribuição de dados para obterem desempenho superior aos semelhantes encontrados na literatura para grandes redes de colaboração. Além disso, como subproduto, uma ferramenta de extração de indicadores bibliométricos para facilitar o estudo de redes de colaboração e com um importador para redes de colaboração de qualquer SGBD relacional ou dos formatos de arquivo XML e JSON.

Tabela 17 – comparação entre os trabalhos correlatos e esse trabalho.

Características	Mena-Chalco et al. (2014)	Valencio et al. (2017)	Esse trabalho
Visualização comparativa das bibliométricas	✓	✗	✓
Extração de bibliométricas	✓	✓	✓
Análise de bibliométricas	✓	✓	✓
Armazenamento em banco de dados orientado a grafos	✓	✓	✓
Ferramenta para extração de rede de colaboração genérica	✗	✓	✓
Paralelização das métricas mais custosas	✗	✓	✓
Filtros para a rede de colaboração	✗	✓	✓
Distribuição das bibliométricas	✗	✗	✓
Importador de redes de colaboração de SGBD relacional PostgreSql, Plataforma Lattes e o formato de arquivo CSV	✗	✓	✓
Importador de redes de colaboração de qualquer SGBD relacional e os formatos de arquivo XML e JSON	✗	✗	✓

Fonte: Elaborado pelo autor.

5.2. Trabalhos futuros

Para a continuação do trabalho, os seguintes pontos são destacados: propor um pré-processador para processar o grafo em um subgrafo de acordo com os filtros utilizados. O subgrafo teria apenas uma aresta entre os pesquisadores e o peso da aresta representaria a quantidade de trabalhos entre ambos. Após isso, adaptar os algoritmos desenvolvidos nesse

trabalho e verificar se a redução da quantidade de arestas melhoraria o desempenho dos algoritmos de forma a compensar o tempo despendido para o pré-processamento do grafo.

Referências

AMAZON. Amazon EMR – Amazon Web Services. Disponível em: https://aws.amazon.com/pt/emr. Acesso em: 17 jul. 2019.

APACHE SPARK. RDD Programming Guide. Disponível em: https://spark.apache.org/docs/latest/rdd-programming-guide.html. Acesso em: 23 mai. 2019.

ARCHAMBAULT, É.; CAMPBELL, D.; GINGRAS, Y.; LARIVIÈRE, V. Comparing bibliometric statistics obtained from the Web of Science and Scopus. **Journal of the Association for Information Science and Technology**, v. 60, n. 7, p. 1320-1326, 2009.

BATRA, S.; TYAGI, C. Comparative analysis of relational and graph databases. **International Journal of Soft Computing and Engineering (IJSCE)**, v. 2, n. 2, p. 509-512, 2012.

BHADANI, A. K.; JOTHIMANI, D. Big Data: Challenges, Opportunities, and Realities. **Effective Big Data Management and Opportunities for Implementation, IGI Global, Pennsylvania, USA**, p. 1-24, 2016.

BORNMANN, L.; WAGNER, C.; LEYDESDORFF, L. BRICS countries and scientific excellence: A bibliometric analysis of most frequently cited papers. **Journal of the Association for Information Science and Technology**, v. 66, n. 7, p. 1507-1513, 2015.

CHEN, M.; MAO, S.; LIU, Y. Big data: A survey. **Mobile Networks and Applications**, v. 19, n. 2, p. 171-209, 2014.

CHEN, Y.; DING, C.; HU, J.; CHEN, R.; HUI, P.; FU, X. Building and analyzing a global co-authorship network using google scholar data. *In*: **Proceedings of the 26th International Conference on World Wide Web Companion**. International World Wide Web Conferences Steering Committee, 2017. p. 1219-1224.

CNPQ. Plataforma Lattes. Disponível em: http://lattes.cnpq.br. Acesso em: 17 jul. 2019.

D3JS. D3.js - Data-Driven Documents. Disponível em: https://d3js.org. Acesso em: 17. jul. 2019.

DA SILVA, A. K. A.; BARBOSA, R. R.; DUARTE, E. N. Rede social de coautoria em Ciência da Informação: estudo sobre a área temática de "Organização e Representação do Conhecimento". **Informação & Sociedade**, v. 22, n. 2, 2012.

DA SILVA MORAIS, T. Survey on frameworks for distributed computing: Hadoop, Spark and Storm. *In*: **Proceedings of the 10th Doctoral Symposium in Informatics Engineering-DSIE**, v. 15, p. 95-105, 2015.

DAVIS, A. L. Functional Programming. *In*: **Modern Programming Made Easy**. California: Editora Apress, Berkeley, CA. p. 49-56, 2016.

DAVOUDIAN, A.; CHEN, L.; LIU, M. A Survey on NoSQL Stores. **ACM Computing Surveys (CSUR)**, v. 51, n. 2, p. 40, 2018.

DBLP. Dblp: computer science bibliography. Disponível em: https://dblp.uni-trier.de. Acesso em: 17 jul. 2019.

DEAN, J.; GHEMAWAT, S. MapReduce: simplified data processing on large clusters. **Communications of the ACM**, v. 51, n. 1, p. 107-113, 2008.

FAHAD, A.; ALSHATRI, N.; TARI, Z.; ALAMRI, A.; KHALIL, I.; ZOMAYA, A. Y.; FOUFOU, S.; BOURAS, A. A survey of clustering algorithms for big data: Taxonomy and empirical analysis. **IEEE transactions on emerging topics in computing**, v. 2, n. 3, p. 267-279, 2014.

FINARDI, U.; BURATTI, A. Scientific collaboration framework of BRICS countries: an analysis of international coauthorship. **Scientometrics**, v. 109, n. 1, p. 433-446, 2016.

GANTZ, J.; REINSEL, D. Extracting value from chaos. **IDC iview**, v. 1142, n. 2011, p. 1-12, 2011.

GBD. Portal GBD - Sistema Computacional de Apoio às Secretarias Departamentais. Disponível em: https://www.grupogbd.com/PortalGBD/projeto_info?idProjeto=9. Acesso em 17 jul. 2019.

GONZALEZ, J. E.; XIN, R. S.; DAVE, A.; CRANKSHAW, D.; FRANKLIN, M. J.; STOICA, I. Graphx: Graph processing in a distributed dataflow framework. *In*: **11th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 14)**. 2014. p. 599-613.

GRAPHX. GraphX - Spark 2.4.3 Documentation. Disponível em: https://spark.apache.org/docs/latest/graphx-programming-guide.html. Acesso em: 04 mai. 2020.

HAN, J.; HAIHONG, E.; LE, G.; DU, J. Survey on NoSQL database. *In*: Pervasive computing and applications (ICPCA), 2011 6th international conference on. IEEE, 2011. p. 363-366.

HEILIG, L.; VOß, S. A scientometric analysis of cloud computing literature. **IEEE Transactions on Cloud Computing**, v. 2, n. 3, p. 266-278, 2014.

HOOD, W.; WILSON, C. The literature of bibliometrics, scientometrics, and informetrics. **Scientometrics**, v. 52, n. 2, p. 291-314, 2001.

JAISWAL, G.; AGRAWAL, A. P. Comparative analysis of Relational and Graph databases. **IOSR Journal of Engineering (IOSRJEN)**, v. 3, n. 8, p. 25-27, 2013.

JUKIC, S.; SUBASI, A. A MapReduce-based rotation forest classifier for epileptic seizure prediction. **arXiv preprint arXiv:1712.06071**, 2017.

KATZ, J. S.; HICKS, D. How much is a collaboration worth? A calibrated bibliometric model. **Scientometrics**, v. 40, n. 3, p. 541-554, 1997.

LEE, K. H.; LEE, Y. J.; CHOI, H.; CHUNG, Y. D.; MOON, B. Parallel data processing with MapReduce: a survey. **AcM sIGMoD Record**, v. 40, n. 4, p. 11-20, 2012.

MACIAS-CHAPULA, C. A. O papel da informetria e da cienciometria e sua perspectiva nacional e internacional. **Ciência da informação**, v. 27, n. 2, p. 134-140, 1998.

MENA-CHALCO, J. P.; DIGIAMPIETRI, L. A.; CESAR-JR, R. M. Caracterizando as redes de coautoria de currículos Lattes. *In*: **Brazilian Workshop on Social Network Analysis and Mining (BraSNAM)**. 2012. p. 1-12.

MENA-CHALCO, J. P.; DIGIAMPIETRI, L. A.; LOPES, F. M.; CESAR, R. M. Brazilian bibliometric coauthorship networks. **Journal of the Association for Information Science and Technology**, v. 65, n. 7, p. 1424-1445, 2014.

MILGRAM, S. The small world problem. **Psychology today**, v. 2, n. 1, p. 60-67, 1967.

MOED, H. F.; BURGER, W. J. M.; FRANKFORT, J. G.; VAN RAAN, A. F. The use of bibliometric data for the measurement of university research performance. **Research Policy**, v. 14, n. 3, p. 131-149, 1985.

NA, H.; LUO, H.; TING, W.; WANG, B. Multi-task parallel algorithm for dsrc. **Procedia Computer Science**, v. 31, p. 1133-1139, 2014.

NASSER, T.; TARIQ, R. S. Big data challenges. **J Comput Eng Inf Technol 4: 3. DOI: 10.4172/2324**, v. 9307, n. 2, 2015.

NEWMAN, M. E. The structure of scientific collaboration networks. **Proceedings of the National Academy of Sciences**, v. 98, n. 2, p. 404-409, 2001a.

NEWMAN, M. E. Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality. **Physical review E**, v. 64, n. 1, p. 016132, 2001b.

NEWMAN, M. E. Assortative mixing in networks. **Physical review letters**, v. 89, n. 20, p. 208-211, 2002.

N'TAKPÉ, T.; SUTER, F.; CASANOVA, H. A comparison of scheduling approaches for mixed-parallel applications on heterogeneous platforms. *In*: **Parallel and Distributed Computing, 2007. ISPDC'07. Sixth International Symposium on**. IEEE, 2007. p. 35-35.

OLAWUMI, T. O.; CHAN, D. W. M. A scientometric review of global research on sustainability and sustainable development. **Journal of cleaner production**, v. 183, p. 231-250, 2018.

POSTGRESQL. PostgreSQL: The world's most advanced open source database. Disponível em: https://www.postgresql.org. Acesso em: 17 jul. 2019.

RAMOS-RODRÍGUEZ, A.; RUÍZ-NAVARRO, J. Changes in the intellectual structure of strategic management research: A bibliometric study of the Strategic Management Journal, 1980–2000. **Strategic Management Journal**, v. 25, n. 10, p. 981-1004, 2004.

SCHOENEMAN, F.; ZOLA, J. Solving All-Pairs Shortest-Paths Problem in Large Graphs Using Apache Spark. In: **Proceedings of the 48th International Conference on Parallel Processing**. p. 1-10, 2019.

SHI, J.; QIU, Y.; MINHAS, U. F.; JIAO, L.; WANG, C.; REINWALD, B.; ÖZCAN, F. Clash of the titans: Mapreduce vs. spark for large scale data analytics. **Proceedings of the VLDB Endowment**, v. 8, n. 13, p. 2110-2121, 2015.

SINGH, D.; REDDY, C. K. A survey on platforms for big data analytics. **Journal of Big Data**, v. 2, n. 1, p. 8, 2015.

SPRING. Spring Data JPA. Disponível em: https://spring.io/projects/spring-data-jpa. Acesso em: 17 jul. 2019.

STANFORD. SNAP: Stanford Network Analysis Project. Disponível em: https://snap.stanford.edu/index.html. Acesso em: 17 jul. 2019.

STOREY, V. C.; SONG, Il-Y. Big data technologies and Management: What conceptual modeling can do. **Data & Knowledge Engineering**, v. 108, p. 50-67, 2017.

SUBRAMANYAM, K. Bibliometric studies of research collaboration: A review. **Information Scientist**, v. 6, n. 1, p. 33-38, 1983.

TAGUE-SUTCLIFFE, J. An introduction to informetrics. **Information processing & management**, v. 28, n. 1, p. 1-3, 1992.

VALENCIO, C. R.; DE FREITAS, J. C.; DE SOUZA, R. C. G.; NEVES, L. A.; ZAFALON, G. F. D.; COLOMBINI, A. C.; TENORIO, W. An efficient parallel optimization for co-authorship network analysis. *In*: **Parallel and Distributed Computing, Applications and Technologies (PDCAT), 2017 18th International Conference on**. IEEE, 2017. p. 127-134.

VALENCIO, C. R.; DE FREITAS, J. C.; DE SOUZA, R. C. G.; NEVES, L. A.; ZAFALON, G. F. D.; COLOMBINI, A. C.; TENORIO, W. Analysing research collaboration through co-authorship networks in a big data environment: an efficient parallel approach. **Int. J. Computational Science and Engineering**, v. 21, n. 3, p. 364–374, 2020.

VAN ECK, N. J.; WALTMAN, L. Software survey: VOSviewer, a computer program for bibliometric mapping. **Scientometrics**, v. 84, n. 2, p. 523-538, 2010.

VAN RAAN, A. Advanced bibliometric methods as quantitative core of peer review based evaluation and foresight exercises. **Scientometrics**, v. 36, n. 3, p. 397-420, 1996.

XIN, R. S.; CRANKSHAW, D.; DAVE, A.; GONZALEZ, J. E.; FRANKLIN, M. J.; STOICA, I. Graphx: Unifying data-parallel and graph-parallel analytics. **arXiv preprint arXiv:1402.2394**, 2014.

WASSERMAN, S.; FAUST, K. **Social network analysis: Methods and applications**. Cambridge: Editora Cambridge university press, 1994.

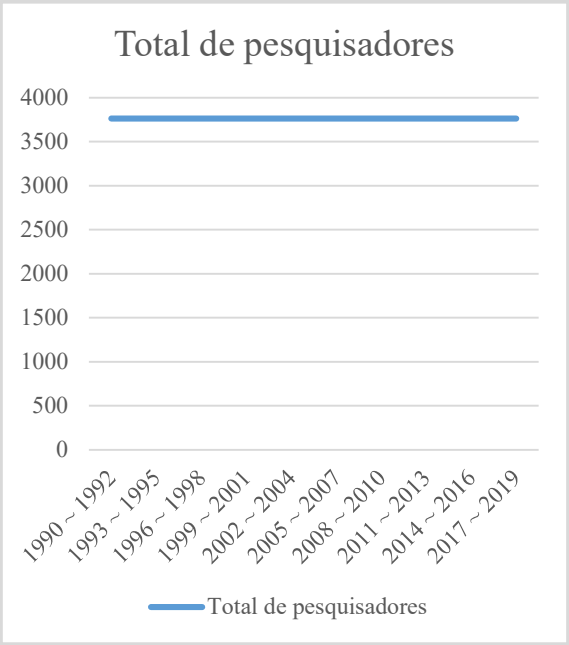
CHURCHILL, W. How to Say a Few Words by David Guy Powers, p. 109, 1959.

WU, X.; ZHU, X.; WU, G. Q.; DING, W. Data mining with big data. **IEEE transactions on knowledge and data engineering**, v. 26, n. 1, p. 97-107, 2013.

ZHANG, C.; BU, Y.; DING, Y.; XU, J. Understanding scientific collaboration: Homophily, transitivity, and preferential attachment. **Journal of the Association for Information Science and Technology**, v. 69, n. 1, p. 72-86, 2018.

Apêndice A – gráficos comparativos das bibliométricas extraídas no estudo trienal

Figura 11 – gráfico trienal de total de pesquisadores

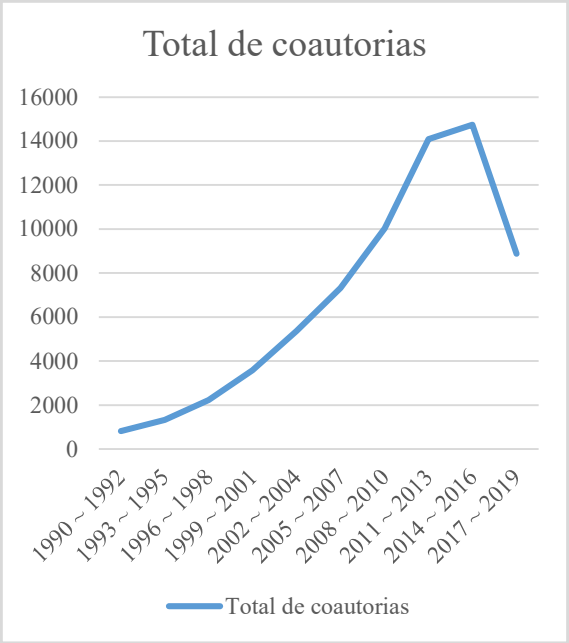


Fonte: Elaborado pelo autor.

Figura 12 – gráfico trienal de total de trabalhos em coautoria

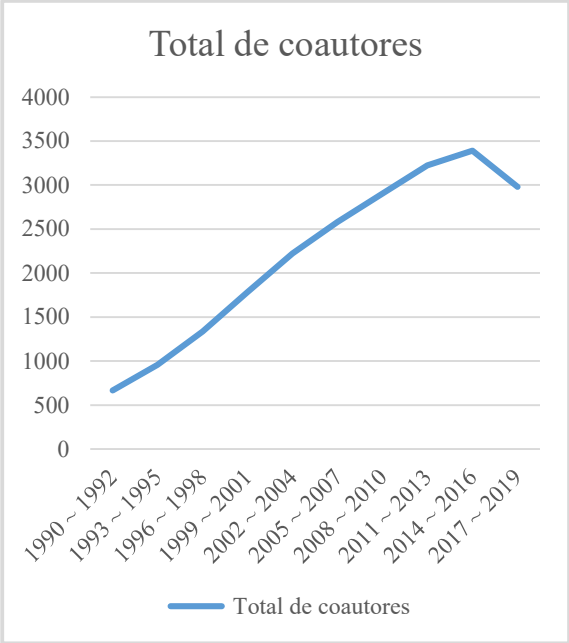
Fonte: Elaborado pelo autor.

Figura 13 – gráfico trienal do total de coautorias



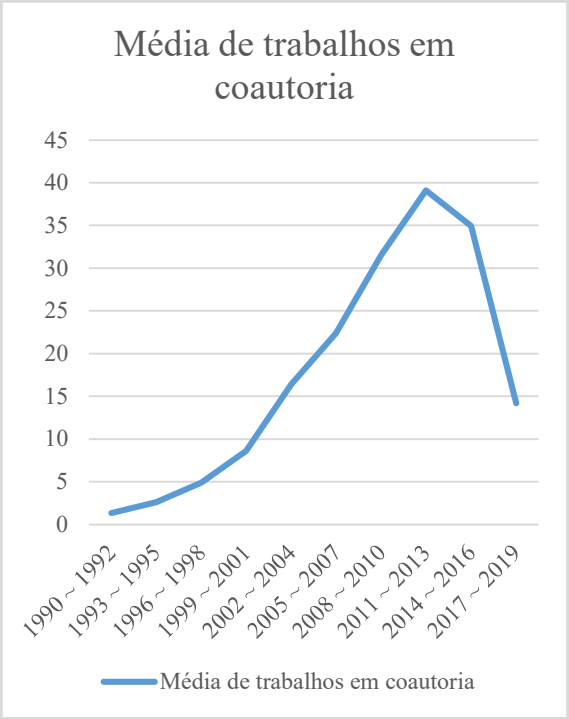
Fonte: Elaborado pelo autor.

Figura 14 – gráfico trienal do total de coautores



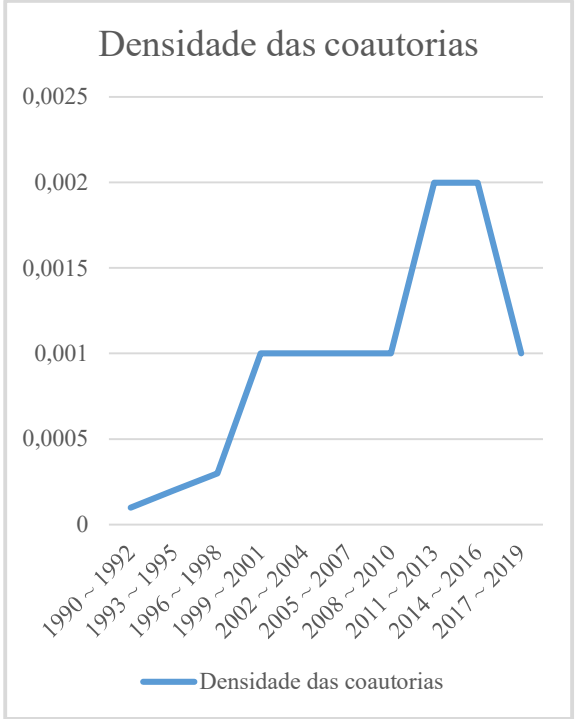
Fonte: Elaborado pelo autor.

Figura 15 – gráfico trienal da média de trabalhos em coautoria



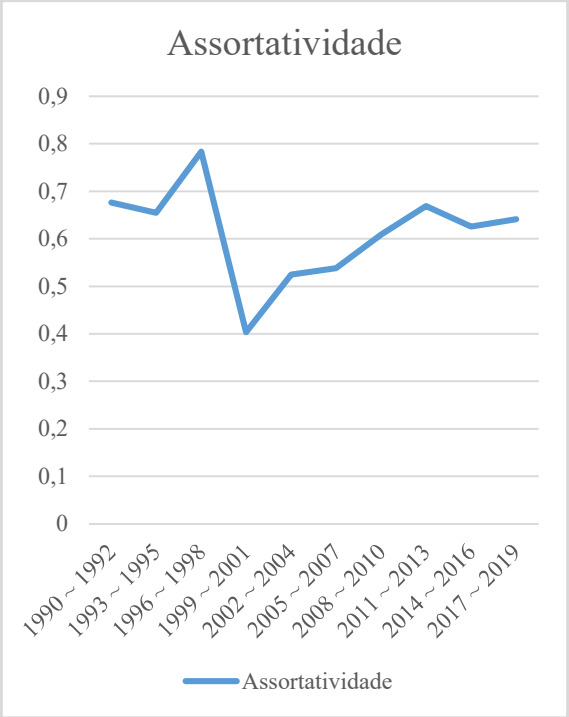
Fonte: Elaborado pelo autor.

Figura 16 – gráfico trienal da densidade das coautorias



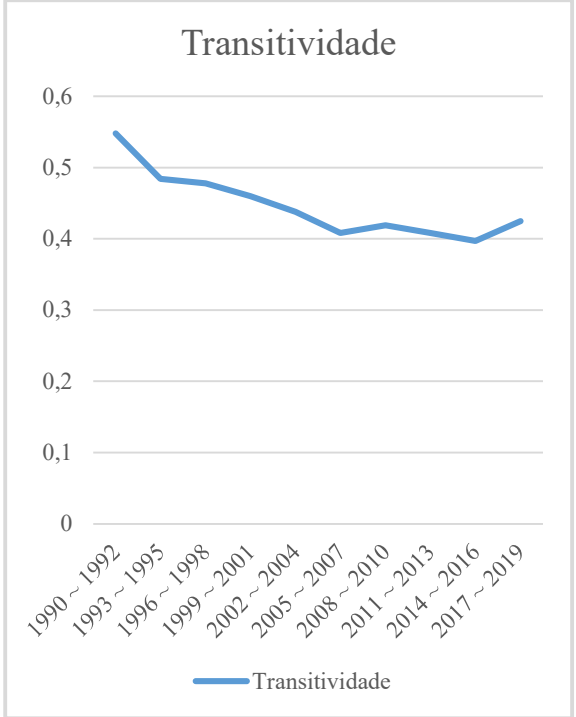
Fonte: Elaborado pelo autor.

Figura 17 – gráfico trienal da Assortatividade



Fonte: Elaborado pelo autor.

Figura 18 – gráfico trienal da Transitividade

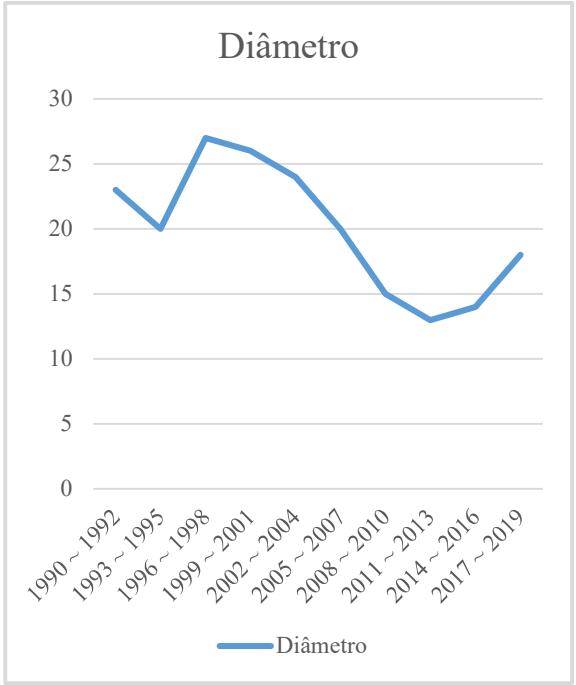


Fonte: Elaborado pelo autor.

Figura 19 – gráfico trienal da distância média entre os pesquisadores

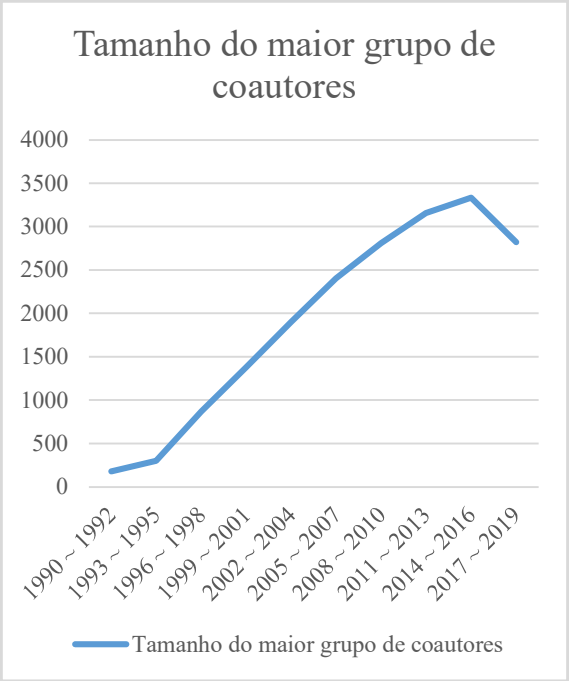
Fonte: Elaborado pelo autor.

Figura 20 – gráfico trienal do Diâmetro da rede de colaboração



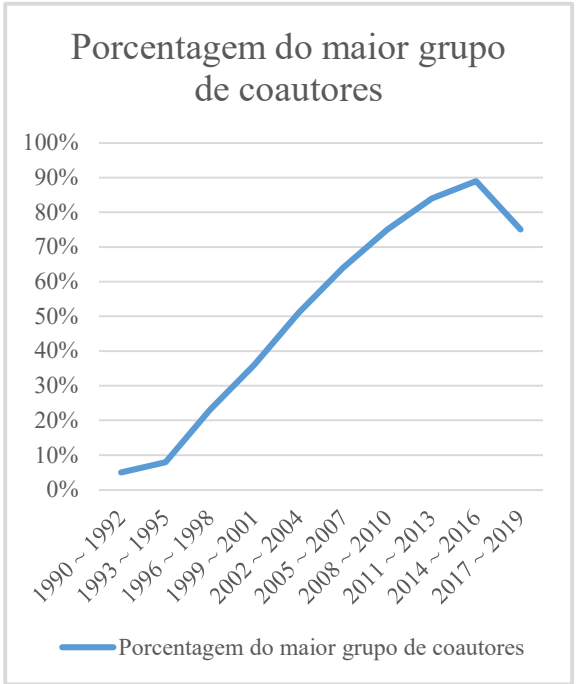
Fonte: Elaborado pelo autor.

Figura 21 – gráfico trienal do tamanho do maior grupo de coautores



Fonte: Elaborado pelo autor.

Figura 22 – gráfico trienal da porcentagem do maior grupo de coautores



Fonte: Elaborado pelo autor.