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 Ciências da Computação no Centro de de Ciências Exatas e Naturais da Universidade Regional de Blumenau como requisito parcial para a obtenção de grau de Bacharel em Ciências da Computação.

Professor Aurelio Faustino Hoppe, Mestre - Orientador

BLUMENAU
2022

FOLHA DE ASSINATURAS

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

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

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

ABSTRACT

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

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

LISTA DE FIGURAS

LISTA DE QUADROS

SUMÁRIO

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 está contida não 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 e Leeuwaarden (2016).

Como apontado por 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 um sub grafo completo, denominado clique (FORTUNATO, 2010), pode ser interpretada como uma propriedade relevante a indicação de que desse conjunto de indivíduos onde todos conhecem todos.

Girvan e Newman (2002) também aponta 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. O trabalho de Fortunato (2010) indica isso na discussão de múltiplas 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) descreve o que é chamado na literatura de um modelo de geração algorítmica de redes complexas onde 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) descreve um modelo mais primitivo, que não realiza a atribuição explicita de comunidades, mas que gera um grafo com essas comunidades ainda assim. Indica-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, como comunidades hierárquicas e comunidades sobrepostas, parecem estar pouco presentes em modelos de geradores de redes complexas.

Tendo esse contexto, este trabalho pretende adaptar 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 a construção de 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 possibilidade de comunidades hierárquicas e de comunidades sobrepostas;

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.

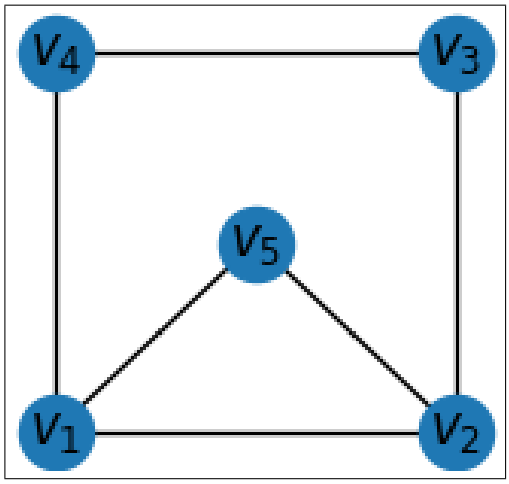
O terceiro capítulo discute o desenvolvimento do modelo algorítmico proposto, incluindo ferramentas e técnicas utilizadas. Também são apresentados os blocos de pseudo código do modelo.

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

Grafos podem trivialmente 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 ajacentes 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 formato pelos vértices v_1, v_2 e v_5 é também um grafo completo. A esse conjunto de vértices que forma um subgrafo completo dá-se o nome de clique (FORTUNATO, 2010).

2.1 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 sub grafo conexo, por exemplo (FORTUNATO, 2010).

Largeron et al. (2015) define comunidade como uma classe de estrutura topológica comum a redes complexas, essas comunidades são categorizadas por terem uma densidade de vértices elevada. Shen et al. (2009) entende que comunidades sejam estruturas que contenham múltiplos cliques dentro de si, e que essas comunidades se dispões em uma

estrutura recursiva. Akoglu e Faloutsos (2009) descreve comunidades como estruturas modulares, onde nodos de um vértice 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) define "Cluster" e comunidade como duas propriedades distintas, o primeiro sendo a probabilidade de dois nodos ambos adjacentes a um terceiro serem também ajacentes 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 á 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 do objeto em questão e que partindo disso se atribui pertencimento á comunidades, ou ainda por comparação com um exemplo nulo 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 tendencia 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 no 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) representa os vértices como sequencias de caracteres de tamanhos variáveis em que a probabilidade de dois vértices estarem ligados é maior conforme mais caracteres eles compartilham; e Largeron et al. (2015) representa os vértices como pontos em um espaço n-dimensional e define 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 á 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 Fortunato (2010) função de qualidade mais comumente utilizada é a modularidade Q de Girvan e Newman (2002).

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

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).

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

Tanto no caso da fórmula da equação 2.1 quanto no da 2.2 a função definida é uma somatórias em que alguns termos se repetem. Primeiramente, é preciso descrever que a função $\delta(C_i C_j)$ retorna um se C_i for igual a C_j, e zero noutro caso (FORTUNATO, 2010). Considerando isso, no caso de uma partição (sem comunidades hierárquicas, e sem comunidades sobrepostas), as duas somatórias iteram sobre os mesmos valores, a primeira com os vértices i e j e a segunda com os vértices v e w.

Essa iteração olha para todos os pares de vértices que compartilham alguma comunidade, e soma o valor de A_{ij}, sendo A a tabela de adjacência do grafo em questão. Então é subtraído um valor ${}^{K_i K_j}/_{2m}$, onde K_i é o grau do vértice i e m é a quantidade de arestas no grafo ($2m$ portanto é a soma dos graus de todos os vértices). Esse valor é o a

probabilidade de uma aresta entre os vértices i e j no modelo nulo de Girvan e Newman (2002), considerando que os graus se mantém mas que a probabilidade da presença de uma aresta é uniforme.

A fórmula EQ de Shen et al. (2009) contém também o termo escalar $1/O_v O_w$. Nesse caso o valor O_i é a quantidade de comunidades a qual pertence o vértice i. Isso permite a aplicação da modularidade estendida para os casos de grafos com comunidades sobrepostas. Vértices que estejam em duas comunidades contribuirão para a modularidade a partir das duas, mas tendo a magnitude da sua contribuição escalada à metade.

2.2 PROPRIEDADES DE REDES COMPLEXAS

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

2.2.1 Mundo pequeno, Anexação preferencial e Liberdade de escala

Largeron et al. (2015) descreve 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 distancia entre os dois vértices que estão a mais arestas de distância, denominada diâmetro, cresce logaritmicamente conforme observa-se exemplos maiores de grafos do 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) define 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) faz 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).

2.2.2 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.

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

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.

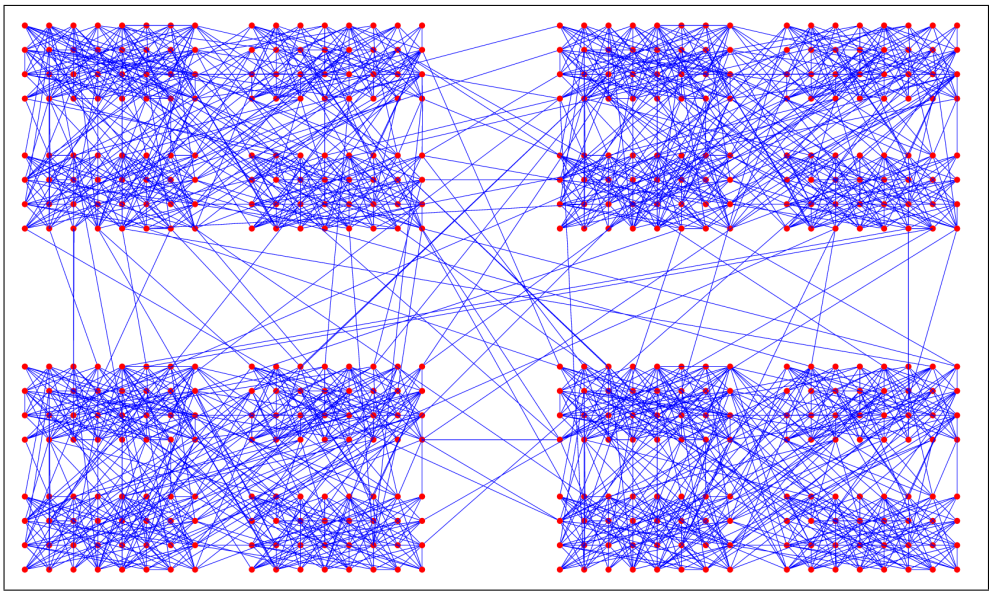
2.2.3 Homofilia e Homogeneidade de comunidades

Homofilia é a tendencia em um sistema de que dois vértices conectados sejam semelhantes entre si (AKOGLU; FALOUTSOS, 2009). Essa propriedade está intimamente ligada á 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 em sistemas do mundo real, não raro 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 de conectam mais do que vértices mais dissemelhantes.

Figura 2 – Exemplo de grafo com comunidades hierárquicas



Fonte: Fortunato (2010)

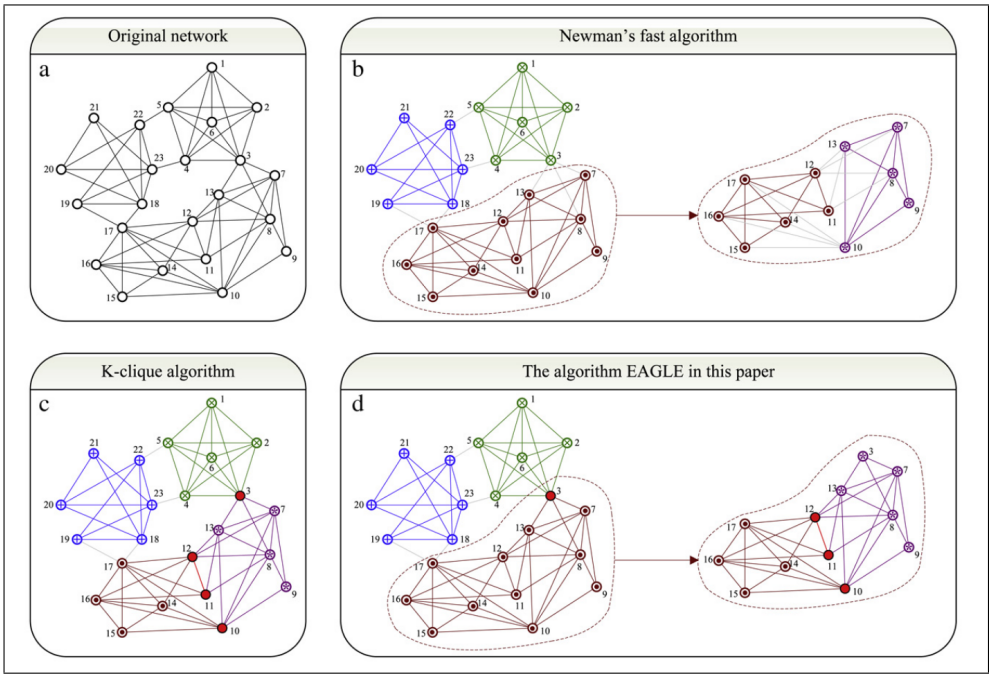
2.2.4 Agrupamentos hierárquicos e sobreposições

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 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 propriedade topológicas relativas a agrupamentos podem ser encontradas se repetindo em 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 sorte de diferentes 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 do que demonstra Shen et al. (2009) não só é possível que um vértice pertença a duas comunidades, é possível que ele pertença a duas comunidades de níveis distintos. Na Figura 3, os resultados de Shen et al.

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



Fonte: Shen et al. (2009)

(2009) são demonstrados no quadro a respeito do algoritmo EAGLE, o vértice denotado como 3 é compartilhado entre duas comunidades de primeira ordem, mas em uma delas o vértice 3 encontra-se como membro de uma comunidade de segunda ordem.

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 grafo implica também na definição de comunidades sobrepostas. 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.

2.3 ESTADO DA ARTE

Existe uma literatura muito prolífica de aplicações dos conceitos de redes complexas, como por exemplo o trabalho de Stegehuis, Hofstad e Leeuwaarden (2016), que faz 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.

2.3.1 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 tendencia 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 para. 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.

2.3.2 Generating Attributed Networks with Communities

Largeron et al. (2015) apresenta um modelo algorítmico de geração de redes complexas com explícita e deliberada para 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 euclideana (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 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 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

O que este trabalho propõe é 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 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 á 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

Esse trabalho se propõe a validar se o modelo a descrito a baixo satisfaz um conjunto de propriedades desejáveis. Para tanto, estende-se a representação clássica de grafo para a inclusão de uma cobertura, bem como a delimitação de dos vértices como vetores em um espaço de p dimensões.

3.1.1 A Representação do grafo

O model gera um grafo com atributos e uma cobertura de comunidades hierarquicamente dispostas.

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

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 níveo 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 folha. 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 flat(C_n) é um conjunto com todas as comunidades do grafo, incluindo a comunidade global. E o mapa L, que 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 a 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 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.

3.1.2 Propriedades desejáveis do modelo

Largeron et al. (2015) implementa 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

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 consigo.	$\forall C(L_C = n \implies \text{flat}(C) = \{ C \})$
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 seus componentes	$\forall C(L_C = 0 \implies V_{C_0} = C_0)$
Para todos os vetores do grafo, existe uma comunidade folha a qual ela pertence	$\forall v(\exists C(v \in C \land L_C = n))$
Pra todo vértice v, pra todo l, existe uma comunidade C que contenha o vértice e seja do nível l	$\forall v \forall l(\exists C(v \in V_{C_x} \land L_C = l))$
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	$\exists K(K \in \mathbb{I}^{n-1} \land \forall C(L_C < n \implies K_{L_C} = \lvert C \rvert))$
Toda a comunidade tem pelo menos uma componente e engloba pelo menos um vértice	$\forall C(\lvert C \rvert \geq 1 \land \lvert V_C \rvert \geq 1)$

Fonte: elaborado pelo autor

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 de 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 explicita das comunidades com base na similaridade dos vértices. Para isso, a similaridade dos vértices é definida com base na distância euclideana 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 euclideana. 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 grupamentos que se sobrepõe 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 caraterizados 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 influencia 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 ás 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 multi dimensional de inteiros maiores que 1. Isso se deve à construção de uma árvore de comunidades hierarquicamente aninhadas.

Quadro 2 – Características da modelagem

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 a 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 a 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.
$\mathcal{A} \in \{\, a \in \mathbb{Q} \mid a > 0 \,\}^{\lvert\mathcal{A}\rvert}$	Vetor de desvios padrão dos atributos dos vértices.
$K \in \{\, k \in \mathbb{N} \mid k \geq 2 \,\}^{\lvert K\rvert}$	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 euclideana 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.

O processo da inicialização se divide em gerar a nuvem de pontos e inicializar as comunidades. A linha quatro inicializa $\mathcal{V}$ com um conjunto vazio, e o laço de repetição das linhas 6 á 10 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

Quadro 3 – fase 1 do modelo

```
1  Output: G_p = (V, E, C_n, R)
2  V ← ∅
3  E ← ∅
4  while |V| < N do
5  begin
6      v ← (N(0, σ_{A_0}), N(0, σ_{A_1}), ..., N(1, σ_{A_{|A|-1}}))
7      V ← V ∪ { v }
8  end
9  Function cover(l, p)
10 begin
11     if l = |K| then
12     begin
13         for v ∈ p do
14         begin
15             p' ← { v' ∈ p | { v, v' } ∉ E ∧ { v, v' } ∉ E ∧ v' ≠ v }
16             s ← Rand_Uni([1, |p'|])
17             for v' ∈ Sample(p', s) do E ← E ∪ (v, v')
18         end
19         return p
20     end
21     s ← NbRep × ∏_{i=l}^{|K|-1} K_i
22     p' ← Sample(p, min { s, |p| })
23     k ← K Medoids(p', K_l)
24     c ← { cover(l + 1, q) | q ∈ k }
25     p' ← { Rand_Uni(c') | c' ∈ V_c }
26     E ← E ∪ { { p'_i, p'_{i+1} } | i ∈ { 1, 2, ... | p'| - 1 } }
27     return c
28 end
29 C_n = cover(0, V)
30
31 for C_i ∈ flat(C_n) do
32     if C_i = flat(C_i) then R_{C_i} ← C_i
33     else R_{C_i} ← ∅
34 G_p ← (V, E, C_n, R)
35 return G_p
```

Fonte: elaborado pelo autor

característica de que a comunidade que se está processando é ou não folha, isso é, se ela possuirá ou não subdivisões internas. Na linha 14 é feita essa ramificação, considerando que l, um parâmetro de controle que é incrementado a cada chamada recursiva. Se l for igual á cardinalidade de k, isso indica que se está processando o último nível a ser gerado, uma folha, o comportamento deixa de ser recursivo.

Este último nível gerado compõe um conjunto de arestas entre os membros da comunidade folha. Nas linhas 16 até 20 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 ajacente. Nas linhas 17 e 18 uma quantidade aleatória das arestas possíveis são construídas.

As funções Rand_{Uni} e Sample são duas funções de escolha aleatória uniformes. $\text{Sample}(P, l)$ escolhe um sub conjunto de P com l elementos uniformemente distribuído, i.e., todos os membros de P tem a mesma chance de estar presente no conjunto construído. $\text{Rand}_{\text{Uni}(P)}$ 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 22 até 28. 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, buscasse 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, é realizado 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 $\text{cover}(l, p)$ 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 á 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 á quantidade de vértices do grafo final, isso é, o parâmetro N. Para tanto, as principais distinções do modelo proposto para com o modelo de Largeron et al. (2015) são para possibilidade de paralelização do processo.

Quadro 4 – fase 2 do modelo, construção dos lotes

Output: $B \subset \{ B' \mid B' \subset \mathcal{V} \}$

$B' \leftarrow \{ v \in \mathcal{V} \mid \neg\exists v'(\{ v, v' \} \in \mathcal{E}) \}$
$B_s \leftarrow \left\lfloor \frac{|\text{flat}(C_n)|}{2} \right\rfloor$
$B'_s \leftarrow (B_s, 2B_s, 4B_s, \ldots, \left\lceil \log_2 \frac{5000}{B_s} \right\rceil B_s)$
$B \leftarrow \emptyset$
for $s \in B'_s$ **do**
begin
    $B_i \leftarrow \{ sample(B', s) \}$
    $B' \leftarrow B' \setminus B_i$
    $B \leftarrow B \cup B_i$
end
while $|B'| > 5000$ **do**
begin
    $B_i \leftarrow B \cup \{ sample(B', \text{Rand}_{\text{Uni}}(5000, 5001, 5002, \ldots, 10000)) \}$
    $B' \leftarrow B' \setminus B_i$
    $B \leftarrow B \cup B_i$
end
$B \leftarrow B \cup B'$

Fonte: elaborado pelo autor

A implementação 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 potencias 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 – fase 2 do modelo, processamento dos lotes

```
1  Output: G = (V, E, C_n)
2
3  Function introduce(v, G_p)
4  begin
5      C_c ← chooseCommunities(v, G_p)
6
7      t_E ← ∅
8      for C_i ∈ C_c do t_E ← t_E ∪ edgesWithin(v, G_p, C_i, |C_c|)
9      t_E ← t_E ∪ edgesBetween(v, G_p, C_c, |t_E|)
10     t_C ← { (v, C_i) | C_i ∈ C_c }
11     return t_C , t_E
12 end
13
14 for b ∈ B do
15 begin
16     for v ∈ b do # esse laço pode ser realizado paralelamente
17     begin
18         t_C , t_E ← introduce(v, G)
19         T_C ← T_C ∪ t_C
20         T_E ← T_E ∪ t_E
21     end
22     E ← E ∪t_E
23     C_n ← buildCover(C_n, T_C)
24     R ← electRepresentants(C_n)
25 end
26 G ← (V, E, C_n)
27 return G
```

Fonte: elaborado pelo autor

Nesse processo, são consumidos os lotes, de forma que a função 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 introduce em si se dá pela escolha de um conjunto

de comunidades C_c por meio da função chooseCommunities$(v, \mathcal{G}_p)$, seguido pela geração de arestas. São geradas as arestas internas a comunidade dentro de uma função edgesWithin$(v, \mathcal{G}_p, C_i)$ e as externas ás comunidades edgesBetween$(v, \mathcal{G}_p, C_c, |t_{\mathcal{E}}|)$. Ambas as funções de geração de arestas retornam um conjunto de pares não ordenados.

Assumindo que as implementações dessa funções se dá 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 emerjam naturalmente.

3.2.3.1 Seleção de comunidades

A seleção de comunidades, realizada dentro da função 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 – fase 2 do modelo, função chooseCommunities$(v, \mathcal{G}_p)$

1 **Function** chooseCommunities$(v, \mathcal{G}_p)$
2 $P \leftarrow$ o conjunto $\{ (C_i, r) \mid C_i \in \text{flat}(C_n) | r \in R_{C_i} \}$ 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 pared ordenados de comunidade e representante são ordenados pela função de semelhança d.

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

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 esparso. Isso é, a função d é dependente do contexto de qual comunidade se está comparando, e θ controla a proporção entre considerar a distância euclideana ou a diferença em um eixo específico. $\theta = 0$ indicando que é considerada apenas a distância euclideana 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 inercia de um conjunto de pontos, como utilizado por Largeron et al. (2015), pode ser expressa como uma soma da inercia consistindo apenas uma dimensão por vez.

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

Considerando n como o indice 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 inercia.

Com a ordenação definida, a função escolhe uma comunidade C á 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 $E_{\text{wth}}^{\max}$ e a quantidade 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 – fase 2 do modelo, função $\text{edgesWithin}(v, \mathcal{G}_p, C, n)$

```
1 Function edgesWithin(v, G_p, C, n)
2       m ← min(E^max_wth, |V_C|)
3       e ← Rand_PL(1, 2, 3, ..., ⌈∛m⌉)
4       W ← ∅
5       for i ∈ {1, 2, 3, ..., e} do W ← W ∪ {Rand_EdgeWth(V_C \ W)}
6       return {{v, u} | u ∈ W}
7 end
```

Fonte: elaborado pelo autor

Uma quantidade e de arestas é gerada utilizado 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.

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

As arestas que ligam o vértice v á outros com os quais ele não compartilha comunidades se dá conforme a função $edgesBetween(v, \mathcal{G}_p, C_c, m)$, 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.

Quadro 8 – fase 2 do modelo, função $edgesBetween(v, \mathcal{G}_p, C_c, m)$

```
1 Function edgesBetween(v, G_p, C_c, n)
2       p' ← flat(C_n) \ C_c
3       p ← ⋃_{C∈p'} {{C, r} | r ∈ R_C}
4       m ← min(E^max_btw, |p|, n)
5       e ← Rand_PL(0, 1, 2, ..., m)
6       W ← ∅
7       for i ∈ {1, 2, 3, ..., e} do W ← W ∪ {Rand_EdgeBtw(p \ W)}
8       return {{v, u_0} | u ∈ W}
9 end
```

Fonte: elaborado pelo autor

O máximo de arestas é definido como m sendo o mínimo entre a cardinalidade conjunto de adjacências possíveis, a quantidade de arestas internas á comunidades, e o

parâmetro $E_{\text{btw}}^{\max}$. 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. Note-se de que com a adaptação que foi realizada a função d é dependente da comunidade que contextualiza a semelhança entre os vértices.

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

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 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 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 $|V_C|$.

É relevante ressaltar de que esses representantes não precisam de 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 distancia 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 – fase 3 do modelo, adição final de arestas

```
1  Output: G = (V, E, C_n)
2
3  l ← max({ L_C | C ∈ flat(C_n) })
4  while |E| < MTE ∧ G ≠ K_|V| do
5  begin
6        T ← { {v, v'} | v, v', v'' ∈ V | {v, v''} ∈ E ∧ {v', v''} ∈ E ∧ {v, v'} ∉ E }
7        T' ← { e ∈ T | ∃C ∈ flat(C_n)(∀v ∈ e(v ∈ C) ∧ L_C = l) }
8        if T' = ∅ then l ← l − 1
9        else E ← E ∪ { Rand_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 serás um laço infinito.

O processo interno ao loop é a identificação das arestas que se adicionadas ao grafo completariam mais um triangulo. 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 euclideana. Isso é, qualquer aresta adicionada nesse processo que ligue dois vértices a e b que compartilham um vizinho c não vai ter uma distancia

maior que soma das duas arestas já presentes no grafo.

$$0 \leq d(a,b) \leq d(a,c) + d(b,c) \tag{3.6}$$

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 influencias da parametrização melhor exploradas com resultados experimentais.

4 RESULTADOS EXPERIMENTAIS

REFERÊNCIAS

AKOGLU, L.; FALOUTSOS, C. Rtg: A recursive realistic graph generator using random typing. In: SPRINGER. **Joint European Conference on Machine Learning and Knowledge Discovery in Databases**. [S.l.], 2009. p. 13–28.

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**. [S.l.: s.n.], 2019. p. 1–14.

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.