

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE CIÊNCIA DA COMPUTAÇÃO – BACHARELADO

**GERAÇÃO DE REDES COMPLEXAS
COM COMUNIDADES SOBREPOSTAS E
COMUNIDADES HIERÁRQUICAS**

GUSTAVO HENRIQUE SPIESS

**BLUMENAU
2022**

GUSTAVO HENRIQUE SPIESS

GERAÇÃO DE REDES COMPLEXAS COM COMUNIDADES SOBREPOSTAS E COMUNIDADES HIERÁRQUICAS

Trabalho de Conclusão de Curso apresentado ao curso de graduação em Ciência da Computação no Centro de Ciências Exatas e Naturais da Universidade Regional de Blumenau como requisito parcial para a obtenção de grau de Bacharel em Ciência da Computação.

Professor Aurelio Faustino Hoppe, Mestre - Orientador

GERAÇÃO DE REDES COMPLEXAS COM COMUNIDADES SOBREPOSTAS E COMUNIDADES HIERÁRQUICAS

Por

GUSTAVO HENRIQUE SPIESS

Trabalho de Conclusão de Curso aprovado para obtenção dos créditos na disciplina de Trabalho de Conclusão de Curso II pela banca examinadora formada por:

Presidente: Aurélio Faustino Hoppe
Prof(a). Aurélio Faustino Hoppe - Orientador(a), FURB

Membro: Gilvan Justino
Prof(a). Gilvan Justino - FURB

Membro: Jonathan Gil Müller
Prof(a). Jonathan Gil Müller - FURB

Blumenau, 07 de julho de 2022

Dedico esse trabalho a minha noiva, cuja paciência em me ouvir falar desse trabalho tornou-o possível.

AGRADECIMENTOS

A meu padrinho, Maiko Rafael Spiess, pelo sempre presente incentivo ao estudo.

Ao meu orientador, Aurélio Faustino Hoppe, por acreditar na conclusão desse trabalho.

A minha família, por todos os anos de apoio que foram necessários para chegar até aqui.

Aos amigos que fiz no percurso do bacharelado, pelo apoio recebido.

Aos professores do Departamento de Sistemas e Computação da Universidade Regional de Blumenau por suas contribuições durante os semestres letivos.

“Se eu vi mais longe, foi por estar sobre ombros
de gigantes.”

Isaac Newton

RESUMO

Sistemas do mundo real são modelados como grafos com atributos, significando uma estrutura de dados onde tem-se uma caracterização dos nodos do sistema, bem como as relações entre eles. Nesses grafos observados no mundo real, algumas propriedades naturalmente estão presentes na topografia da rede. Uma dessas propriedades é a tendência de formação de agrupamentos que podem ser descritos como comunidades. Elas em muitos sistemas possuem a característica de serem organizadas de forma auto semelhantes, isso é, comunidades que são compostas por sub-comunidades, formando uma estrutura aninhada. Comunidades também tendem, em alguns sistemas do mundo real, a apresentarem áreas de sobreposição, onde nodos pertencem simultaneamente a múltiplas comunidades. Este trabalho apresenta um modelo algorítmico de geração de redes complexas com comunidades hierarquicamente aninhadas e com comunidades sobrepostas. O objetivo principal do modelo é a parametrização e controle dessas propriedades durante o processo de construção do grafo para a disponibilização de uma *ground truth* contra a qual algoritmos de detecção de comunidades podem ser avaliados. A avaliação da presença dessas propriedades é feita utilizando as funções de inércia e modularidade.

Palavras-chave: Redes complexas. Geração de redes complexas. Comunidades. Comunidades sobrepostas. Comunidades hierárquicas.

ABSTRACT

It is not rare that real world systems get modeled as attributed graph, meaning a data structure where there is the categorization of the nodes, and also the relationships between nodes. In this graphs from the real world, some properties naturally happen in the network topography. One of this properties is the tendency to form clusters that can be described as communities. In many systems those communities are organized in a self similar way, e.g. communities composed by sub communities, forming a nested structure. Communities also tend, in some real world systems, to have overlapping regions, where nodes belong to both communities. This paper presents an algorithmic model for generating complex networks with hierarchical communities and with overlapping communities. The main goal of this model is the parameterization and control of those properties during the building process of the graph to make available a ground truth against which detection algorithms can be tested. The validation of the presence of those properties is done using the inertia and modularity properties.

Keywords: Complex networks. Complex networks generation. Communities. Overlapping communities. Hierarchical communities

LISTA DE FIGURAS

Figura 1 – Exemplo de grafo	16
Figura 2 – Exemplo de grafo com comunidades hierárquicas	22
Figura 3 – Demonstração dos resultados de diferentes algoritmos de detecção em um grafo com comunidades hierárquicas e com sobreposição	23
Figura 4 – M_0 renderizado aproximando os vértices adjacentes	49
Figura 5 – M_0 renderizado com os vértices posicionado pelos seus atributos	50
Figura 6 – M_0 renderizado com os vértices posicionado pelos seus atributos (apenas arestas)	50
Figura 7 – B_0 renderizado com os vértices posicionado pelos seus atributos	52
Figura 8 – B_0 renderizado com os vértices posicionado pelos seus atributos (apenas arestas)	52
Figura 9 – B_0 renderizado aproximando os vértices adjacentes	53
Figura 10 – B_0 renderizado aproximando os vértices adjacentes (apenas arestas)	53

LISTA DE QUADROS

Quadro 1 – Características da modelagem	29
Quadro 2 – Parâmetros do modelo	31
Quadro 3 – Primeira fase do modelo – inicialização dos vértices e comunidades . .	32
Quadro 4 – Segunda fase do modelo – Construção dos lotes	34
Quadro 5 – Segunda fase do modelo – Processamento dos lotes	35
Quadro 6 – Segunda fase do modelo – função $\text{chooseCommunities}(v, \mathcal{G}_p)$	36
Quadro 7 – Segunda fase do modelo – função $\text{edgesWithin}(v, \mathcal{G}_p, C, n)$	38
Quadro 8 – Segunda fase do modelo – função $\text{edgesBetween}(v, \mathcal{G}_p, C_c, m)$	39
Quadro 9 – Terceira fase do modelo – adição final de arestas	40
Quadro 10 – Parâmetros básicos	42

LISTA DE TABELAS

Tabela 1 – Modularidade com os parâmetros básicos	43
Tabela 2 – Modularidade com $K = (9, 2, 2, 2)$	44
Tabela 3 – Modularidade com variação de K	44
Tabela 4 – Homogeneidade e homofilia com os parâmetros básicos	47
Tabela 5 – Homofilia e Homogeneidade com $K = (9, 2, 2, 2)$	47
Tabela 6 – Homofilia e Homogeneidade com θ variável	48

SUMÁRIO

1	INTRODUÇÃO	13
1.1	OBJETIVOS	14
1.2	ESTRUTURA	14
2	FUNDAMENTAÇÃO TEÓRICA	16
2.1	DEFINIÇÃO DE GRAFOS	16
2.2	REDES COMPLEXAS E COMUNIDADES	17
2.3	PROPRIEDADES DE REDES COMPLEXAS	20
2.3.1	Mundo pequeno, Anexação preferencial e Liberdade de escala	20
2.3.2	Cluster e comunidades	20
2.3.3	Homofilia e Homogeneidade de comunidades	21
2.3.4	Agrupamentos hierárquicos e sobreposições	21
2.4	TRABALHOS CORRELATOS	24
2.4.1	RTG: a recursive realistic graph generator using random typing	24
2.4.2	Generating Attributed Networks with Communities	25
3	MODELO	27
3.1	HIPÓTESE	27
3.1.1	Representação do grafo	27
3.1.2	Propriedades desejáveis do modelo	30
3.2	IMPLEMENTAÇÃO DO MODELO	31
3.2.1	Parâmetros	31
3.2.2	Inicialização do grafo	32
3.2.3	Processamento dos vértices	34
3.2.3.1	Seleção de comunidades	36
3.2.3.2	Geração de arestas	37
3.2.3.3	Atualização do estado	39
3.2.4	Adição final de arestas	40

4	RESULTADOS EXPERIMENTAIS	42
4.1	PRESENÇA DE COMUNIDADES	42
4.2	HOMOFILIA E HOMOGENEIDADE	45
4.2.1	Inércia	45
4.2.2	Distância esperada	46
4.2.3	Resultados	46
4.3	VISUALIZAÇÃO	48
5	CONCLUSÕES	54
5.1	EXTENSÕES	54
	REFERÊNCIAS	57

1 INTRODUÇÃO

Redes complexas, como definido por Metz et al. (2007), são grafos com uma topologia não trivial. Isso é, são grafos onde parte ou toda a informação de interesse não está contida nos vértices e arestas individualmente, mas em propriedades do conjunto de vértices e arestas. Esses grafos e as suas propriedades são aplicáveis as mais diversas áreas, como por exemplo na propagação de uma epidemia (STEGEHUIS; HOFSTAD; LEEUWAARDEN, 2016).

Segundo Girvan e Newman (2002), um dos sistemas do mundo real que se pode modelar em uma rede complexa é o conjunto de relações sociais. Uma modelagem simplista desse sistema é a representação de cada indivíduo como um vértice, e vértices adjacentes sendo pares de indivíduos que se conhecem. Nesse tipo de sistema pode se encontrar um subgrafo completo, denominado clique (FORTUNATO, 2010). Esse clique pode ser interpretado como uma propriedade relevante, indicando a presença de que um objeto composto por esses vértices.

Girvan e Newman (2002) também apontam que outros sistemas, como cadeias alimentares, cadeias de metabolização, redes de transmissão elétrica e redes de computadores podem ser representadas como redes complexas. Muitas vezes propriedades que se observam em redes complexas de um domínio estão presentes também nas redes complexas de outros domínios, mas com interpretações distintas sobre o objeto modelado. Fortunato (2010) indica isso nas múltiplas discussão e interpretações do que constitui uma comunidade em uma rede complexa, dividindo-se principalmente em características estruturais, e por semelhança de vértice.

Largeron et al. (2015) descrevem o que é chamado na literatura de um modelo de geração algorítmica de redes complexas no qual os vértices do grafo estão dispostos em uma nuvem de ponto e a distribuição deles em diferentes comunidades leva em conta sua posição espacial, e as arestas são construídas em função desse pertencimento a uma comunidade. Akoglu e Faloutsos (2009) descrevem um modelo mais primitivo, que não realiza a atribuição explícita de comunidades, mas que gera um grafo com essas comunidades. Ressalta-se, observando o trabalho de Fortunato (2010) de que há uma vasta literatura a respeito dos processos de detecção dessas comunidades. Observando-se a literatura da qual os trabalhos de Largeron et al. (2015), Akoglu e Faloutsos (2009) e Slota et al. (2019), é indicada a existência dos modelos necessários para a geração de redes complexas com comunidades. No entanto, propriedades adjacentes a presença de comunidades para os quais existe literatura a respeito da detecção parecem estar pouco presentes em modelos

de geradores de redes complexas.

Dessas propriedades de redes complexas para as quais não foram identificados modelos de geração destampam-se comunidades hierárquicas e comunidades sobrepostas. Comunidades hierárquicas, ou hierarquicamente ordenadas, são estruturas recursivas que podem ser encontradas em redes complexas, onde a característica de distribuição de arestas faz com que comunidades sejam compostas por sub comunidades. Comunidades sobrepostas são estruturas comunitárias onde vértices podem estar em mais de uma comunidade simultaneamente, de forma que a descrição das comunidades se dá em uma cobertura, não em uma partição (FORTUNATO, 2010).

Tendo esse contexto, este trabalho adaptou os modelos presentes na literatura de geração de redes complexas para a incorporação dessas propriedades, comunidades sobrepostas e comunidades hierárquicas.

1.1 OBJETIVOS

O objetivo desse trabalho é construir um modelo de geração algorítmica de redes complexas com comunidades hierarquicamente organizadas e com comunidades sobrepostas.

Os objetivos específicos são:

- a) a construção de um modelo algorítmico de geração de redes complexas que inclua a propriedade de comunidades;
- b) a especificação de uma *ground truth* de quais vértices pertencem a quais comunidades;
- c) a presença de comunidades hierárquicas e de comunidades sobrepostas nos grafos gerados;
- d) a representação dos vértices como uma nuvem de pontos, para a definição de semelhança de vértices por distância.

1.2 ESTRUTURA

Esse trabalho se estrutura em quatro capítulos sendo o primeiro uma introdução aos temas abordados, bem como a apresentação dos objetivos do trabalho.

O segundo capítulo apresenta a fundamentação teórica da pesquisa, descrevendo o estado da arte do objeto de estudo.

O terceiro capítulo discute o desenvolvimento do modelo algorítmico proposto, incluindo ferramentas e técnicas utilizadas. Também são apresentados os blocos de pseudo código do modelo.

O quarto capítulo compõe os dados obtidos na avaliação dos resultados, bem como quaisquer discussões de implementações futuras ou outras formas de continuação.

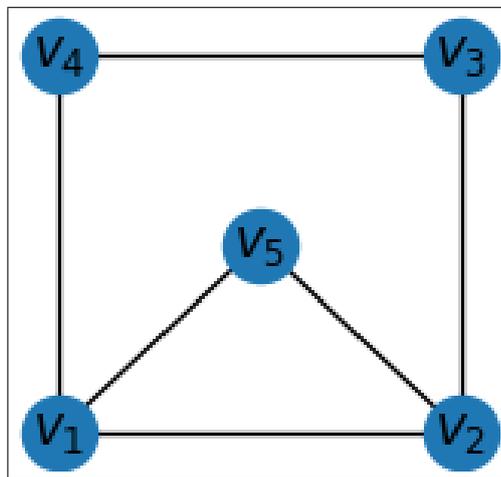
2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo está organizado em 4 seções. Na seção 2.1 é apresentada a definição básica de grafo. A seção 2.2 define redes complexas e comunidades. Na seção 2.3 aborda-se propriedades de redes complexas. A seção 2.4 apresenta o estado da arte dos modelos de geração de redes complexas.

2.1 DEFINIÇÃO DE GRAFOS

Grafos podem ser definidos como $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ onde \mathcal{V} é um conjunto dos vértices de \mathcal{G} e \mathcal{E} é um conjunto de pares não ordenados de vértices adjacentes em \mathcal{G} , i.e. as arestas.

Figura 1 – Exemplo de grafo



Fonte: elaborado pelo autor.

No exemplo da Figura 1, pode-se representar o mesmo grafo com $\mathcal{V} = \{v_1, v_2, v_3, v_4, v_5\}$ e $\mathcal{E} = \{\{v_1, v_2\}, \{v_2, v_3\}, \{v_3, v_4\}, \{v_4, v_1\}, \{v_1, v_5\}, \{v_2, v_5\}\}$. Dentro desse grafo, o subgrafo formado pelos vértices v_1, v_2 e v_5 é também um grafo completo¹. A esse conjunto de vértices que forma um subgrafo completo dá-se o nome de clique (FORTUNATO, 2010).

Variações dessas definições incluem dígrafos, isso é, grafos direcionados, onde as arestas são pares ordenados. Alternativamente, é possível a utilização de triplas para definir pares de vértices adjacentes, com um peso para essa aresta. Grafos onde encontram-se auto referências (arestas ligando um vértice a ele mesmo) ou arestas paralelas (onde um par de vértices está ligado por mais de uma aresta) são chamados de multigrafos.

¹ Um grafo onde todos os vértices estão conectados a todos os outros.

2.2 REDES COMPLEXAS E COMUNIDADES

A definição de clique pode ser utilizada como um ponto de partida para conceito de comunidade, partindo de uma perspectiva local (FORTUNATO, 2010). É facilmente observável, no entanto, que essa é uma definição muito limitante de comunidade. É raro que comunidades de pessoas apresentem tanta homogeneidade a ponto de todos os membros conhecerem todos os outros membros. De fato, Fortunato (2010) indica que a definição do que é uma comunidade varia de acordo também com o contexto de estudo, mas que algumas características são universais. Uma comunidade, dentro de qualquer definição, deve ser um subgrafo conexo², por exemplo (FORTUNATO, 2010).

Largeron et al. (2015) definem comunidade como uma classe de estrutura topológica comum a redes complexas, essas comunidades são categorizadas por terem uma densidade de vértices elevada. Essa densidade elevada pode ser interpretada como a uma maior chance de qualquer par de vértices descrever uma aresta para pares onde os dois membros pertençam a mesma comunidade. Shen et al. (2009) entendem que comunidades sejam estruturas que contenham múltiplos cliques dentro de si, e que essas comunidades se dispõem em uma estrutura recursiva. Akoglu e Faloutsos (2009) descrevem comunidades como estruturas modulares, onde nodos de uma rede formam grupos distintos entre si por que os membros do grupo tem maior chance de estarem conectados entre si do que estarem conectar com membros de outros grupos. Girvan e Newman (2002) definem “Cluster” e comunidade como duas propriedades distintas, o primeiro sendo a probabilidade de dois nodos ambos adjacentes a um terceiro serem também adjacentes entre si, e a segunda como sendo condutos de vértices densamente conectados entre si, e esparsamente conectados para além de si.

As definições são agrupadas em três classes distintas por Fortunato (2010): definição local; definição global; e definição por similaridade de vértice. Essas definições não são mutuamente exclusivas, mas também não são ortogonais uma a outra. Segundo Fortunato (2010), a definição local parte das características topológicas internas à comunidade. Nominalmente, isso significa a existência de um conjunto considerável de arestas internas a comunidade e um conjunto limitado de arestas para além da comunidade.

A definição global de comunidades é aplicada aos casos onde a presença de clusters é uma característica inerente ao grafo ao qual se está estudando (FORTUNATO, 2010). Essa propriedade inerente ao grafo pode ser definida como alguma propriedade dos vértices

² Um grafo conexo é um grafo onde para qualquer par de vértices u, v existe uma sequência de arestas $(u, w_0), (w_0, w_1), \dots, (w_{n-1}, w_n), (w_n, v)$ que os ligam

do objeto em questão e que partindo disso se atribui pertencimento a comunidades, ou ainda por comparação com um exemplo nulo (onde a propriedade não se encontraria) (FORTUNATO, 2010). No caso de comparação com um exemplo nulo, define-se uma comunidade pela característica de uma não ser presente dentro do que é chamado de “grafo aleatório” (FORTUNATO, 2010). Essa definição de um modelo nulo é crucial para o trabalho de Girvan e Newman (2002), o modelo nulo considerado é um modelo de um grafo construído a partir do grafo original onde os vértices tem o mesmo grau, mas a probabilidade de dois vértices estarem ligados é constante independente de quais os vértices.

A definição de comunidade por similaridade de vértice se baseia na tendência de que em muitas aplicações, membros de comunidades são mais similares entre si do que seria esperado de um conjunto do mesmo tamanho escolhido aleatoriamente (FORTUNATO, 2010). Essa definição se faz visível nos trabalhos de Akoglu e Faloutsos (2009) e de Largeron et al. (2015). Na observação desses dois trabalhos também é interessante o questionamento de como se define semelhança, Akoglu e Faloutsos (2009) representam os vértices como sequencias de caracteres³ de tamanhos variáveis em que a probabilidade de dois vértices estarem ligados é maior conforme mais caracteres eles compartilham; e Largeron et al. (2015) representam os vértices como pontos em um espaço n -dimensional e definem que vértices são mais semelhantes quando a distância euclideana deles é menor.

Também independente de qual definição de comunidade que se esteja utilizando, existem os conceitos de partição e cobertura. Segundo Fortunato (2010), uma partição é uma divisão dos vértices de um grafo tal que cada vértice pertença a um e exatamente um cluster. O caso de um vértice “livre”, não pertencendo a nenhuma comunidade, é trivialmente resolvido incluindo ele a comunidade com a qual ele tem mais adjacências. Mas, o caso de vértices que pertençam a mais de uma comunidade, i.e. comunidade que se sobreponham, é mais interessante. Fortunato (2010) define uma cobertura como uma divisão dos vértices em clusters onde cada vértice pertence a um ou mais clusters. Fortunato (2010) também descreve o conceito de comunidades hierárquicas, como sendo comunidades cuja estrutura interna também se organiza em clusters de escala menor do que o original.

Fortunato (2010) oferece também o conceito de “função de qualidade”, sendo uma função que mapeia uma partição para um espaço de comparação, usualmente em números reais, onde partições que mapeiem para valores maiores são consideradas melhores. Segundo

³ Caracteres são interpretados como letras do alfabeto, mas qualquer conjunto discreto de identificadores poderia ser utilizado.

Fortunato (2010), a função de qualidade mais comumente utilizada é a modularidade Q de Girvan e Newman (2002).

$$Q = \frac{1}{2m} \sum_{ij} \left(A_{ij} - \frac{K_i K_j}{2m} \right) \delta(C_i, C_j) \quad (2.1)$$

Onde m é a quantidade de arestas no grafo (de forma que $2m$ é a soma dos graus de todos os vértices). A_{ij} é a célula da tabela de adjacência entre os vértices i e j , leia-se 1 se os vértices forem adjacentes e 0 caso contrário. E $\frac{K_i K_j}{2m}$ é a probabilidade de i e j serem adjacentes em um modelo nulo. O modelo nulo neste caso é um grafo com os mesmos vértices, com os mesmos graus, mas que a chance de quaisquer dois vértices estarem anexos é uniformemente distribuída entre todos os vértices, o que faz com que não sejam esperadas estruturas comunitárias. Por fim a função $\delta(C_i, C_j)$ retorna um se C_i for igual a C_j , e zero em outro caso (FORTUNATO, 2010).

Essa função no entanto não se aplica adequadamente ao caso de comunidades sobrepostas ou comunidades hierárquicas, para tanto, é necessário utilizar a função de modularidade estendida, conforme desenvolvido por Shen et al. (2009).

$$EQ = \frac{1}{2m} \sum_i \sum_{v \in C_i, w \in C_i} \frac{1}{O_v O_w} \left[A_{vw} - \frac{K_v K_w}{2m} \right] \quad (2.2)$$

Onde O_u é a quantidade de comunidades onde o vértice u pode ser encontrado⁴. E $\frac{K_v K_w}{2m}$ é equivalente ao termo $\frac{K_i K_j}{2m}$ da função Q , bem como o termo A_{vw} é equivalente com o termo A_{ij} (FORTUNATO, 2010).

Notadamente, a fórmula Q e a fórmula EQ são iguais para partições, isso é, para uma distribuição de comunidades onde não sejam encontrados vértices em regiões de sobreposição, as duas funções descrevem o mesmo valor. Isso ocorre pois na fórmula EQ de Shen et al. (2009) tem-se também o termo escalar $1/O_v O_w$, em uma partição esse termo é sempre igual a um. As demais diferenças nas funções são apenas nos símbolos usados e na forma como se itera pelos membros do grafo, enquanto Q itera sobre todos os pares de vértices e filtra eles para considerar apenas os vértices que pertençam a uma mesma comunidade, EQ itera sobre as comunidades e dentro desse contexto pelos pares.

Esse escalonamento feito por $1/O_v O_w$, permite a aplicação da modularidade estendida para os casos de grafos com comunidades sobrepostas. Vértices que estejam em duas

⁴ Essa quantidade é contextual, para a execução recursiva dessa fórmula, as comunidade organizadas hierarquicamente são consideradas em momentos distintos.

comunidades contribuirão para a modularidade a partir das duas, mas tendo a magnitude da sua contribuição reduzida pela metade.

2.3 PROPRIEDADES DE REDES COMPLEXAS

Além de estruturas topológicas que podem ser denominadas comunidades, redes complexas tem algumas propriedades topológicas bastante comuns e relevantes. Algumas delas são: mundo pequeno; anexação preferencial; liberdade de escala; e homofilia.

2.3.1 Mundo pequeno, Anexação preferencial e Liberdade de escala

[Largeron et al. \(2015\)](#) descrevem a propriedade de mundo pequeno como a característica de um sistema de ter um diâmetro logaritmicamente proporcional a quantidade de vértices em um grafo. Isso é, a distância entre os dois vértices que estão a mais arestas de distância, denominada diâmetro, cresce logaritmicamente conforme observa-se grafos maiores de um mesmo sistema. Essa propriedade implica que em sistemas bastante grandes, é preciso uma quantidade relativamente pequena de saltos de nodo a nodo para se atingir qualquer membro do grafo. Essas propriedades serão exploradas a diante.

[Largeron et al. \(2015\)](#) definem a anexação preferencial como uma propriedade de um sistema em que vértices tendem a se ligar com outros vértices que sejam parecidos e que tenham grau elevado. A implicação é que dado um sistema onde se vai adicionar um vértice, a maior parte das arestas desse novo vértice devem ligá-lo a outro com grau igual ou maior do que o próprio.

Para atingir essa distribuição característica, o modelo de [Slota et al. \(2019\)](#) fazem com que os vértices se dividam em diferentes escalas, de forma que os vértices de uma escala se liguem apenas entre si e com os membros das escalas imediatamente vizinhas. De grafos com essa distribuição onde o grau relativo de dois vértices adjacentes tende a não apresentar saltos demasiadamente grandes, se diz que são livres de escala ([LARGERON et al., 2015](#)).

2.3.2 Cluster e comunidades

[Girvan e Newman \(2002\)](#) diferenciam explicitamente entre a definição de clusters e de comunidades. Os autores apontam um cluster como sendo um triângulo, em outras palavras, um subgrafo completo com três vértices. Essa definição aparentemente arbitrária

é relevante no entendimento do coeficiente de clusterização, definido como a proporção de quantas triplas conexas são triângulos.

$$C = \frac{3 \times (\text{número de triângulos do grafo})}{(\text{número de triplas conexas do grafo})} \quad (2.3)$$

Isso é, C é a probabilidade de que, dado um vértice w , dois vértices u e v adjacentes a ele sejam também adjacentes entre si.

A conceitualização de um cluster é relevante dentro do estudo de redes complexas na medida em que a implicação é de dois vértices serem ligados por compartilharem uma relação com um terceiro. O coeficiente C sendo igual a 1 implica que o grafo é um grafo completo (GIRVAN; NEWMAN, 2002). Mais que isso, esse coeficiente, dado um vértice, é a probabilidade de quaisquer dois vértices adjacentes a ele serem adjacentes entre si.

2.3.3 Homofilia e Homogeneidade de comunidades

Homofilia é a tendência em um sistema de que dois vértices conectados sejam semelhantes entre si (AKOGLU; FALOUTSOS, 2009). Essa propriedade está intimamente ligada a definição de comunidade por semelhança de vértices. Essa definição de semelhança é deliberadamente vaga, pois dentro de sistemas distintos é trivial imaginar funções de semelhança distintas. Observa-se que em grafos obtidos a partir de sistemas do mundo real, a topografia otimiza alguma função de forma aos vértices serem mais parecidos com os vértices aos quais são adjacentes (LARGERON et al., 2015).

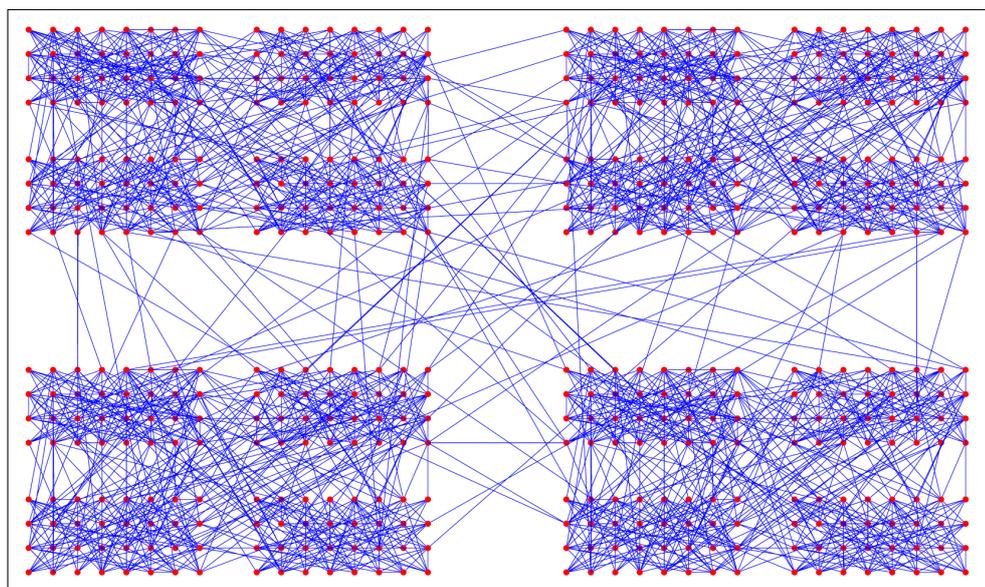
Homofilia como propriedade é intimamente ligada a uma outra característica que se observa de grafos do mundo real, em diferentes aplicações as comunidades tendem a ser mais homogêneas do que o grafo ao qual pertencem (LARGERON et al., 2015). Pode-se afirmar que um grafo onde isso ocorra tem a propriedade de “comunidades homogêneas”.

O exemplo da Figura 2 demonstra como a homofilia as vezes pode ter a presença visualmente verificada. Considerando que a posição dos vértices na imagem corresponde a duas características ortogonais, a distância entre os vértices pode ser interpretada como uma função de similaridade de dois vértices. Nesse caso é intuitivamente entendido que vértices mais parecidos se conectam mais do que vértices mais dissemelhantes.

2.3.4 Agrupamentos hierárquicos e sobreposições

No exemplo de grafo da Figura 2 é possível demonstrar um entendimento intuitivo de como comunidades se organizam. A estrutura topológica de grupos densamente conectados

Figura 2 – Exemplo de grafo com comunidades hierárquicas



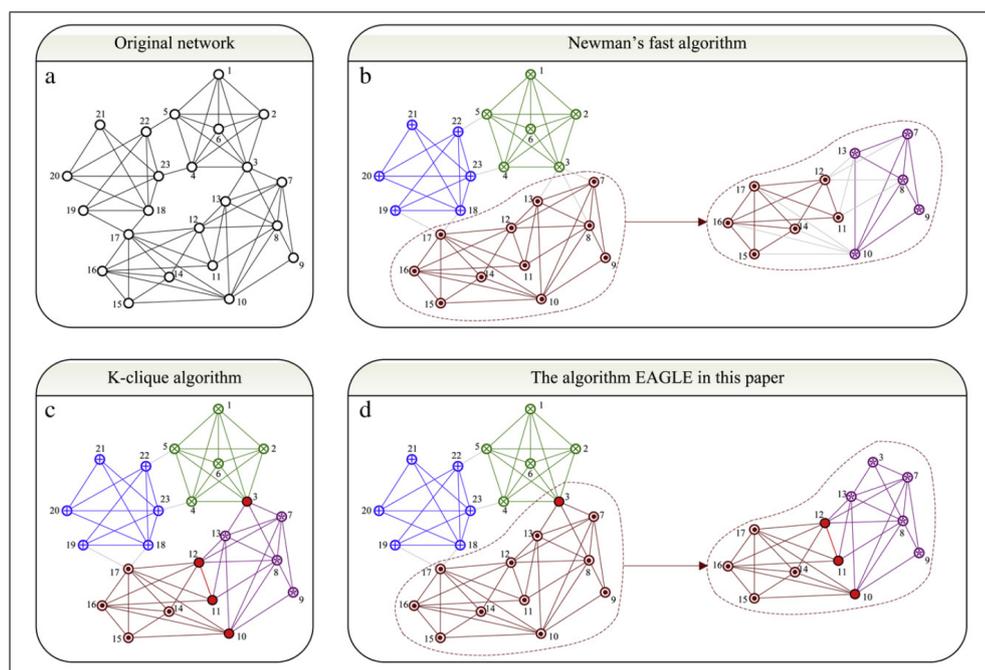
Fonte: Fortunato (2010).

fica visualmente identificável, onde cada quadrante contém uma comunidade coesa. Também visualmente acessível, cada comunidade desse exemplo tem uma estrutura interna auto semelhante.

Essa construção de estruturas topológicas recursivas é denominada por Girvan e Newman (2002) como “meta grupo”, onde as propriedades topológicas relativas a agrupamentos podem ser encontradas se repetindo em escalas menores dentro das componentes de escalas maiores. Comunidades podem funcionalmente ser compostas por comunidades menores. Esse mesmo conceito recebe uma outra nomenclatura nos trabalhos de Largeron et al. (2015), Shen et al. (2009) e Fortunato (2010), onde são descritas como comunidades hierárquicas.

Essas estruturas seguem uma característica recursiva, como demonstrado pelo processo de detecção proposto por Shen et al. (2009), podendo ser concebidos exemplos de sistemas com qualquer quantidade de níveis. E a elas também se aplica a compreensão de partição ou cobertura, na Figura 2 as comunidades de primeiro e de segundo nível caracteristicamente não compartilham vértices. No caso que demonstram Shen et al. (2009) um vértice pode pertencer a duas comunidades. Além disso ele pode pertencer a duas comunidades de níveis distintos. Na Figura 3, os resultados de Shen et al. (2009) são demonstrados no quadro a respeito do algoritmo EAGLE, o vértice denotado como 3 é compartilhado entre duas comunidades de primeiro nível, mas em uma delas o vértice 3 encontra-se como membro de uma comunidade de segunda ordem.

Figura 3 – Demonstração dos resultados de diferentes algoritmos de detecção em um grafo com comunidades hierárquicas e com sobreposição



Fonte: Shen et al. (2009).

Ressaltando que a exata definição de comunidade é altamente dependente do contexto (FORTUNATO, 2010), parece ser consenso na literatura que quando se consideram comunidades hierárquicas, todos os membros de uma comunidade de primeiro nível, devem fazer parte de uma das comunidades que compõe a primeira, como observado nos trabalhos de Fortunato (2010) e Shen et al. (2009). I.e.: nenhum vértice pertence exclusivamente a uma comunidade sem pertencer a alguma das sub-comunidades. Alternativamente claro, o exemplo da Figura 3 mostra que a implementação de Girvan e Newman (2002) (quadrante superior direito) é capaz de produzir partições recursivas (note-se a distinção entre uma cobertura e uma partição).

Essa distinção entre cobertura e partição implica também na definição de comunidades sobrepostas. Enquanto a definição de uma partição separa os vértices de um grafo em conjuntos de forma que cada vértice pertença a uma comunidade, uma cobertura abre a possibilidade de um mesmo vértice esteja em mais de um dos conjuntos. Em sistemas do mundo real que produzem redes complexas, é possível que comunidades compartilhem vértices pois alguma parte de um sistema é componente em dois grupos estruturalmente significantes (SHEN et al., 2009). Diz-se de duas comunidades que compartilham vértices que elas são comunidades sobrepostas.

O método de detecção de comunidades por K-cliques oferece alguma inspiração no

entendimento das propriedades de comunidades sobrepostas. [Fortunato \(2010\)](#) descreve que a forma como esse método trabalha é pivotando subgrafos completos do grafo. Isso é, dado que um K -clique é um subgrafo completo de k vértices, se dois k -cliques compartilham $k - 1$ vértices, todos os $K + 1$ vértices fazem parte de uma mesma comunidade. Com isso é possível propor que uma cobertura ideal deveria priorizar comunidades com grandes subgrafos completos internamente, mas de que os vértices da intersecção de duas comunidades deveriam participar de k -cliques distintos, preferencialmente não estando adjacentes. A ideia por trás disso é que se a intersecção de duas comunidades deveria ser parte da periferia das respectivas comunidades ([FORTUNATO, 2010](#)). Se a intersecção fosse tão densamente conexa quando o centro das duas comunidades, esses vértices não seriam mais valores intersectados entre duas comunidades distintas, e as comunidades seriam apenas uma só.

2.4 TRABALHOS CORRELATOS

Existe uma literatura muito prolífica de aplicações dos conceitos de redes complexas, como por exemplo o trabalho de [Stegehuis, Hofstad e Leeuwaarden \(2016\)](#), que fazem uma análise do espalhamento de doenças em uma rede com comunidades, onde é demonstrado que a presença das comunidades tem um efeito significativo. Muitos métodos para a detecção de comunidades foram propostos, como fica evidente na ampla revisão feita por [Fortunato \(2010\)](#). Por fim, existe uma literatura também cobrindo diferentes modelos para a geração de redes complexas com comunidades, apesar de bastante mais escassa.

2.4.1 RTG: a recursive realistic graph generator using random typing

[Akoglu e Faloutsos \(2009\)](#) desenvolveram um modelo que gera redes complexas com uma série de proporções que conhecidamente ocorrem em sistemas do mundo real. Isso é, construindo grafos que se assemelhem aos produzidos pelos sistemas do mundo real. O modelo também demonstradamente produz grafos com a presença de comunidades.

A implementação realizada por [Akoglu e Faloutsos \(2009\)](#) se baseia em um gerador de arestas que tem as probabilidades tendenciosas. Esse gerador é o que os autores chamam de um teclado recursivo, na realidade é uma matriz de possibilidades de escolha de uma característica discreta para a origem e o destino em simultâneo. Nominalmente, os vértices são uma sequência aleatória de caracteres de um conjunto finito de possibilidades, é repetidamente escolhido uma letra para o destino e uma para a origem simultaneamente. Com um parâmetro controlando um reforço que é feito para que a célula da matriz

escolhida a cada interação seja da diagonal principal, existe uma tendência de que vértices adjacentes tenham as mesmas letras nas mesmas posições.

Essas regras simples são o bastante para que o modelo de [Akoglu e Faloutsos \(2009\)](#) tenha como emergentes algumas das propriedades desejáveis em um modelo de geração de redes complexas. Além das proporções, mundo pequeno, anexação preferencial e homofilia, o sistema de [Akoglu e Faloutsos \(2009\)](#) gera comunidades homogêneas.

Existe também um interesse relevante em quanto os algoritmos de geração de redes complexas precisam em tempo. Nesse quesito, a implementação de [Akoglu e Faloutsos \(2009\)](#) apresenta algumas das características mais desejáveis, ele é totalmente paralelizável, significando que para a construção de um grafo com o dobro de arestas, é possível dobrar a quantidade de recursos de processamento e assim dobrar a quantidade de arestas produzidas em um tempo constante.

2.4.2 Generating Attributed Networks with Communities

[Largeron et al. \(2015\)](#) apresentam um modelo algorítmico de geração de redes complexas com a explícita e deliberada definição de quais condicionantes são utilizadas para afetar quais propriedades. Para isso, em uma fase inicial, gera-se uma nuvem de pontos com uma distribuição aleatória e uma amostra dessa população é usada para inicializar as comunidades. Essa amostragem é processada em um algoritmo k-means, gerando os clusters iniciais, e as arestas iniciais são geradas. Uma segunda fase processa os demais vértices, escolhendo qual a comunidade serão inseridos se baseando na distância euclidiana (homofilia), e gerando os vértices baseado na distribuição de graus. Numa última fase, opcional, é realizada a introdução de novas arestas, essas arestas são escolhidas de forma a fecharem triângulos, para aumentar o coeficiente de clusterização C .

Notadamente, a modelagem de grafo utilizada por [Largeron et al. \(2015\)](#) é $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$ onde \mathcal{A} é um conjunto de atributos dos vértices, de forma que cada $v \in \mathcal{V}$ é um vetor de valores v_A . A modelagem tem acompanhando o grafo também um conjunto \mathcal{P} , composto por conjuntos de vértices, isolando assim as partições. Essa abordagem oferece uma funcionalidade bastante desejável, que é gerar um catálogo de qual vértice pertence a qual comunidade, dessa forma criando um “ground truth” contra o qual o desempenho de alguns algoritmos pode ser testado. Considerando os trabalhos que utilizam análise de redes complexas como [Stegehuis, Hofstad e Leeuwaarden \(2016\)](#), a possibilidade de regerar um grafo com características topológicas conhecidas, podendo-se manipular o coeficiente de clusterização por exemplo, existem um conjunto de possíveis análises com

relevância acadêmica. Para muitas dessas análises, as possibilidades de parametrização do modelo de [Largeron et al. \(2015\)](#) parece ser interessante. É feita também uma discussão de performance por parte de [Largeron et al. \(2015\)](#), mas é possível fazer algumas críticas a forma como o modelo foi desenhado, que previnem a paralelização do processo de construção dos grafos.

3 MODELO

Este trabalho propõe um modelo algorítmico de geração de redes complexas que produza grafos mais realistas. Realista nesse contexto é entendido como tendo a presença de um conjunto de propriedades que se observam em grafos que modelam sistema do mundo real. A proposta desse trabalho é a extensão do modelo de [Largeron et al. \(2015\)](#) para a construção de grafos em que se apresentem também as propriedades de comunidades hierárquicas e comunidades sobrepostas. A modelagem proposta se baseia na construção de uma cobertura (em oposição à construção de uma partição) recursiva, de forma semelhante ao que é produzido pelo algoritmo de detecção proposto por [Shen et al. \(2009\)](#).

3.1 HIPÓTESE

A hipótese a ser avaliada é a de que a extensão do modelo descrito por [Largeron et al. \(2015\)](#) para a introdução de comunidades hierárquicas e sobrepostas pode ser feita com a aplicação recursiva de uma escolha de comunidades. Isso é, modificando o modelo para que a construção inicial das comunidades se dê de forma recursiva, e modificando o processo de escolha de comunidade para que seja possível a escolha de um par de comunidades, é esperado que se mantenham as propriedades se mantenham.

Essa modificação do modelo não deve de afetar a capacidade dele de gerar grafos com comunidades estruturalmente relevantes, e ainda assim homogêneas. Para tanto, estende-se a representação clássica de grafo para a inclusão de uma cobertura, bem como a delimitação dos vértices como vetores em um espaço de p dimensões.

3.1.1 Representação do grafo

O modelo proposto gera um grafo com atributos e uma cobertura de comunidades hierarquicamente dispostas. Este grafo é representado da seguinte maneira.

$$\begin{aligned}
 \mathcal{G} &= (\mathcal{V}, \mathcal{E}, C_n) \\
 \mathcal{V} &\subset \mathbb{Q}^p \\
 \mathcal{E} &\subset \{v_1, v_2 \mid v_1 \in \mathcal{V}, v_2 \in \mathcal{V}\} \\
 C_0 &\subset \{v \mid v \in \mathcal{V}\} \\
 C_n &= \{C_{n-1_0}, C_{n-1_1}, \dots, C_{n-1_m}\}
 \end{aligned} \tag{3.1}$$

Onde \mathcal{G} é o grafo, \mathcal{V} é o conjunto de vértices, \mathcal{E} o conjunto de arestas e C_n o conjunto

inicial da representação da cobertura. Os vértices em \mathcal{V} são uma nuvem de pontos em um espaço de p dimensões, cada um sendo um vetor com p valores racionais (representados com ponto flutuante). As arestas em \mathcal{E} são simples pares não ordenados de vértices. A cobertura C_n é uma estrutura recursiva de conjuntos com n níveis, onde o conjunto C_n é composto por m conjuntos C_{n-1} . No último nível, o conjunto C_0 é composto por vértices do grafo, esses conjuntos são as comunidades *folha*, significando que elas não são compostas por outras comunidades.

Essa estrutura recursiva é a representação das comunidades hierárquicas, onde os vértices que pertencem a uma comunidade C , denotado V_C são os membros do conjunto união dos membros de C , isso é, $\bigcup_{S \in C} V_S$. Como característica dessa modelagem, os vértices da comunidade representada por C_n são a totalidade dos vértices do grafo, portanto $V_{C_n} = \mathcal{V}$. Isso implica que para qualquer vértice em \mathcal{V} , ele deve de estar presente pelo menos uma comunidade folha¹. Ao estar presente em uma comunidade folha, o vértice é considerando também parte de todas as comunidades compostas por esta comunidade folha.

A cardinalidade de cada um dos conjuntos que formam a cobertura é variável de acordo com o nível, isso é, dado um nível x todas as coberturas C_{n-x} possuem a mesma quantidade de elementos, mas conjuntos de níveis distintos podem possuir quantidades de elementos distintas. É considerada também a existência de um vetor, denotado K , em um espaço de $n - 1$ dimensões, que denota a cardinalidade das coberturas compostas por outras coberturas. Todas as comunidades do grafo \mathcal{G} contém pelo menos um membro.

Uma característica de notação é a função flat que mapeia uma comunidade para um conjunto de quais as comunidades que a compõe. Para efeito de notação $\text{flat}(C_n)$ é um conjunto com todas as comunidades do grafo, incluindo a comunidade global. O mapa L relaciona cada comunidade com a quantidade de ancestrais que a comunidade possui, ou seja, com a quantidade de comunidades que ela compõe. As notações utilizadas serão L_C , leia-se nível de C , ou L , leia-se níveis. A raiz da cobertura é o nível zero ($L_{C_n} = 0$) e o nível de um nó folha é igual a n ($L_{C_0} = n$). As características que devem ser verdadeiras se um grafo for representado nesta modelagem encontram-se descritas no [Quadro 1](#).

Além dessa estruturação de \mathcal{G} , o modelo faz uso de uma segunda representação do grafo, denotada $\mathcal{G}_p = (\mathcal{V}, \mathcal{E}, C_n, R)$. Ela representa um estado parcial do grafo sendo gerado. Neste estado parcial o grafo não necessariamente é conexo e vértices com grau zero

¹ Pela organização da cobertura em uma estrutura de árvore refere-se a comunidades do último nível, que não possuem subdivisões internas, como comunidades folha.

Quadro 1 – Características da modelagem

Característica	Formalismo
Para toda a comunidade C , se ela não for folha, a função flat dela é a união de C com a função flat de seus componentes.	$\forall C(L_C < n \implies \text{flat}(C) = C \cup \bigcup_{S \in C} \text{flat}(S))$
Para toda a comunidade C , se ela for folha, a função flat dela é um conjunto contendo apenas ela.	$\forall C(L_C = n \implies \text{flat}(C) = \{C\})$
A comunidade raiz engloba todos os vértices do grafo.	$V_{C_n} = \mathcal{V}$
Para todas as comunidades C , se C não for folha, os vértices englobados em C são a união dos vértices englobados em seus componentes	$\forall C(L_C > 0 \implies V_C = \bigcup_{S \in C} V_S)$
Para todas as comunidades C , se C for folha, os vértices englobados em C são seus membros.	$\forall C(L_C = 0 \implies V_{C_0} = C_0)$
Para todos os vetores do grafo, existe uma comunidade folha a qual ela pertence.	$\forall v(\exists C(v \in C \wedge L_C = n))$
Pra todo vértice v , pra todo l , existe uma comunidade C que contenha o vértice e seja do nível l .	$\forall v \forall l(\exists C(v \in V_{C_x} \wedge L_C = l))$
Todos as comunidades não folha tem a mesma quantidade de componentes se forem do mesmo nível, a cardinalidade de uma comunidade não folha é expressa num vetor K .	$\exists K(K \in \mathbb{I}^{n-1} \wedge \forall C(L_C < n \implies K_{L_C} = C)$
Toda a comunidade tem pelo menos uma componente e engloba pelo menos um vértice.	$\forall C(C \geq 1 \wedge V_C \geq 1)$

Fonte: elaborado pelo autor.

não fazem parte de nenhuma comunidade definida na cobertura C_n . Nessa representação também é incluso um novo dado R , que identifica os representantes de uma determinada comunidade, isso é, os membros eleitos durante o processo com quem se compara um vértice v ao considerar introduzir este à comunidade. Esse estado \mathcal{G}_p mantém algumas das características de notação descritas no [Quadro 1](#). Notadamente as funções $\text{flat}(C)$, L_C e V_C são usadas no modelo em contextos em que C é definida como uma comunidade em \mathcal{G}_p .

3.1.2 Propriedades desejáveis do modelo

Largeron et al. (2015) implementam um modelo algorítmico de geração de redes complexas que mantém uma série de propriedades desejáveis. Como a implementação proposta se baseia no modelo de Largeron et al. (2015), é desejável que as propriedades sejam mantidas. Nominalmente, são elas:

- a) mundo pequeno: o diâmetro das redes complexas geradas pelo modelo deve ter uma relação logarítmica com a quantidade de vértices no modelo;
- b) distribuição de graus em lei de potência: Os graus dos vértices devem estar distribuídos com uma lei de potência;
- c) homofilia: O grafo gerado deve apresentar uma tendência de priorização da adjacência com vértices semelhantes;
- d) estrutura de comunidades: O grafo gerado deve ter comunidades, conforme etiquetadas na cobertura, de forma que todo vértice pertença a uma ou mais comunidades, e as comunidades se organizem em uma estrutura hierárquica;
- e) comunidades homogêneas: As comunidades devem ser coesas não apenas na perspectiva topológica, mas em similaridade.

Para tanto, a abordagem do modelo é a construção explícita das comunidades com base na similaridade dos vértices. Para isso, a similaridade dos vértices é definida com base na distância euclidiana dos vetores de atributos dos vértices. As arestas do grafo são definidas com base nas comunidades das quais o vértice faz parte.

Essa implementação visa garantir a homogeneidade das comunidades e a homofilia ao selecionar os membros das comunidades estocasticamente preferindo vértices com menor distância euclidiana. A construção das arestas é feita priorizando a introdução de vértices a vértices com mais arestas dentro da comunidade, de forma a reforçar a distribuição de graus em lei de potência e a estrutura de comunidade, bem como a propriedade de mundo pequeno.

É introduzida também a conceitualização de ortogonalidade de comunidades. É trivial a identificação, em sistemas do mundo real, de agrupamentos que se sobrepõem devido ao compartilhamento de características distintas, isso é, cada uma das comunidades na área sobreposta tem como definição uma característica distinta. Em uma definição de comunidade por semelhança de vértice, dado um sistema onde os vértices são caracterizados por dois ou mais características independentes, cada comunidade pode ter uma semelhança não em função da soma das características, mas de uma categorização em específico.

Considerando comunidades de indivíduos caracterizados por sua área de atuação e por sua crença religiosa, assumindo que não há uma influência direta entre essas duas características, uma comunidade de membros de uma religião poderia estar sobreposta a uma comunidade de profissionais de uma determinada área. Essas duas comunidades seriam ortogonais.

3.2 IMPLEMENTAÇÃO DO MODELO

A implementação do modelo se divide em três etapas, que consumindo um conjunto de parâmetros produzem um grafo conforme a representação previamente descrita. As etapas inicialmente constroem uma nuvem de pontos e os cluster iniciais para as comunidades. Com esses vértices e as comunidades iniciais, iterativamente são adicionados novos vértices às comunidades, e são geradas arestas nesse processo. Por fim, as arestas finais são adicionadas.

3.2.1 Parâmetros

As propriedades descritas podem ser controladas utilizando uma série de parâmetros. Os parâmetros seguem descritos no [Quadro 2](#). Eles são uma adaptação bastante direta dos parâmetros do modelo de [Largeron et al. \(2015\)](#), a diferença mais significativa é no parâmetro K , que é um vetor multidimensional de inteiros maiores que 1. Isso se deve à construção de uma árvore de comunidades hierarquicamente aninhadas.

Quadro 2 – Parâmetros do modelo

Parâmetro	Descrição
$N \in \{n \in \mathbb{N} \mid n \geq 1\}$	Quantidade de vértices.
$E_{\text{wth}}^{\max} \in \{i \in \mathbb{N} \mid i \geq 1\}$	Número máximo de arestas (internas à comunidade) inseridas a um vértice ao introduzir ele a uma comunidade.
$E_{\text{btw}}^{\max} \in \{i \in \mathbb{N} \mid i \geq 1\}$	Número máximo de arestas (externas à comunidade) inseridas a um vértice ao introduzir ele as comunidades.
$MTE \in \{m \in \mathbb{N} \mid m \geq 1\}$	Número mínimo de arestas no grafo produzido.
$\mathcal{A} \in \{a \in \mathbb{Q} \mid a > 0\}^{ \mathcal{A} }$	Vetor de desvios padrão dos atributos dos vértices.
$K \in \{k \in \mathbb{N} \mid k \geq 2\}^{ K }$	Vetor de quantidade de comunidades por nível
$\theta \in \{t \in \mathbb{Q} \mid 0 \geq t \geq 1\}$	Valor de interpolação entre homogeneidade por distância euclidiana e distância por ortogonalidade de comunidade.
$\text{NbRep} \in \{n \in \mathbb{N} \mid n \geq 1\}$	Número de representantes por comunidade.

Fonte: elaborado pelo autor.

3.2.2 Inicialização do grafo

A primeira fase do algoritmo é a inicialização dos vértices e das comunidades. Conforme definido no [Quadro 3](#).

Quadro 3 – Primeira fase do modelo – inicialização dos vértices e comunidades

```

1 Output:  $\mathcal{G}_p = (\mathcal{V}, \mathcal{E}, C_n, R)$ 
2  $\mathcal{V} \leftarrow \emptyset$ 
3  $\mathcal{E} \leftarrow \emptyset$ 
4 while  $|\mathcal{V}| < N$  do
5 begin
6    $v \leftarrow (\mathcal{N}(0, \sigma_{\mathcal{A}_0}), \mathcal{N}(0, \sigma_{\mathcal{A}_1}), \dots, \mathcal{N}(1, \sigma_{\mathcal{A}_{|\mathcal{A}|-1}}))$ 
7    $\mathcal{V} \leftarrow \mathcal{V} \cup \{v\}$ 
8 end
9 Function  $\text{cover}(l, p)$ 
10 begin
11   if  $l = |K|$  then
12     begin
13       for  $v \in p$  do
14         begin
15            $p' \leftarrow \{v' \in p \mid \{v, v'\} \notin \mathcal{E} \wedge \{v, v'\} \notin \mathcal{E} \wedge v' \neq v\}$ 
16            $s \leftarrow \text{RandUni}([1, |p'|])$ 
17           for  $v' \in \text{Sample}(p', s)$  do  $\mathcal{E} \leftarrow \mathcal{E} \cup (v, v')$ 
18         end
19       return  $p$ 
20     end
21      $s \leftarrow \text{NbRep} \times \prod_{i=l}^{|K|-1} K_i$ 
22      $p' \leftarrow \text{Sample}(p, \min\{s, |p|\})$ 
23      $k \leftarrow K \text{ Medoids}(p', K_l)$ 
24      $c \leftarrow \{\text{cover}(l+1, q) \mid q \in k\}$ 
25      $p' \leftarrow \{\text{RandUni}(c') \mid c' \in c\}$ 
26      $\mathcal{E} \leftarrow \mathcal{E} \cup \{\{p'_i, p'_{i+1}\} \mid i \in \{1, 2, \dots, |p'|-1\}\}$ 
27     return  $c$ 
28   end
29  $C_n = \text{cover}(0, \mathcal{V})$ 
30
31 for  $C_i \in \text{flat}(C_n)$  do
32   if  $C_i = \text{flat}(C_i)$  then  $R_{C_i} \leftarrow C_i$ 
33   else  $R_{C_i} \leftarrow \emptyset$ 
34  $\mathcal{G}_p \leftarrow (\mathcal{V}, \mathcal{E}, C_n, R)$ 
35 return  $\mathcal{G}_p$ 

```

Fonte: elaborado pelo autor.

O processo da inicialização se divide em gerar a nuvem de pontos e inicializar as comunidades. A linha 2 inicializa \mathcal{V} com um conjunto vazio, e o laço de repetição das linhas 4 à 8 insere vetores neste conjunto enquanto ele tiver menos de N membros. O

vetor em si é definido como uma série de distribuições aleatórias com o centro em zero e o desvio padrão informado pelo parâmetro \mathcal{A} .

O processo de geração das estruturas de comunidade é mais complexo, exigindo uma função para possibilitar recursividade. A função `cover` tem como condicionante a característica de que a comunidade que se está processando é ou não folha, isso é, se ela possuirá ou não subdivisões internas. Na linha 11 é feita essa ramificação, considerando que l , um parâmetro de controle que é incrementado a cada chamada recursiva. Se l for igual à cardinalidade de k , isso indica que se está processando o último nível a ser gerado, uma folha, o comportamento deixa de ser recursivo.

O último nível gerado compõe um conjunto de arestas entre os membros da comunidade folha. Nas linhas 13 até 15 é iterado sobre os vértices, os vértices com quem é possível formar arestas, nomeado p' , são definidos como os vértices em p diferentes de v com quem v não é adjacente. Nas linhas 16 e 17 uma quantidade aleatória das arestas possíveis são construídas.

As funções `RandUni` e `Sample` são duas funções de escolha aleatória uniformes. `Sample(P, l)` escolhe um subconjunto de P com l elementos uniformemente distribuído, i.e., todos os membros de P tem a mesma chance de estar presente no conjunto construído. `RandUni(P)` funciona da mesma forma, mas retorna um único membro de P .

No caso de não ser uma comunidade folha, o processo de construção da comunidade encontra-se nas linhas 21 até 26. Para isso, primeiramente é definido um tamanho de amostragem s . Esse tamanho é definido como um produto dos valores de K , filtrando para o nível atual em diante. Com isso, buscase uma amostra p' , com tamanho s ou o valor máximo possível se s for maior que a quantidade de membros em p .

Com essa amostra, realiza-se um agrupamentos utilizando o algoritmo *K Medoids* (LARGERON et al., 2015). Nesses clusters iniciais é realizada a chamada recursiva da função `cover`, que faz a construção da comunidade composta pelos vértices do cluster. Com as comunidades definidas e agrupadas no conjunto c , que representa a comunidade que se está processando, é realizada a introdução de arestas para que a comunidade seja conexa. Assumindo que todas as comunidades geradas por meio da função `cover` sejam conexas, é construído um caminho que liga um membro de cada comunidade. A função se conclui retornando a comunidade criada.

A chamada original para a função `cover(l, p)` é feita com l sendo zero e p sendo a nuvem de pontos. Por fim, o processo também realiza a atribuição dos representantes de cada comunidade folha como sendo a totalidade dos membros da comunidade, e mantendo

as demais comunidades sem representantes.

3.2.3 Processamento dos vértices

O processamento dos vértices, isso é, a sistemática introdução deles à comunidades bem como a definição de arestas que reforcem a comunidade considerando os membros introduzidos, é esperado que seja mais custoso. Esse processo tem um forte componente do custo relacionado à quantidade de vértices do grafo final, isso é, o parâmetro N . Para tanto, as principais distinções do modelo proposto para com o modelo de [Largergeron et al. \(2015\)](#) são para possibilidade de paralelização do processo.

Quadro 4 – Segunda fase do modelo – Construção dos lotes

1	Output: $B \subset \{ B' \mid B' \subset \mathcal{V} \}$
2	
3	$B' \leftarrow \{ v \in \mathcal{V} \mid \neg \exists v' (\{ v, v' \} \in \mathcal{E}) \}$
4	$B_s \leftarrow \left\lfloor \frac{ \text{flat}(C_n) }{2} \right\rfloor$
5	$B'_s \leftarrow (B_s, 2B_s, 4B_s, \dots, \lceil \log_2 \frac{5000}{B_s} \rceil B_s)$
6	$B \leftarrow \emptyset$
7	for $s \in B'_s$ do
8	begin
9	$B_i \leftarrow \{ \text{sample}(B', s) \}$
10	$B' \leftarrow B' \setminus B_i$
11	$B \leftarrow B \cup B_i$
12	end
13	while $ B' > 5000$ do
14	begin
15	$B_i \leftarrow B \cup \{ \text{sample}(B', \text{Rand}_{\text{Uni}}(5000, 5001, 5002, \dots, 10000)) \}$
16	$B' \leftarrow B' \setminus B_i$
17	$B \leftarrow B \cup B_i$
18	end
19	$B \leftarrow B \cup B'$

Fonte: elaborado pelo autor.

A implementação feita trabalha com lotes que serão processados sequencialmente, os membros de cada lote serão processados de forma assíncrona. Para isso, as arestas adicionadas no processamento de cada vértice individual não serão consideradas como existentes no processamento de vértices do mesmo lote. No mesmo sentido, os vértices processados individualmente não serão considerados como membros de comunidade alguma enquanto o lote é processado. Em outros termos, é estabelecido um estado do grafo que será considerado imutável, e cada vértice será processado com o mesmo estado, acumulando as informações geradas no processamento e cada vértice para a construção de um novo estado

quando da conclusão de todos os vértices do lote. Para tanto, os lotes são construídos e processados de acordo com o [Quadro 4](#)

Os lotes tem tamanhos definidos, o tamanho base B_s é dado pela quantidade total de comunidades $|\text{flat}(C_n)|$. E, a sequência de tamanhos B'_s é dada pelas potências de dois, multiplicadas por B_s , até o primeiro valor que seja maior ou igual a cinco mil. Os lotes gerados depois dessa primeira sequência tem tamanho definido aleatoriamente entre cinco mil e dez mil. O consumo desses lotes para a construção das comunidades hierárquicas e sobrepostas é implementada conforme descrição no [Quadro 5](#).

Quadro 5 – Segunda fase do modelo – Processamento dos lotes

```

1 Output:  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, C_n)$ 
2
3 Function introduce( $v, \mathcal{G}_p$ )
4 begin
5    $C_c \leftarrow \text{chooseCommunities}(v, \mathcal{G}_p)$ 
6
7    $t_{\mathcal{E}} \leftarrow \emptyset$ 
8   for  $C_i \in C_c$  do  $t_{\mathcal{E}} \leftarrow t_{\mathcal{E}} \cup \text{edgesWithin}(v, \mathcal{G}_p, C_i, |C_c|)$ 
9    $t_{\mathcal{E}} \leftarrow t_{\mathcal{E}} \cup \text{edgesBetween}(v, \mathcal{G}_p, C_c, |t_{\mathcal{E}}|)$ 
10   $t_C \leftarrow \{(v, C_i) \mid C_i \in C_c\}$ 
11  return  $t_C, t_{\mathcal{E}}$ 
12 end
13
14 for  $b \in B$  do
15 begin
16   for  $v \in b$  do # esse laço pode ser realizado paralelamente
17   begin
18      $t_C, t_{\mathcal{E}} \leftarrow \text{introduce}(v, \mathcal{G})$ 
19      $T_C \leftarrow T_C \cup t_C$ 
20      $T_{\mathcal{E}} \leftarrow T_{\mathcal{E}} \cup t_{\mathcal{E}}$ 
21   end
22    $\mathcal{E} \leftarrow \mathcal{E} \cup T_{\mathcal{E}}$ 
23    $C_n \leftarrow \text{buildCover}(C_n, T_C)$ 
24    $R \leftarrow \text{electRepresentants}(C_n)$ 
25 end
26  $\mathcal{G} \leftarrow (\mathcal{V}, \mathcal{E}, C_n)$ 
27 return  $\mathcal{G}$ 

```

Fonte: elaborado pelo autor.

Nesse processo, são consumidos os lotes, de forma que a função $\text{introduce}(v, \mathcal{G}_p)$ possa ser executada paralelamente. Isso é, fazendo uso de um estado $\mathcal{G}_p = (\mathcal{V}, \mathcal{E}, C_n, R)$ imutável, o processamento de cada vértice de um lote pode ser feito de forma distribuída. Ao final desse processamento, é trivial acumular os dados gerados por cada execução, e

com esses dados gerar um novo estado do grafo.

A implementação da função introduz em si se dá pela escolha de um conjunto de comunidades C_c por meio da função $\text{chooseCommunities}(v, \mathcal{G}_p)$, seguido pela geração de arestas. São geradas as arestas internas à comunidade dentro de uma função $\text{edgesWithin}(v, \mathcal{G}_p, C_i)$ e as externas às comunidades $\text{edgesBetween}(v, \mathcal{G}_p, C_c, |t_\varepsilon|)$. Ambas as funções de geração de arestas retornam um conjunto de pares não ordenados.

Assumindo que as implementações dessas funções ocorrem de forma a reforçar as propriedades desejadas, isso é, as comunidades escolhidas serem estocasticamente selecionadas para que os membros sejam semelhantes ao vértice v ; e as arestas criadas reforçarem as características estruturais da comunidade. É esperado que as propriedades de homofilia, comunidades homogêneas, entre outras emergem naturalmente.

3.2.3.1 Seleção de comunidades

A seleção de comunidades, realizada dentro da função $\text{chooseCommunities}(v, \mathcal{G}_p)$ visa reforçar as características de semelhança dos membros das comunidades. Para tanto, a comunidade escolhida deveria, estocasticamente, ter os vértices mais semelhantes ao vértice v . No entanto, para essa comparação não é viável, dado o custo de processamento, comparar todos os vértices já processados com o vértice v . Realiza-se portanto uma amostragem, na qual para cada comunidade é considerado um número de representantes, que devem caracterizar significativamente o perfil dos membros da comunidade.

Os representantes de uma comunidade, denota-se R_C , são, em um primeiro momento definidos como a totalidade dos membros da comunidade se essa for uma comunidade folha. A cada lote processado, o processo de construção de um novo estado para o grafo, realiza uma nova seleção dos representantes de cada comunidade.

Quadro 6 – Segunda fase do modelo – função $\text{chooseCommunities}(v, \mathcal{G}_p)$

```

1 Function chooseCommunities( $v, \mathcal{G}_p$ )
2    $P \leftarrow$  o conjunto  $\{(C_i, r) \mid C_i \in \text{flat}(C_n) \mid r \in R_{C_i}\}$  ordenado pela função  $d$ 
3    $C \leftarrow \text{Rand}_{\text{PL}}(P)$ 
4    $(C', C'') \leftarrow (C, C)$ 
5   while  $C' \neq \text{flat}(C')$  do  $C' \leftarrow \text{Rand}_{\text{PL}}(\{p \in P \mid p_0 \in \text{flat}(C'_0) \wedge p \neq C'\})$ 
6   while  $C'' \neq \text{flat}(C'')$  do  $C' \leftarrow \text{Rand}_{\text{PL}}(\{p \in P \mid p_0 \in \text{flat}(C''_0) \wedge p \neq C'' \wedge p \neq C'\})$ 
7   return  $\{C'_0, C''_0\}$ 
8 end

```

Fonte: elaborado pelo autor.

Conforme apresentado no [Quadro 6](#), tendo os representantes definidos, é utilizada a função Rand_{PL} , definida por [Largeron et al. \(2015\)](#), para escolher um par ordenado de comunidade e representante. Essa função escolhe um membro de um conjunto ordenado de cardinalidade m com a distribuição $x \mapsto \frac{x^{-2}}{\sum_{i=1}^m i^{-2}}$. Para tanto, os pares ordenados de comunidade e representante são organizados pela função de semelhança d .

$$d(v, v') = (1 - \theta)|v - v'| + \theta|v_a - v'_a| \quad (3.2)$$

Onde θ é o parâmetro do modelo, e a é o eixo em que a comunidade com a qual se contextualiza essa distância é menos esparsa. Isso é, a função d é dependente do contexto de qual comunidade se está comparando, e θ controla a proporção entre considerar a distância euclidiana ou a diferença em um eixo específico. $\theta = 0$ indicando que é considerada apenas a distância euclidiana e $\theta = 1$ indicando que não se considerará ela.

O eixo a utilizado na função d é dependente de comunidade e é definido como a dimensão em que a comunidade é menos esparsa. Para tanto, identifica-se que a função de inércia de um conjunto de pontos, como utilizado por [Largeron et al. \(2015\)](#), pode ser expressa como uma soma da inércia consistindo apenas uma dimensão por vez.

$$\sum_{v \in C} |g - v|^2 = \sum_{a=0}^n \sum_{v \in C} (g_a - v_a)^2 \quad (3.3)$$

Considerando n como o índice do último componente dos vetores em C e g sendo o centro de gravidade de g . O eixo considerado em d é aquele com a menor contribuição para a inércia.

Com a ordenação definida, a função escolhe uma comunidade C à qual adicionar o vértice v . Se essa comunidade for uma comunidade folha, os dois laços de repetição não serão executados e o conjunto $\{C'_0, C''_0\}$ terá apenas uma comunidade ($C' = C''$). Caso a comunidade possua sub-comunidades, o processo de escolha executado iterativamente com as variáveis C' e C'' , restringindo para que sejam escolhidas apenas comunidades contidas nas variáveis. É restringido também, na seleção de C'' que este não seja igual à C' .

3.2.3.2 Geração de arestas

O processo de geração das arestas internas às comunidades às quais se está adicionando o vértice se dá conforme descrito no [Quadro 7](#). Primeiramente, é definida uma quantidade máxima de arestas m como sendo o mínimo entre o parâmetro $E_{\text{wth}}^{\text{max}}$ e a quan-

tidade de vértices já presentes na comunidade C . Essa quantidade máxima é escalonada de acordo com a quantidade de comunidades em que o vértice v será adicionado, o máximo é o próximo inteiro maior ou igual a enésima raiz de m . A quantidade final de arestas a serem selecionadas é definida com a função Rand_{PL} .

Quadro 7 – Segunda fase do modelo – função $\text{edgesWithin}(v, \mathcal{G}_p, C, n)$

```

1 Function edgesWithin( $v, \mathcal{G}_p, C, n$ )
2    $m \leftarrow \min(E_{\text{wth}}^{\max}, |V_C|)$ 
3    $e \leftarrow \text{Rand}_{\text{PL}}(1, 2, 3, \dots, \lceil \sqrt[n]{m} \rceil)$ 
4    $W \leftarrow \emptyset$ 
5   for  $i \in \{1, 2, 3, \dots, e\}$  do  $W \leftarrow W \cup \{\text{Rand}_{\text{EdgeWth}}(V_C \setminus W)\}$ 
6   return  $\{\{v, u\} \mid u \in W\}$ 
7 end

```

Fonte: elaborado pelo autor.

Uma quantidade e de arestas é gerada utilizando a função $\text{Rand}_{\text{EdgeWth}}(W)$, definida por [Largeron et al. \(2015\)](#). Essa função escolhe um vértice u aleatório dentre o conjunto W , utilizando a densidade probabilística descrita na equação 3.4. A probabilidade de escolher um vértice é proporcional a seu grau dividido pela soma dos graus em W .

$$u \mapsto \frac{\deg(u)}{\sum_{u' \in W} \deg(u')} \quad (3.4)$$

As arestas que ligam o vértice v a outros com os quais ele não compartilha comunidades se dá conforme a função $\text{edgesBetween}(v, \mathcal{G}_p, C_c, m)$, como descrito no [Quadro 8](#). A função elenca um conjunto p de vértices que podem ser escolhidos. Esse conjunto é a união de todos os representantes de comunidades nas quais o vértice v não está sendo introduzido.

O máximo de arestas é definido como m sendo o mínimo entre a cardinalidade conjunto de adjacências possíveis, a quantidade de arestas internas à comunidades, e o parâmetro E_{btw}^{\max} . A quantidade de arestas a serem geradas e é definida com a função Rand_{PL} , de zero a m . As arestas em si são construídas com base na função $\text{Rand}_{\text{EdgeBtw}}(W)$ ([LARGERON et al., 2015](#)), que escolhe o representante com o qual o vértice se ligará usando a densidade probabilística descrita na equação 3.5. Ressalta-se que com a adaptação que foi realizada a função d é dependente da comunidade que contextualiza a semelhança entre os vértices.

Quadro 8 – Segunda fase do modelo – função $edgesBetween(v, \mathcal{G}_p, C_c, m)$

```

1 Function  $edgesBetween(v, \mathcal{G}_p, C_c, n)$ 
2    $p' \leftarrow \text{flat}(C_n) \setminus C_c$ 
3    $p \leftarrow \bigcup_{C \in p'} \{ \{ C, r \} \mid r \in R_C \}$ 
4    $m \leftarrow \min(E_{\text{btw}}^{\text{max}}, |p|, n)$ 
5    $e \leftarrow \text{Rand}_{\text{PL}}(0, 1, 2, \dots, m)$ 
6    $W \leftarrow \emptyset$ 
7   for  $i \in \{ 1, 2, 3, \dots, e \}$  do  $W \leftarrow W \cup \{ \text{Rand}_{\text{EdgeBtw}}(p \setminus W) \}$ 
8   return  $\{ \{ v, u_0 \} \mid u \in W \}$ 
9 end

```

Fonte: elaborado pelo autor.

$$u \mapsto \frac{d(v, u)^{-1}}{\sum_{u' \in W} d(v, u')^{-1}} \quad (3.5)$$

3.2.3.3 Atualização do estado

A parte final de cada iteração de lote de vértices, é a construção de um novo estado, isso é, a redefinição de \mathcal{G}_p para que este seja usado no processamento do lote que sucede o que se está terminando de processar. A expansão das arestas \mathcal{E} é apenas a união das arestas geradas no processamento de cada vértice. A reconstrução da cobertura C_n é trivial, abstraído para dentro da função $\text{buildCover}(C_n, T_C)$, a árvore de comunidades mantém a mesma topologia, mas nas comunidades folhas são adicionados os vértices conforme mapa em T_C .

A eleição dos novos representantes apresenta um ponto de interesse mais relevante, considerando o impacto que a escolha de quais vértices representam a comunidade pode ter. A alternativa utilizada neste trabalho é a seleção dos representantes mais próximos ao centro de gravidade da comunidade. A quantidade de representantes é o menor valor entre NbRep e $|V_C|$.

Destaca-se que esses representantes não precisam ser vértices do grafo, já que são utilizados apenas para a comparação de distância. Em implementações alternativas poderia ser utilizado o centro de gravidade. Alternativamente, poderia se utilizar pontos que maximizam a distância do centro de gravidade, para que os representantes sejam exemplos da periferia da comunidade.

3.2.4 Adição final de arestas

Com a conclusão da segunda fase do algoritmo, tem-se um grafo conexo com uma cobertura que engloba todos os vértices de forma que não hajam comunidades vazias. Para todos os efeitos, o grafo gerado até este ponto já tem a maioria das propriedades que o modelo se propõe a gerar. Esta etapa final de geração de arestas é executado para o reforço dessas características.

Quadro 9 – Terceira fase do modelo – adição final de arestas

```

1 Output:  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, C_n)$ 
2
3  $l \leftarrow \max(\{L_C \mid C \in \text{flat}(C_n)\})$ 
4 while  $|\mathcal{E}| < MTE \wedge \mathcal{G} \neq K_{|\mathcal{V}|}$  do
5 begin
6    $T \leftarrow \{\{v, v'\} \mid v, v', v'' \in \mathcal{V} \mid \{v, v''\} \in \mathcal{E} \wedge \{v', v''\} \in \mathcal{E} \wedge \{v, v'\} \notin \mathcal{E}\}$ 
7    $T' \leftarrow \{e \in T \mid \exists C \in \text{flat}(C_n)(\forall v \in e(v \in C) \wedge L_C = l)\}$ 
8   if  $T' = \emptyset$  then  $l \leftarrow l - 1$ 
9   else  $\mathcal{E} \leftarrow \mathcal{E} \cup \{\text{Rand}_{\text{Uni}}(T')\}$ 
10 end

```

Fonte: elaborado pelo autor.

O processo descrito no [Quadro 9](#) inicia identificando o nível máximo que uma comunidade pode possuir, l . Com esse valor, é iniciado um processo iterativo, enquanto forem encontradas triplas conexas dentro de comunidades com este nível, ele se mantém. Quando todas as comunidades desse nível forem subgrafos completos o valor de l é decrementado.

O laço em si itera enquanto a quantidade de arestas no grafo não for igual ao parâmetro MTE e o grafo não for um grafo completo. Isso é, se o parâmetro denotar uma quantidade de arestas superior ao que é possível com a quantidade de vértices, este não será um laço infinito.

O processo interno ao loop é a identificação das arestas que se adicionadas ao grafo completariam mais um triângulo. Depois, essas arestas são filtradas para considerar apenas as que seriam internas a alguma comunidade de nível l . Se T' é vazio, isso indica que todas as comunidade de nível l ou superiores são grafos completos (ou seriam grafos desconexos, o que é trivialmente demonstrável como impossível neste ponto do algoritmo). Neste caso, l é decrementado para que se considerem as comunidades hierarquicamente superiores a estas. Caso T' seja não vazio, é escolhido uma aresta aleatória para ser preenchida no grafo.

Essa implementação deliberadamente otimiza o coeficiente de clusterização, definido na equação 2.3. Assumindo que as propriedades de homofilia estejam presentes. Esse processo deveria de ter pouco ou nenhum impacto na mesma, pela característica transitiva da semelhança como distância euclidiana. Isso é, qualquer aresta adicionada nesse processo que ligue dois vértices a e b que compartilham um vizinho c não vai ter uma distância maior que soma das duas arestas já presentes no grafo.

$$0 \leq d(a, b) \leq d(a, c) + d(b, c) \quad (3.6)$$

Da mesma forma, o impacto desse processo na distribuição de graus intuitivamente parece ser mínimo. A proporção de triplas conexas a qual um dado vértice v é diretamente proporcional ao grau de v . No entanto é possível demonstrar que valores mais elevados de MTE tem o efeito de diminuir a quantidade de vértices com grau um.

Da perspectiva das características estruturais de uma comunidade, é intuitiva também a compreensão de que valores razoáveis do parâmetro reforçarão a estrutura, construindo novas arestas nas comunidades folha. Mas também é perceptível que a geração de arestas a ponto de transformar comunidades que não são folha em cliques faz com que a estrutura das comunidades folha contidas nesta sejam destruídas. Mas, essas influências da parametrização são melhor exploradas com resultados experimentais.

4 RESULTADOS EXPERIMENTAIS

Neste capítulo serão descritos os resultados experimentais obtidos com a execução do modelo. Os valores apresentados em quadros e tabelas estarão arredondados para uma precisão de cinco casas decimais, e os valores apresentados textualmente estarão arredondados para uma casa decimal apenas.

A implementação do algoritmo descrito foi feita na linguagem python 3.8. A partir disso, foram executados os testes para avaliar a presença das propriedades desejadas no modelo. Para os testes iniciais, foram utilizados valores descritos no [Quadro 10](#).

Quadro 10 – Parâmetros básicos

N	E_{wth}^{\max}	E_{btw}^{\max}	MTE	\mathcal{A}	K	θ	NbRep
1000	45	3	7000	(1, 1)	(6, 2)	1/10	20

Fonte: elaborado pelo autor.

4.1 PRESENÇA DE COMUNIDADES

A presença de comunidades como uma propriedade de redes complexas está tipicamente atrelada a ideia de haverem mais arestas internas a uma comunidade do que arestas que atravessem a fronteira entre duas comunidades. Esse entendimento é solidificado na função de modularidade Q de [Girvan e Newman \(2002\)](#), conforme descrito na equação 2.1. [Shen et al. \(2009\)](#) descrevem uma adaptação, modularidade estendida EQ , para a aplicação em coberturas (em oposição a partições), conforme equação 2.2. Essa aplicação em coberturas destaca a possibilidade de comunidades sobrepostas. A aplicação de EQ recursivamente, de forma a considerar o subgrafo com os membros da comunidade dividido entre as comunidades que a compõe descreve a propriedade de comunidades hierárquicas.

$$O_i = \left[\sum_{c \in C_n | L_c = i-1} V_c \right] - \left[\sum_{c \in C_n | L_c = i} V_c \right] \quad (4.1)$$

A presença de sobreposição nas comunidades pode ser medida como um função O_i que calcula a quantidade de vértices presentes em mais de uma comunidade de nível i . O valor é dado pela diferença da quantidade de vértices em comunidades de nível i e a quantidade de vértices em comunidades de nível $i + 1$. O efeito disso é que vértices que estejam em duas comunidades distintas de nível $i + 1$ ou menores, serão contados duas vezes, mas os que pertencem a duas comunidades de nível i ou menores serão compensadas. Por fim, a soma de O_i para todos os níveis será anotada como O .

Observa-se na [Tabela 1](#) os valores de EQ para 10 execuções do modelo com os parâmetros descritos no [Quadro 10](#), B_0 a B_9 . Para as comunidades de primeiro nível, isso é, as que são compostas por sub-comunidades, a função gera uma média de aproximadamente 0,7, que indica uma estrutura de comunidade muito expressiva. No segundo nível, o valor médio foi de aproximadamente 0,8 indicando comunidades ainda mais expressivas. Nos dois casos, o desvio padrão é bastante baixo, indicando a consistência com a qual a propriedade se manifestou nos grafos gerados.

Tabela 1 – Modularidade com os parâmetros básicos

Execução	EQ_1	EQ_2	O_1	O_2
B_0	0,71695	0,79204	16	235
B_1	0,72441	0,79760	18	211
B_2	0,70039	0,79391	14	252
B_3	0,73633	0,79622	13	190
B_4	0,71758	0,78549	26	235
B_5	0,73179	0,79201	33	224
B_6	0,74779	0,79828	15	213
B_7	0,73912	0,80411	20	200
B_8	0,74237	0,80504	19	194
B_9	0,71339	0,79877	17	237
Média	0,72701	0,79635	19	219
Desvio	0,01247	0,00441	4,3	17,5

Fonte: elaborado pelo autor.

A [Tabela 1](#) indica também os valores de sobreposição entre comunidades de primeiro (O_1) e segundo nível (O_2). Os valores indicam quem em média, dos mil vértices presentes em um grafo como o observado, dezessete estarão em duas comunidades de primeiro nível e em média 219 vértices estarão em duas comunidades de segundo nível. O que esses valores indicam é que até vinte e quatro por cento dos vértices, para essa parametrização, estariam distribuídos em mais de uma comunidade.

Naturalmente, a escolha de valores distintos para os parâmetros pode ser feita para reforçar as características desejáveis. Limitando o valor de MTE e considerando uma estrutura de comunidades gerada com $K = (9, 2, 2, 2)$, obteve-se os dados distritos na [Tabela 2](#). Observa-se que a redução do número mínimo foi feita para evitar a descaracterização das comunidades folha, considerando que o valor de K gera 72 dessas.

O incremento nos valores obtidos é atribuível à existência de mais comunidades

Tabela 2 – Modularidade com $K = (9, 2, 2, 2)$

Execução	EQ_1	EQ_2	EQ_3	EQ_4	O
M_0	0,96702	0,95833	0,93128	0,87673	0
M_1	0,96721	0,95979	0,93388	0,88221	0
M_2	0,96514	0,95933	0,93357	0,87722	0
M_3	0,96541	0,95711	0,92974	0,87888	0
M_4	0,96672	0,95868	0,93582	0,88090	0
Média	0,96630	0,95865	0,93286	0,87919	0
Desvio	0,00082	0,00074	0,00188	0,00189	0

Fonte: elaborado pelo autor.

para as quais distribuir os vértices, não apenas em profundidade mas também em diâmetro. Descartando-se a possibilidade de ser a alteração do valor de MTE ao realizar testes com este zerado mas mantendo o K padrão. Em compensação, essa parametrização faz com que a escolha de comunidades folha se torne muito mais provável, anulando a possibilidade de comunidades sobrepostas. Esse comportamento ocorre pois no processo de escolha das comunidades a ordenação das comunidades ocorre preferindo comunidades que têm os representantes mais próximos, e as comunidades folha por serem em maior quantidade são muito mais prováveis de serem escolhidas.

Nos testes descritos na [Tabela 3](#), onde com $MTE = 0$ realizou-se uma exploração dos valores de EQ para diversos valores em K . Note-se o caso M_9 como contra exemplo do impacto de K na modularidade, bem como o exemplo M_6 , que nas comunidades de primeiro tem $EQ = 0,5$.

Tabela 3 – Modularidade com variação de K

Execução	K	Média de EQ
M_5	(5, 5, 5)	0,77889
M_6	(2, 4, 8)	0,77721
M_7	(8, 2, 4)	0,92434
M_8	(100)	0,95152
M_9	(3)	0,49024

Fonte: elaborado pelo autor.

Alternativamente, é possível modificar significativamente os valores obtidos para a modularidade com os parâmetros E_{btw}^{\max} e E_{wth}^{\max} . Na execução M_{10} utilizou-se os demais parâmetros nos valores padrão, mas com $E_{btw}^{\max} = 45$ e $K = (2, 2)$, para um EQ médio de

0,46.

4.2 HOMOFILIA E HOMOGENEIDADE

Homogeneidade é a principal propriedade das comunidades definidas por semelhança de vértices, ela está intimamente ligada à propriedade de homofilia. Assumindo a distância euclidiana como função de semelhança, a forma natural de medir a homogeneidade das comunidades descritas por uma cobertura é a inércia e a razão da inércia entre diferentes níveis de comunidades.

4.2.1 Inércia

Inércia, definida como a soma dos quadrados das distâncias de cada membro de um conjunto com a média do conjunto, sumariza quão semelhantes ou dissemelhantes os membros são. A razão entre a inércia em diferentes níveis é facilmente extrapolada do conceito de razão de inércia introduzido por [Largeron et al. \(2015\)](#). Assumindo que as comunidades de uma rede complexa sejam homogêneas e hierárquicas, é natural que as comunidades mais específicas (sub-comunidades), tenham menos diversidade que as comunidade mais generalistas. Tendo em vista que a inércia calculada com uma cobertura deve levar em conta que vértices serão contados múltiplas vezes se estiverem presentes em múltiplas comunidades, a inércia para um nível l segue descrita na equação 4.2. Considerando que C_l é o conjunto de comunidades de nível l .

$$I_l = \sum_{c \in C_l} \frac{\sum_{v \in c} |v - g_c|^2}{|c|} \quad (4.2)$$

Onde C_l é o conjunto de comunidades de nível l e g_c é o centro de gravidade da comunidade c . I_l apresenta uma forma de comparação entre diferentes coberturas, de forma a identificar qual seria mais homogênea dado um mesmo conjunto do vértices. A extensão do conceito de inércia para a razão entre a inércia aplicada a dois níveis distintos se dá para isolar o fator de escala. Isso é, I_l/I_0 descreve não apenas quão homogêneo são os agrupamentos estabelecidos, mas o quão homogêneos eles são em relação à população em geral. Note-se que I_0 nesse contexto se refere à comunidade C_n , isso é, a comunidade que contém todos os vértices.

4.2.2 Distância esperada

A homofilia, como propriedade de redes complexas onde os vértices tendem a estarem adjacentes à vértices semelhantes. Em um sistema com atributos discretos, é facilmente estabelecido uma função Δ que descreve o quão semelhantes os vértices adjacentes são do que seria esperado em um modelo nulo. O modelo nulo trivial neste caso é a consideração de vértices conectados com igual probabilidade, mantendo-se a mesma quantidade de vértices e arestas, conforme descrito por [Easley e Kleinberg \(2010\)](#).

A adaptação para um modelo de um espaço contínuo de múltiplas dimensões se dá calculando uma aproximação da distância média entre cada par de vértices, essa média, H_e , é o valor que seria esperado se o grafo não fosse homofílico. O valor real é calculado como a média da distância dos pontos adjacentes, H_r .

4.2.3 Resultados

Na [Tabela 4](#) estão discriminados os valores médios de inércia para as comunidades organizadas por nível. A leitura que se pode fazer desses dados, em especial das médias, é: em média uma comunidade de primeiro nível tem 36,6% da diversidade que se encontra na população geral; e em média uma comunidade de segundo nível (sub-comunidade) tem 31,5% da diversidade presente na população em geral. Por fim, em média a distância entre dos vértices adjacentes é metade do que seria esperado sem a homofilia, indicando a presença dessa propriedade.

Naturalmente, existe uma influência significativa que a quantidade de comunidades tem neste valor. Observa-se na [Tabela 5](#), com mais comunidades para distribuir os vértices, as comunidades ficam significativamente mais homogêneas. Como efeito disso, a distância entre vértices conectados é bastante reduzida.

Também de forma natural, existe uma relação entre o parâmetro θ e a inércia. É apontado na [Tabela 6](#) a variação da inércia conforme se altera θ . Duas características dos dados descritos são relevantes, primeiramente com $\theta = 1$ se as comunidades não estivessem sendo construídas ortogonalmente, seriam esperadas comunidades onde a diversidade interna se aproximaria da diversidade da população geral, mas não é isso o que se observa. Isso é, em um modelo nulo, onde não existe correlação entre os atributos do vértice e a comunidade em que ele participa, seria esperado uma razão de inércia que tendesse a um.

O segundo ponto a ser destacado é que θ não é um controle direto sobre a inércia, nos casos com $\theta = 0,66$, $\theta = 0,33$ e $\theta = 0,1$ ([Tabela 4](#)), a razão se manteve em um

Tabela 4 – Homogeneidade e homofilia com os parâmetros básicos

Execução	I_1/I_0	I_2/I_0	$I_1 + I_2/2I_0$	H_r/H_e
B_0	0,38193	0,34155	0,36174	0,45942
B_1	0,34067	0,30168	0,32118	0,45920
B_2	0,34262	0,29697	0,31980	0,49265
B_3	0,35961	0,29986	0,32974	0,48305
B_4	0,41670	0,37592	0,39631	0,52725
B_5	0,35321	0,28246	0,31784	0,46047
B_6	0,37795	0,29310	0,33553	0,46127
B_7	0,35209	0,29903	0,32556	0,45348
B_8	0,36251	0,32820	0,34536	0,46364
B_9	0,36874	0,32936	0,34905	0,50499
Média	0,36560	0,31481	0,34021	0,47654
Desvio	0,01658	0,02316	0,01832	0,02035

Fonte: elaborado pelo autor.

Tabela 5 – Homofilia e Homogeneidade com $K = (9, 2, 2, 2)$

Execução	I_1/I_0	I_2/I_0	I_3/I_0	I_4/I_0	H_r/H_e
M_0	0,18740	0,11020	0,05650	0,03240	0,15975
M_1	0,18766	0,11563	0,06017	0,03277	0,16491
M_2	0,18446	0,11812	0,05993	0,03526	0,17281
M_3	0,18566	0,11307	0,06003	0,03169	0,16211
M_4	0,18111	0,11163	0,05563	0,03016	0,16158
Média	0,18526	0,11373	0,05845	0,03246	0,16423
Desvio	0,00198	0,00252	0,00191	0,00125	0,00370

Fonte: elaborado pelo autor.

Tabela 6 – Homofilia e Homogeneidade com θ variável

Execução	θ	I_1/I_0	I_2/I_0	Média
I_0	1,00	0,78255	0,73782	0,71637
I_1		0,70558	0,67610	
I_2		0,70773	0,68841	
I_3	0,66	0,40448	0,34173	0,36539
I_4		0,40280	0,33749	
I_5		0,37041	0,33541	
I_6	0,33	0,34582	0,29460	0,34005
I_7		0,36632	0,30172	
I_8		0,40126	0,33059	

Fonte: elaborado pelo autor.

mesmo patamar, entre 34% e 40%. Existem outros fatores que afetam diretamente a homogeneidade das comunidades, e que são mais dificilmente controlados por parâmetros.

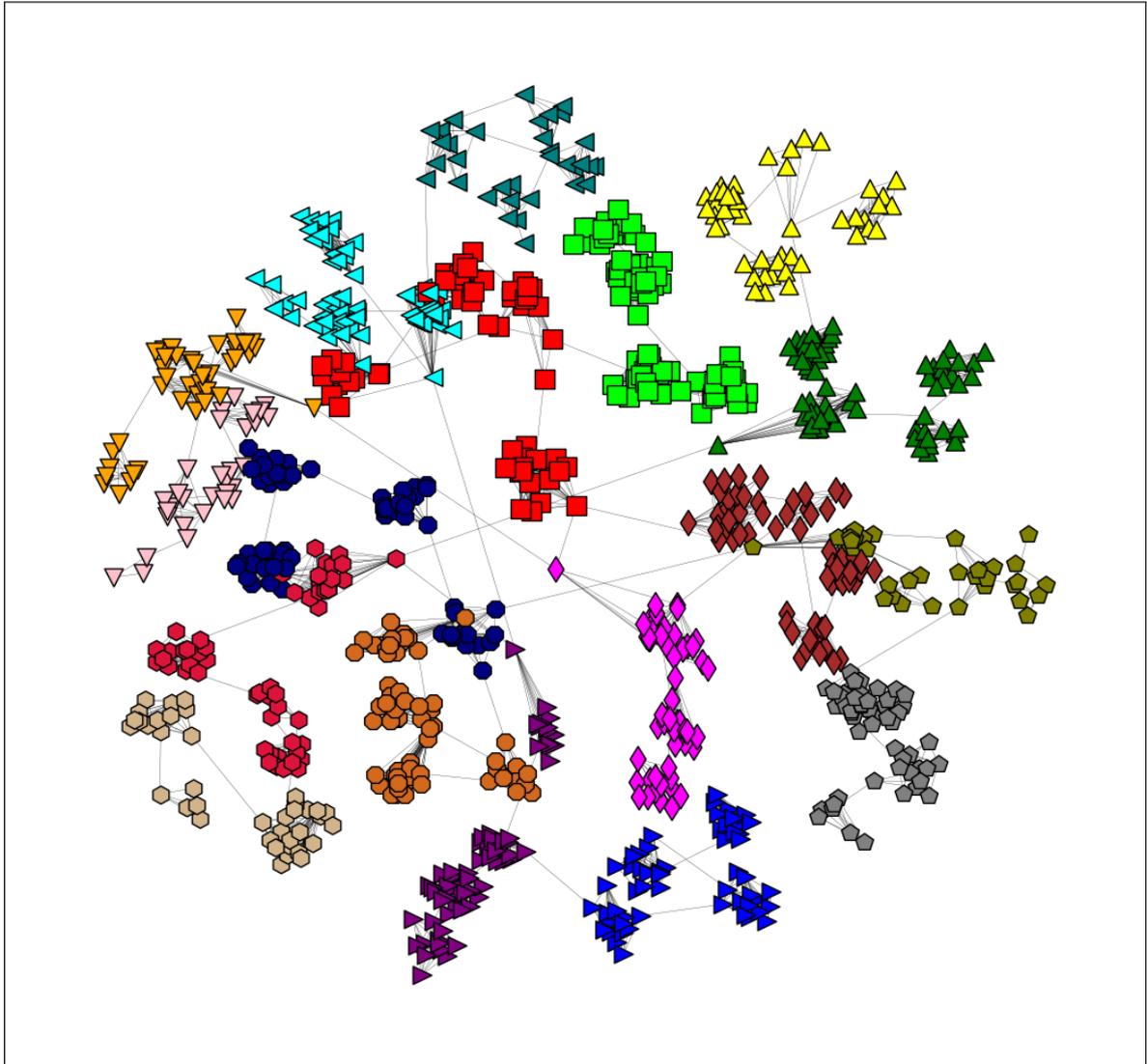
4.3 VISUALIZAÇÃO

Uma das características da propriedade de existência de comunidades é que a definição de comunidade é dependente de contexto. Ainda assim, parece haver um entendimento natural de comunidades e de suas propriedades a partir da visualização das mesmas. Segue a baixo a renderização dos grafos B_0 e M_0 , gerados respectivamente com os parâmetros padrão e com os parâmetros otimizando para a modularidade (mais valores em K e valores maiores).

A renderização desses grafos conta com uma codificação de forma que vértices que estejam na mesma comunidade de primeiro nível compartilhem o formato, e vértices que estejam na mesma comunidade de segundo nível compartilhem as cores, como é observável na [Figura 4](#). Vértices que estejam em mais de uma comunidade de segundo nível terão a cor branca, e vértices que estejam em mais de uma comunidade de primeiro nível terão o formato de círculo. Dessa forma, é possível identificar as áreas de sobreposição, bem como os primeiros níveis da hierarquia de comunidades.

Na [Figura 4](#) está a renderização do grafo M_0 com o posicionamento dos vértices de forma a minimizar a distância deles na imagem. Além das divisões entre comunidades que estão categorizadas com os formatos e com as cores, é visível uma divisão interna de pequenos grupos densamente conexos. Essa visualização e entendimento intuitivo da

Figura 4 – M_0 renderizado aproximando os vértices adjacentes

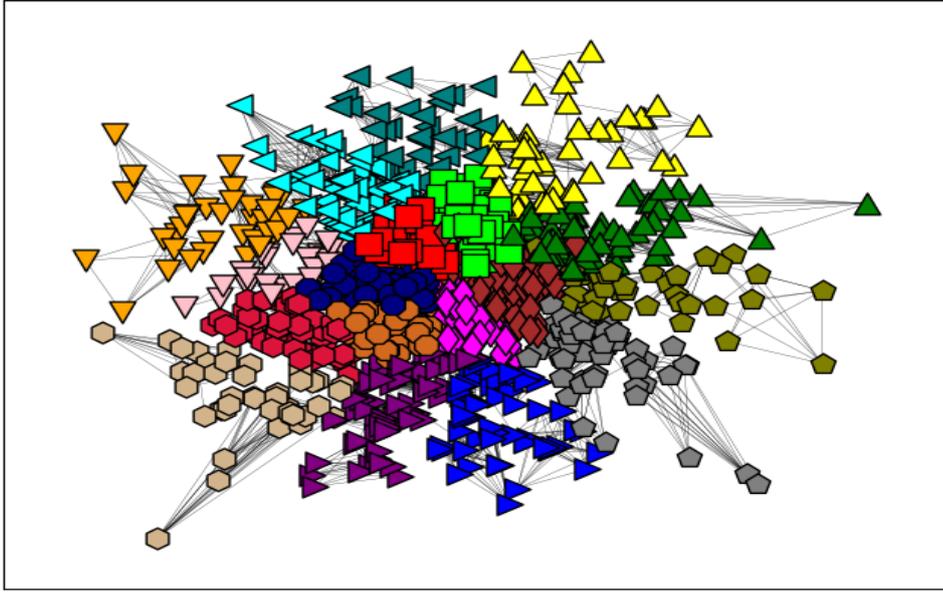


Fonte: elaborado pelo autor.

divisão é decorrente de uma presença muito acentuada da propriedade de comunidades.

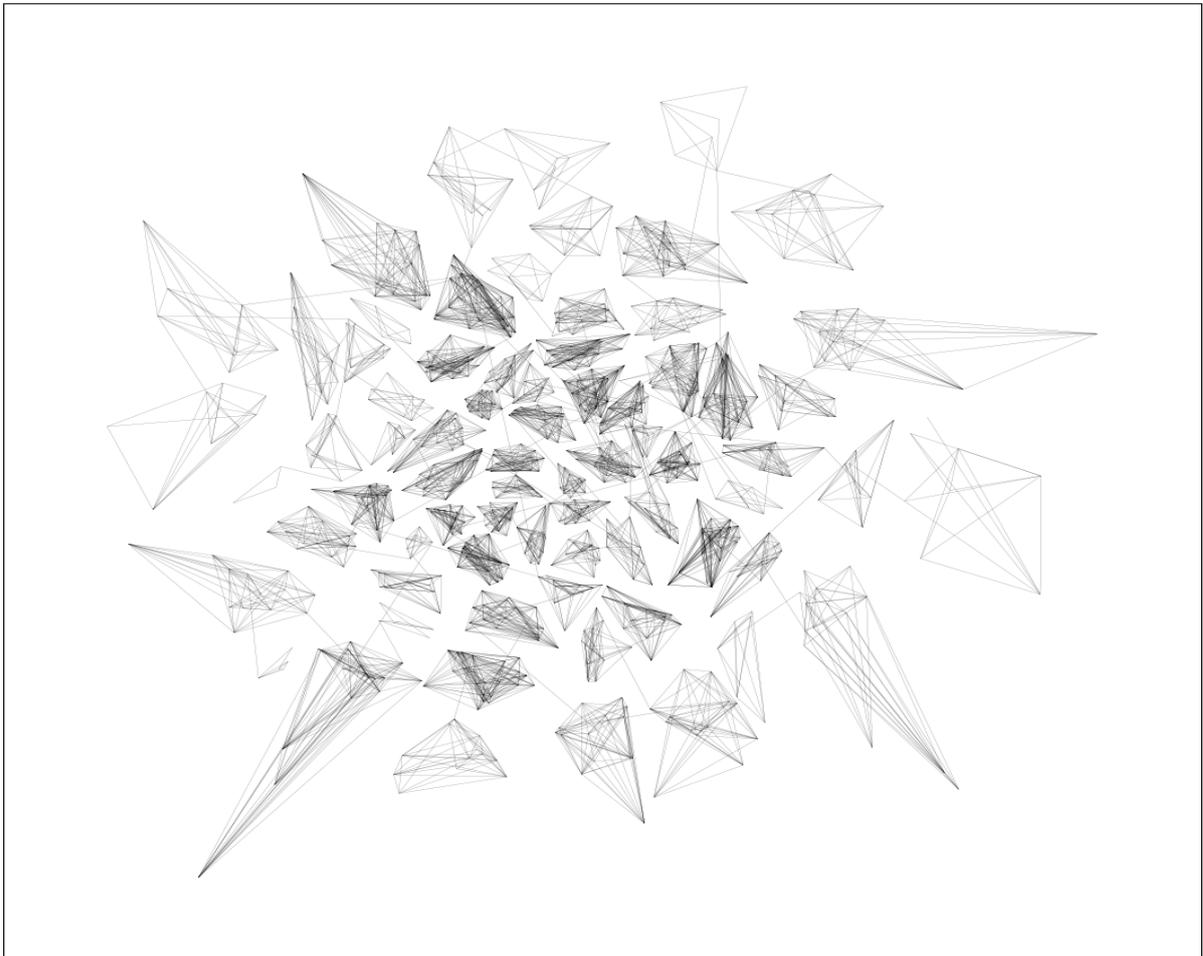
Essa divisão mais apresenta, conforme já foi distrito anteriormente, é responsável também pela falta de vértices em regiões sobrepostas, de certa forma sacrificando ganhos nessa propriedade (sobreposição) para ganhar maior definição em outra (comunidades hierárquicas). A [Figura 5](#) e a [Figura 6](#) demonstram isso. Ambas apresentam os vértices posicionados com base nos seus atributos, de forma que o centro (mais próximo a origem) e mais densamente povoado, no entanto a [Figura 6](#) não renderiza os vértices, apenas as arestas. Com isso, é abundantemente claro o quão fortemente delineadas são as comunidades. Demonstra-se que cada comunidade possui um domínio absoluto sobre alguma região do espaço onde os vértices se encontram.

Figura 5 – M_0 renderizado com os vértices posicionado pelos seus atributos



Fonte: elaborado pelo autor.

Figura 6 – M_0 renderizado com os vértices posicionado pelos seus atributos (apenas arestas)



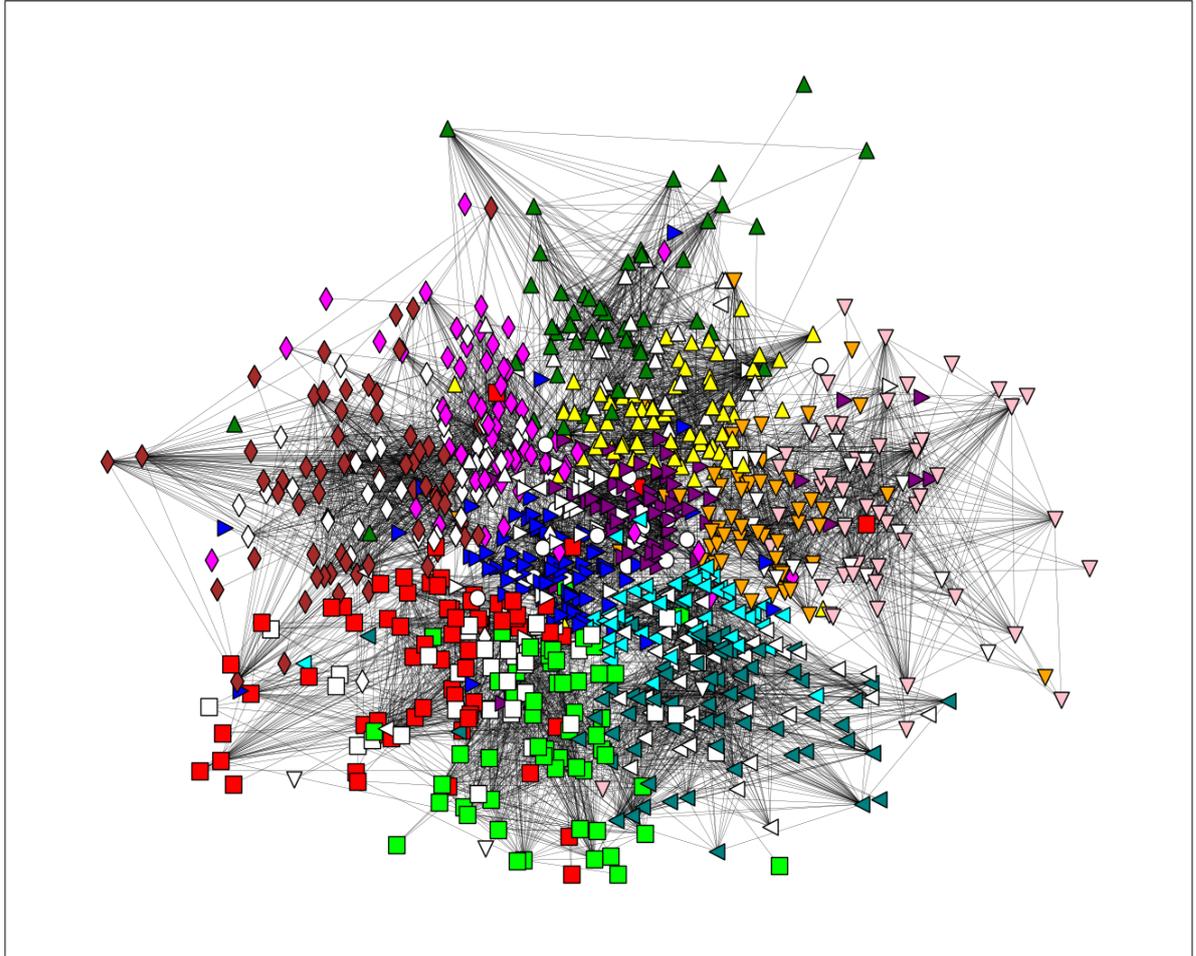
Fonte: elaborado pelo autor.

A [Figura 7](#) e a [Figura 8](#) fazem a mesma distinção, mas com o grafo B_0 . Este grafo apresenta um conjunto de comunidades onde ocorre a sobreposição, não apenas no sentido de comunidades sobrepostas, mas em um sentido mais simples, de que há vértices que não pertencem a comunidade (e que não tem ligações com ela) ocupando o mesmo espaço. Novamente, esse é um efeito de uma menor proporção de comunidades.

A visualização das arestas mostra uma divisão visual das comunidades, com uma comunidade no centro e cinco comunidades a rodeando. Dessas seis comunidades, é possível identificar que existem grupos distintos dentro delas, isso é, as sub-comunidades são visíveis, mas pela natureza delas (e presença de vértices que pertencem as duas), são divisões muito menos precisas.

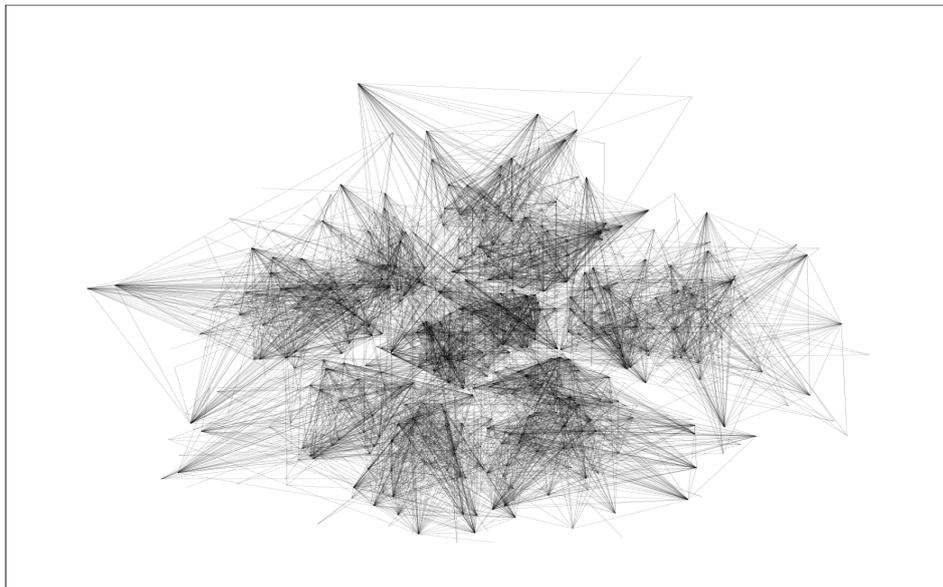
O mesmo grafo está exibido novamente na [Figura 9](#) e na [Figura 10](#). Nelas é possível identificar mais claramente as áreas de sobreposição das comunidades de segundo nível (identificadas por não terem cor) bem como a presença de mais arestas internas as comunidades, naturalmente.

Figura 7 – B_0 renderizado com os vértices posicionado pelos seus atributos



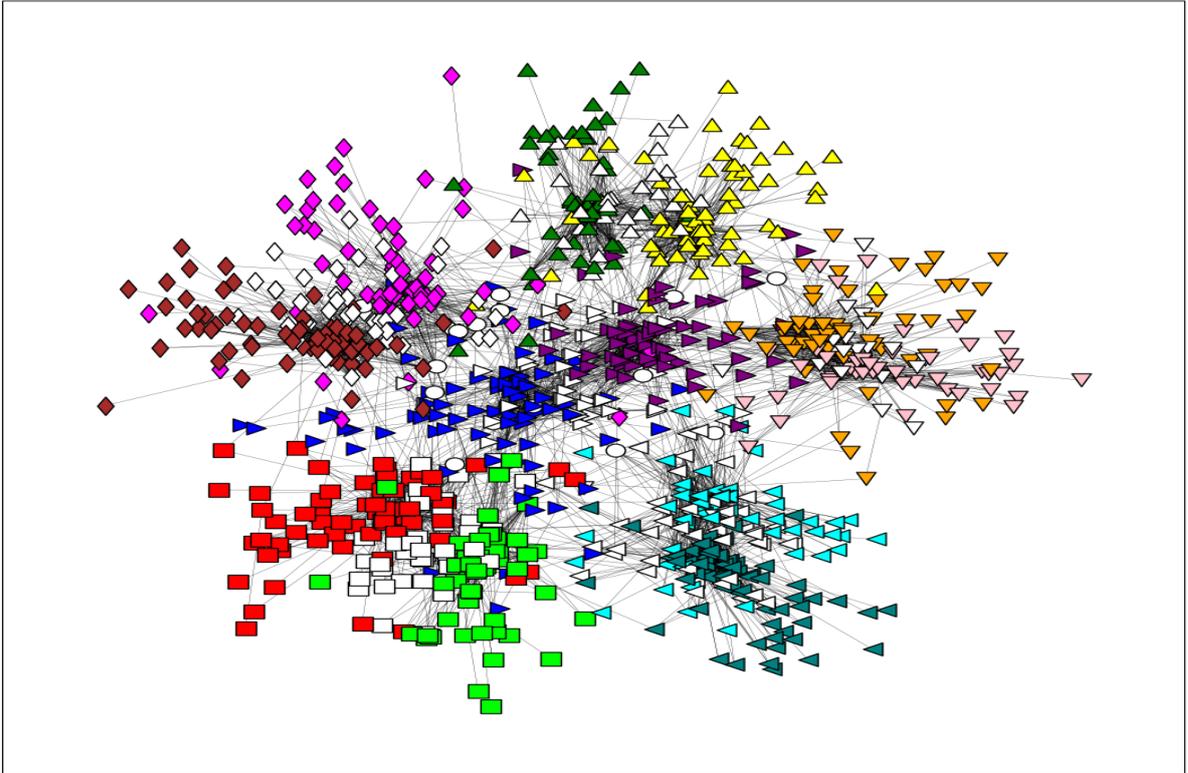
Fonte: elaborado pelo autor.

Figura 8 – B_0 renderizado com os vértices posicionado pelos seus atributos (apenas arestas)



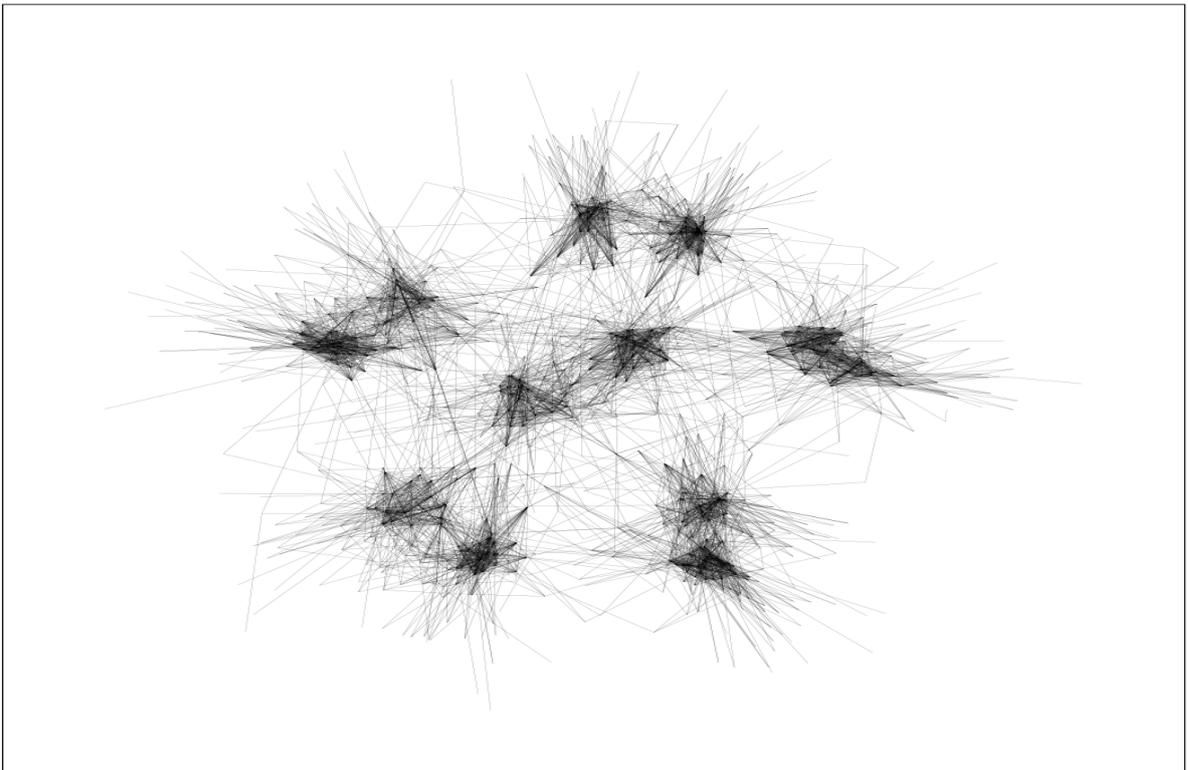
Fonte: elaborado pelo autor.

Figura 9 – B_0 renderizado aproximando os vértices adjacentes



Fonte: elaborado pelo autor.

Figura 10 – B_0 renderizado aproximando os vértices adjacentes (apenas arestas)



Fonte: elaborado pelo autor.

5 CONCLUSÕES

Este trabalho demonstra um modelo algorítmico para a geração de redes complexas com comunidades hierárquicas e comunidades sobrepostas. Junto do algoritmo foram identificados valores básicos para a parametrização que geram grafos com todas as propriedades desejadas. A fundamentação teórica apresenta uma exploração em língua portuguesa academicamente relevante, descrevendo redes complexas, comunidades e as suas propriedades. Os resultados experimentais demonstram o sucesso nos objetivos gerais e específicos estipulados para o modelo.

Quando ao objetivo específico da inclusão de comunidades na rede gerada, bem como a construção de uma *ground truth* para quais são os membros das comunidades é demonstrado como bem sucedido na exploração dos resultados da função de modularidade estendida EQ . A presença das comunidades, bem como o pertencimento dos vértices a elas é confirmado pela função de qualidade de cobertura. Notadamente, essa função e a sua aplicação recursiva destacam a presença de comunidades hierarquicamente organizadas, de forma que se aninham em comunidades auto semelhantes, e sobrepostas, de forma a compartilharem vértices. Demonstradamente, o compartilhamento de vértices é sensível à parametrização do modelo, se mostrando mais presente em grafos gerados a partir de parâmetros que geram uma estrutura com menos comunidades.

O objetivos relativos à comunidades homogêneas foram bem sucedidos conforme demonstrado experimentalmente, sendo parametrizável o quão pronunciada é essa propriedade. Com a escolha de parâmetros que reforcem comunidades de tamanho reduzido (maiores valores para K) os grafos gerados consistentemente apresentam comunidades com menos de quatro por cento da diversidade global. Os parâmetros básicos, consideravelmente conservadores no quanto eles reforçam características específicas, geram grafos onde as comunidades apresentam consistentemente menos de trinta e quatro por cento da diversidade que o grafo.

Por fim, acredita-se que esse trabalho poderá se tornar uma referência relevante, dada a falta de literatura em língua portuguesa para a área de redes complexas e comunidades.

5.1 EXTENSÕES

Redes complexas como área de estudo é um campo promissor e mais estudos devem ser realizados. O processo de geração de redes complexas com comunidades é uma área de relevância acadêmica, e o modelo apresentado neste trabalho tem um conjunto de

extensões desejáveis, são elas:

A otimização da cobertura gerada para algoritmos de percolação de cliques, por meio da redução de arestas entre vértices que estejam entre duas comunidades. Isso poderia ser implementado no processo de geração final de arestas, removendo estes vértices e não os considerando na avaliação de que a comunidade é um grafo completo. Idealmente, isso deveria ser parametrizado com um valor entre zero e um, identificando a proporção de arestas entre vértices na mesma área de superposição.

A elaboração de um método mais deliberado nas escolhas de comunidades, que permita um controle mais fino da proporção de vértices em áreas de superposição. Uma implementação intuitiva seria a parametrização de uma proporção, quantos vértices estariam em uma comunidade folha apenas, quantos estariam em duas, quantos estariam em três (O modelo não é incompatível com a escolha de três ou mais comunidades) etc. Cada vértice teria primeiramente uma quantidade de comunidades folha decidido com essas proporções, e a escolha poderia ser feita ou repetindo a chamada para a função, ou escolhendo as comunidades mais próximas.

O desenvolvimento de novos modelos de eleição de representantes, considerando que naturalmente a escolha de representantes com características distintas pode otimizar o processo. As escolhas de metodologia naturais seriam a minimização da função d para os vértices já dentro da comunidade, tanto considerando a média dos vértices para com os representantes eleitos, quanto para o seu mínimo.

O desenvolvimento de novos modelos para a função d , especialmente em vista da ortogonalidade. Uma abordagem que se desejaria explorar é a limitação da quantidade de níveis hierárquicos para a quantidade de dimensões nos vértices, para que sejam usados eixos distintos como eixo de ortogonalidade por nível. Isso é, as comunidades de primeiro nível teriam predileção por discriminar seus membros com base em um eixo, as de segundo nível teriam por outro eixo (ortogonal ao primeiro) e assim sucessivamente. De fato, a implementação de uma estrutura assim poderia suplantiar a necessidade de um parâmetro de controle θ , pois a variação em \mathcal{A} seria capaz de modificar esse comportamento controlando inclusive para diferentes níveis.

Um estudo mais aprofundado das proporcionalidades identificadas em sistemas do mundo real e a sua presença nos grafos gerados pelo modelo. Um exemplo disso é a distribuição de graus em função logarítmica, que apesar de ter sido observado em testes, não pode ser avaliada mais rigorosamente.

Uma avaliação comparativa da cobertura gerada pelo modelo com as coberturas

identificadas por algoritmos de identificação de comunidades canonicamente aceitos na literatura.

REFERÊNCIAS

- AKOGLU, L.; FALOUTSOS, C. Rtg: A recursive realistic graph generator using random typing. In: BUNTINE, W. et al. (Ed.). **Machine Learning and Knowledge Discovery in Databases**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. p. 13–28. ISBN 978-3-642-04180-8.
- EASLEY, D.; KLEINBERG, J. **Networks, crowds, and markets: Reasoning about a highly connected world**. [S.l.]: Cambridge university press, 2010.
- FORTUNATO, S. Community detection in graphs. **Physics reports**, Elsevier, v. 486, n. 3-5, p. 75–174, 2010.
- GIRVAN, M.; NEWMAN, M. E. Community structure in social and biological networks. **Proceedings of the national academy of sciences**, National Acad Sciences, v. 99, n. 12, p. 7821–7826, 2002.
- LARGERON, C. et al. Generating attributed networks with communities. **PloS one**, Public Library of Science, v. 10, n. 4, p. e0122777, 2015.
- METZ, J. et al. **Redes complexas: conceitos e aplicações**. São Carlos, SP, Brasil., 2007.
- SHEN, H. et al. Detect overlapping and hierarchical community structure in networks. **Physica A: Statistical Mechanics and its Applications**, Elsevier, v. 388, n. 8, p. 1706–1712, 2009.
- SLOTA, G. M. et al. Scalable generation of graphs for benchmarking HPC community-detection algorithms. In: **Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC 2019, Denver, Colorado, USA, November 17-19, 2019**. [S.l.]: ACM, 2019. p. 73:1–73:14.
- STEGEHUIS, C.; HOFSTAD, R. V. D.; LEEUWAARDEN, J. S. V. Epidemic spreading on complex networks with community structures. **Scientific reports**, Nature Publishing Group, v. 6, n. 1, p. 1–7, 2016.