UNIVERSIDADE FEDERAL DO PARANÁ

BRUNA VELLO COLNAGO

UMA PROPOSTA PARA A FORMALIZAÇÃO DO PROBLEMA DE CLUSTERIZAÇÃO EM GRAFOS

CURITIBA

2012

UNIVERSIDADE FEDERAL DO PARANÁ

BRUNA VELLO COLNAGO

UMA PROPOSTA PARA A FORMALIZAÇÃO DO PROBLEMA DE CLUSTERIZAÇÃO EM GRAFOS

Dissertação apresentada ao Curso de Pós-Graduação em Informática, Área de Concentração em Algoritmos, Departamento de Informática, Setor de Ciências Exatas, Universidade Federal do Paraná, como parte das exigências para a obtenção do título de Mestre em Ciência da Computação.

Orientador: Prof. Dr. André Luiz Pires Guedes

CURITIBA

2012

Colnago, Bruna Vello
Uma proposta para a formalização do problema de clusterização em grafos / Bruna Vello Colnago. – Curitiba, 2012.
160 f.: il., tab.

Dissertação (mestrado) – Universidade Federal do Paraná, Setor de Ciências Exatas, Programa de Pós-Graduação em Informática.
Orientador: André Luiz Pires Guedes

1. Teoria dos grafos. 2. Algoritmos. I. Guedes, André Luiz Pires. II. Universidade Federal do Paraná. III. Título.

CDD: 005.1

Ministério da Educação
Universidade Federal do Paraná
Programa de Pós-Graduação em Informática

PARECER

Nós, abaixo assinados, membros da Banca Examinadora da defesa de Dissertação de Mestrado em Informática, da aluna Bruna Vello Colnago, avaliamos o trabalho intitulado, *"UMA PROPOSTA PARA A FORMALIZAÇÃO DO PROBLEMA DE CLUSTERIZAÇÃO EM GRAFOS"*, cuja defesa foi realizada no dia 31 de agosto de 2012, às 17:00 horas, no Auditório do Departamento de Informática do Setor de Ciências Exatas da Universidade Federal do Paraná. Após a avaliação, decidimos pela aprovação do candidato.

Curitiba, 31 de agosto de 2012.

Prof. Dr. André Luiz Pires Guedes
DINF/UFPR – Orientador

Prof. Dr. Murilo Vicente Gonçalves da Silva
UTFPR – Membro Externo

Prof. Dr. André Vignatti
DINF/UFPR – Membro Interno

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO PARANÁ
Setor de Ciências Exatas
Coordenação de Pós-Graduação em Informática

Aos meus pais Christina e Luiz que são a razão de tudo.

AGRADECIMENTOS

Aos meus pais, irmãos e namorado, por todo amor, carinho e compreensão que recebi ao longo desse trabalho.

Aos meus tios Ary e Marília, por me receberem em sua casa todas as vezes que eu precisei de colo.

Ao Professor Dr. André Luiz P. Guedes, meu orientador nessa dissertação, pela paciência, dedicação, ajuda e compreensão em tudo que precisei nesse trabalho e também pela confiança depositada em mim.

Ao professor Dr. Renato Carmo, por me incentivar a "não entrar no balaio de gato"e "transformar o problema em computação".

Aos demais professores e funcionários do Departamento de Informática da UFPR, por sua dedicação.

Aos colegas de mestrado, por todo carinho com que fui recebida. Em especial a Aléssio, Alexandre, Francieli, Josiney, Leandro e Sílvio, por todos os momentos felizes que me proporcionaram.

Ao também colega de mestrado Jaime Cohen e ao querido amigo Giovanni, pelas sugestões e dicas.

Aos meus grandes amigos Ana, Alvaro, Felipe, Janice, João, Laila, Letícia e Victor, por não me abandonarem mesmo quando estive ausente por tanto tempo.

A CAPES, pelo auxílio financeiro.

A todos aqueles que direta ou indiretamente tenham contribuído para a execução dessa dissertação de mestrado.

Ciência da computação tem tanto a ver com o computador como a Astronomia com o telescópio, a Biologia com o microscópio, ou a Química com os tubos de ensaio. A Ciência não estuda ferramentas, mas o que fazemos e o que descobrimos com elas.

Edsger Wybe Dijkstra

RESUMO

A possibilidade de agrupar dados para descobrir padrões e correlações interessantes é muito importante em diversas áreas do conhecimento. Essa tarefa pode ser realizada de forma automática através dos métodos de clusterização. Quando os dados apresentam uma estrutura de grafos, como no caso de redes sociais, esse processo é chamado clusterização em grafos. A importância da clusterização em grafos em diversas áreas levou vários cientistas a desenvolver algoritmos paralelamente. Por isso, os artigos apresentam assunções que muitas vezes são incompatíveis, resultando em uma falta de consenso sobre quais as propriedades que caracterizam o resultado de um procedimento de clusterização em um grafo específico. Isto é, não há uma fundamentação teórica para caracterização do que seria uma solução válida para um determinado grafo.

Esse trabalho propõe uma definição formal do problema de clusterização em grafos. É desejável que essa formalização seja robusta o suficiente para descrever o problema resolvido por uma grande parte dos algoritmos de clusterização em grafos. Nesse trabalho, o problema de clusterização em grafos é descrito como o problema de encontrar uma solução que satisfaça um conjunto de restrições e que minimize uma função objetivo. O conjunto de restrições e a função objetivo são utilizados para definir quais a características desejáveis da solução de clusterização natural. Com isso, é mostrado que essa formalização engloba vários algoritmos de clusterização em grafos. Por fim, é apresentada uma solução geral exaustiva para o problema proposto. Como essa solução é muito custosa, esse trabalho propõe combinações das características do conjunto de restrições e da função objetivo a fim de reduzir o espaço de busca do algoritmo que é solução geral do problema.

Palavras-chave: Clusterização em grafos, problema computacional.

ABSTRACT

The possibility of grouping data to discover interesting patterns and correlations is fundamental in many areas of study. This task can be automatically performed through the use of clustering methods. When the data show a graph structure, such as in the case of social networks, this process is called graph clustering. The importance of graph clustering in several areas has led many researchers to develop algorithms in parallel. Therefore, the papers present assumptions that are often incompatible which result in a lack of consensus on the features that characterize the results of a clustering procedure in a specific graph. Thus, there is a lack of a fundamental theory for the characterization of a valid solution for a given graph.

This paper proposes a formal definition of the graph clustering problem. It is desirable that this formalization is robust enought in order to address the problem associated with a large number of graph clustering algorithms. In this paper, the graph clustering problem is described as the problem which involves finding a solution that satisfies a given set of constraints and minimizes a given objective function. The set of constraints and the objective function are used to define the desirable characteristics of natural clustering solutions. Thus, it is shown that this formalization is able to encompass various graph clustering algorithms. Finally, it's presented a general exhaustive solution to the aforementioned problem. As this solution is costly, this paper also proposes the combination of the characteristics of the set of constraints and objective function in order to reduce the search space of the algorithm that the general solution to the problem.

Keywords: Graph clustering, computational problem.

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE ALGORITMOS

LISTA DE SÍMBOLOS

G — Grafo

$\omega(e)$ — Peso de uma aresta e

$V(G)$ — Conjunto de vértices do grafo G

$E(G)$ — Conjunto de arestas do grafo G

ω_G — Função peso do grafo G

A — Matriz de adjacências de um grafo

$A(i,j)$ — Elemento da matriz de adjacencias na linha i e coluna j.

P — Passeio (inclusive caminhos e ciclos

$E(P)$ — Conjunto de arestas internas a P

$\Gamma(v)$ — Vizinhança do vértice v

$d(v)$ — Grau do vértice v

$dist(u,v)$ — Distância entre os vértices u e v

$G[S]$ — Subgrafo induzido por S em G

$Diam(G)$ — Diâmetro do grafo G

$Cint(G)$ — Cintura do grafo G

$d(G)$ — Grau médio de G

$\omega(G)$ — Custo de G

$\delta(G)$ — Densidade do grafo G

$G - S$ — Remoção de um conjunto S de vértices ou arestas de G

$k(G)$ — Conexidade de G

$\lambda(G)$ — Aresta conexidade de G

$E(S,X)$ — Conjunto de arestas que liga S a X

$\omega(S,X)$ — Custo do conjunto de arestas que liga S a X

$E(S,V(G)\backslash S)$ — Corte definido por S em G

T — Árvore

T_G — Árvore de cortes mínimos

C — Solução de clusterização em grafos

C_i — Cluster pertencente a clusterização C

$Clust(G)$ — Conjunto de todas as soluções de clusterização em grafos que são exclusivas, completas e que induzem subgrafos conexos no grafo G

$dist(C_i, C_j)$ — Distância entre os clusteres C_i e C_j

$dist(v, C_j)$ — Distância entre um vértice $v \notin C_j$ e um cluster C_j

$d_{int}(v, C_i)$ — Grau interno de v em relação ao cluster C_i

$d_{ext}(v, C_i)$ — Grau externo de v em relação ao cluster C_i

$E(C_i)$ — Conjunto de arestas intra-cluster do cluster C_i

$E(C)$ — Conjunto de arestas intra-cluster da solução de clusterização C

$E'(C_i)$ — Conjunto de arestas inter-cluster do cluster C_i

$E'(C)$ — Conjunto de arestas inter-cluster da solução de clusterização C

$E(C_i, C_j)$ — Conjunto das arestas inter-cluster com extremos em C_i e C_j

$\delta(C_i)$ — Densidade de um cluster C_i

$Cp(C_i)$ — Compacidade Cp de um cluster C_i

$sim(u, v)$ — Similaridade entre dois vértices u e v.

$Cp^*(C_i)$ — Compacidade Cp^* de um cluster C_i

$D(C)$ — Índice de Dunn de uma solução C

$DB(C)$ — Índice de Davies Bouldin de uma solução C

$\rho(C_i)$ — Densidade relativa de um cluster C_i

$cobertura(C)$ — Cobertura de uma solução de clusterização C

$\psi(S)$ — Expansão do corte definido por S em um grafo G

$\psi(G)$ — Expansão de um grafo G

$\psi(C_i)$ — Expansão de um cluster C_i

$\psi(C)$ — Expansão de uma solução C

$\phi(S)$ — Condutância do corte definido por S em um grafo G

$\phi(G)$ — Condutância de um grafo G

$\phi(S, C_i)$ — Condutância do corte definido por S no grafo induzido por C_i

$\phi(C_i)$ — Condutância do cluster C_i

$\phi(C)$ — Condutância de uma solução C

(α, ϵ) — Medida de qualidade bi-critério (α, ϵ)

$performance(C)$ — Performance de uma solução C

$e(C_i, C_j)$ — Fração de extremos de arestas entre os clusteres C_i e C_j que pertencem a C_i

b_i — Fração de todos os extremos de arestas que pertencem a C_i

$Q(C)$ — Modularidade de uma solução C

$C_D(v)$ — Centralidade C_D de um vértice v baseada no grau

$C'_D(v)$ — Centralidade C'_D de um vértice v baseada no grau

$C_C^{-1}(v)$ — Centralidade C_C^{-1} de um vértice v baseada em proximidade

$C'_C(v)$ — Centralidade C'_C de um vértice v baseada em proximidade

$\sigma(u, v)$ — Número de caminhos mínimos entre u e v

$\sigma(u, v, w)$ — Número de caminhos mínimos entre u e v que contém w

$b(u, v, w)$ — *Betweenness* de w para u e v

$C_B(w)$ — Centralidade C_B de um vértice w baseada betweeness

$C'_B(w)$ — Centralidade C'_B de um vértice w baseada em betweeness

$\epsilon(u, v)$ — Eficiência com que dois vértices u e v se comunicam em uma rede

$\epsilon(G)$ — Eficiência do grafo G

$C^I(e)$ — Centralidade se informação de uma aresta e

$\sigma(u, v, e)$ — Número de caminhos mínimos entre u e v que contém a aresta e

$C_B(e)$ — Centralidade *betweenness* de uma aresta e

2^D — Conjunto dos subconjuntos de D, conjunto das partes

$Part(D)$ — Conjunto de partições do conjunto D

$Refinamento$ — Relação refinamento definida no conjunto $Part(D)$

$Raiz$ — Partição do conjunto D composta por um único bloco igual a D

$Folhas$ — Partição do conjunto D composta por $|D|$ blocos unitários

SUMÁRIO

1 INTRODUÇÃO

Com a evolução das ciências, bem como com o constante desenvolvimento e aprimoramento de instrumentos e de metodologias, é possível obter dados cada vez mais precisos e em uma taxa de amostragem cada vez maior. Isso, aliado ao crescimento da capacidade de armazenamento, faz com que o pesquisador se depare com grandes massas de dados, tornando o trabalho do cientista mais complexo e a averiguação visual muito mais difícil.

Quando se observa uma massa de dados heterogênea, geralmente é possível identificar subconjuntos dos dados que são muito mais homogêneos que a massa como um todo. Esses subconjuntos são chamados grupos naturais e são caracterizados por seus elementos serem muito similares e muito diferentes de qualquer elemento não pertencente ao conjunto. Por isso, cada elemento do conjunto pode ser bem representado pelo grupo natural ao qual pertence, isto é, pela sua classificação. Isso diminui a granularidade do problema e aumenta a facilidade de fazer assunções visuais.

Essa possibilidade é ilustrada por Carvalho, Mata e Resende (2007) através de um conjunto de 141 empresas caracterizadas pelo número de empregados e pela produtividade média. Observe na Figura 1.1 (acima) que o gráfico é visualmente confuso e poluído. Considerando uma empresa em isolado, como classificá-la? Essa é uma tarefa simples para os pontos localizados nos valores extremos, como os coloridos de verde e azul, mas não tão intuitiva para aqueles que estão mais próximos à parte central do gráfico, como os coloridos de vermelho. Considere, agora, que essas empresas foram agrupadas como na Figura 1.1 (abaixo). Como o número de grupos é consideravelmente menor se comparado ao de empresas, a inspeção visual é mais simples. Dessa forma, é imediato dizer se uma empresa tem produtividade alta ou baixa e se tem muitos ou poucos empregados.

Uma classificação é um conjunto de classes que divide um espaço de elemen-

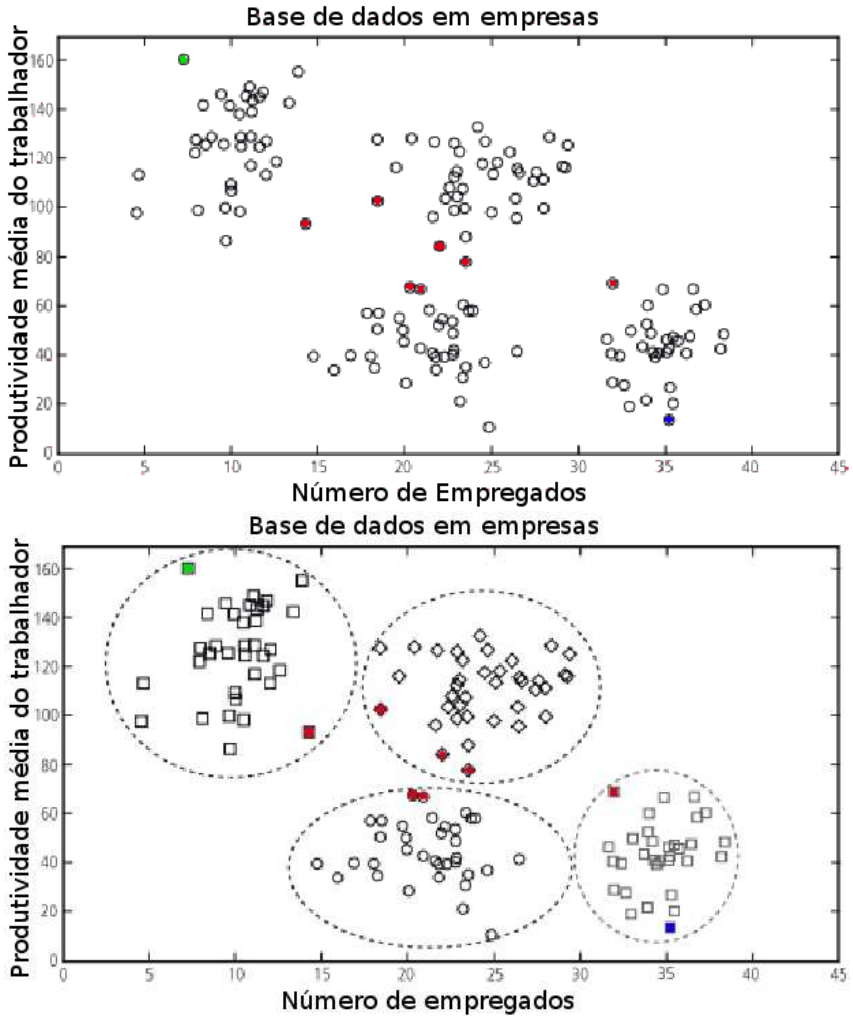

FIGURA 1.1 — (acima) Informações sobre número de empregados e produtividade média do trabalhador para uma base de dados de 141 empresas. (abaixo) Identificação visual de quatro grupos homogêneos de empresas, de acordo com o número de empregados e a produtividade média dos trabalhadores. Imagens modificadas de Carvalho, Mata e Resende (2007).

tos, de forma que cada classe é um grupo natural. Duas classificações de um mesmo conjunto de dados podem ser diferentes. Um conjunto de animais, por exemplo, pode ser naturalmente classificado segundo seus filos ou classes. Mas poderia, também, ser classificado dentre animais perigosos e não perigosos. Note que o conceito de grupo natural é diferente nessas duas situações. Disso, decorre que os critérios que definem a similaridade entre dois elementos podem ser diferentes conforme o objetivo de pesquisa, ainda que para um mesmo conjunto de entrada.

Uma classificação pode ser obtida através da clusterização, que é o processo de identificar grupos em uma massa de dados baseado na similaridade entre os elementos, que é uma medida do quanto seus atributos são parecidos (JAIN; MURTY; FLYNN, 1999) (LESOT; RIFQI; BENHADDA, 2009). Alguns atributos, entretanto, podem ser inadequados ao objetivo de pesquisa. Quando se deseja classificar um conjunto de animais de acordo com a classificação taxonômica, por exemplo, considerar a cor dos olhos pode levar a erros. Um cachorro de olhos azuis tenderia a ser assinalado a um grupo diferente de um cachorro de olhos negros, que, por sua vez, teria propensão a ser agrupado juntamente a um caranguejo. Outros atributos podem ser desnecessários. A identificação de gatos e cachorros baseada em centenas de atributos morfológicos leva a uma classificação clara, mas poderia ser feita considerando apenas o número de dentes, afinal gatos tem 30 e cachorros 42 (EVERITT, 1993).

Por isso, muitas vezes é necessário pesar os atributos, mudando sua contribuição na similaridade entre os objetos (KIRA; RENDELL, 1992). O pesquisador pode atribuir peso nulo aos atributos prejudiciais ou desnecessários, ou seja, não utilizá-los nas análises subsequentes. Por outro lado, um atributo muito significativo poderia receber um peso maior que atributos não relevantes. Essa manipulação, entretanto, apenas faz sentido quando o pesquisador sabe concretamente o objetivo da pesquisa e tem uma forte intuição da modelagem dos dados. Uma representação ruim dos atributos, por outro lado, pode levar a uma classificação complexa cuja verdadeira estrutura é difícil de discernir.

1.1 Representação e similaridade entre dados através de grafos

Existem casos em que não é possível ou viável aferir a similaridade dentre todos os pares de dados, mas que a similaridade entre dois dados pode ser obtida através da similaridade desses dados com outro dado qualquer. Nessa situação, o conjunto de dados pode ser modelado por um grafo. Um grafo é um conjunto de elementos (os vértices) e um conjunto de relações binárias (arestas) entre esses elementos, cada qual valorada por um peso que caracteriza a intensidade da similaridade ou da dissimilaridade entre os elementos.

Grafos são utilizados em diversas áreas. Na sociologia, por exemplo, redes sociais são grafos usados para modelar a iteração entre indivíduos, representados por vértices na rede. A existência de uma aresta entre um par de vértices indica que há algum tipo de interação social ou vínculo entre eles.

O termo rede é muitas vezes usado como sinônimo para grafo, mas tem um significado muito mais amplo para cientistas através de uma variedade de campos (PORTER; ONNELA; MUCHA, 2009) e é justamente nesse sentido que será usado nesse texto. Girvan e Newman (2002) lista algumas propriedades estatísticas que a maioria das redes compartilha:

- A distância média entre pares de vértices é pequena se comparada ao tamanho da rede. Essa propriedade é denominada efeito mundo pequeno.
- Existem muitos vértices com grau pequeno e apenas uns poucos com grau alto.
- Dois vértices que possuem um vizinho em comum têm uma probabilidade maior de serem vizinhos que dois vértices escolhidos aleatoriamente. Esta propriedade é chamada clusterização ou transitividade de rede.
- As redes são constituídas por grupos de vértices densamente conectados, mas

esparsamente conectados entre si, em uma estrutura de comunidades. Isso pode ser exemplificado pela Figura 1.2, que mostra uma rede que modela a coautoria em artigos. As comunidades são representadas por cores diferentes. Observe que membros de uma mesma comunidade apresentam várias arestas entre si, mas há poucas conexões entre comunidades.

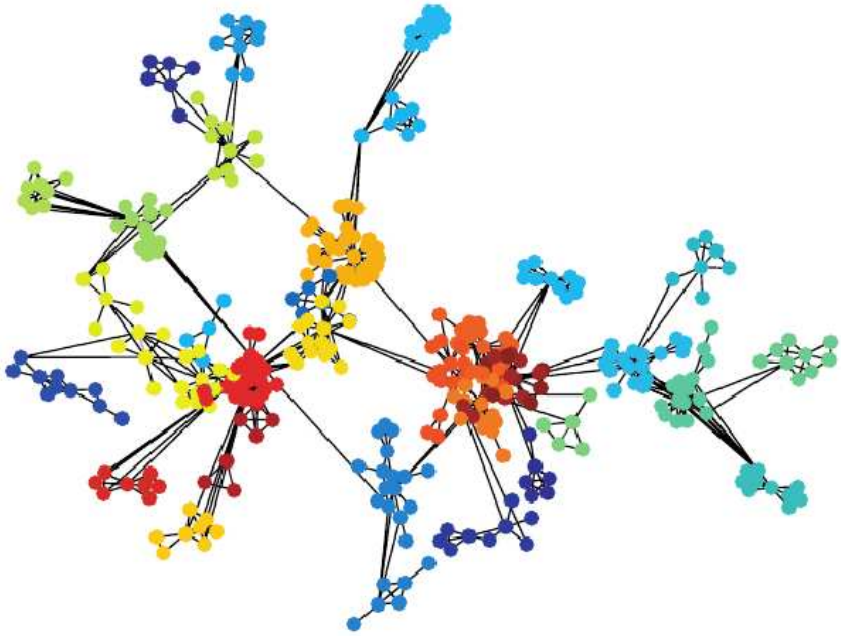

FIGURA 1.2 — Comunidades em uma rede de cientistas, onde as arestas representam coautoria em artigos. Imagem retirada de Porter, Onnela e Mucha (2009).

- As ligações dentro das comunidades tendem a ser fortes e as ligações entre comunidades tendem a ser fracas. Essa propriedade é chamada hipótese de Granovetter (GRANOVETTER, 1973 apud PORTER; ONNELA; MUCHA, 2009).

A noção da existência de comunidades em uma rede é intuitiva. Girvan e Newman (2002) e Porter, Onnela e Mucha (2009) apresentam diversos exemplos do que seriam comunidades: em redes sociais, podem representar agrupamentos sociais, grupos de amigos; em uma rede de citações, podem representar trabalhos relacionados em um único tópico; em uma rede metabólica, podem representar ciclos e outros grupos funcionais; na Web, podem representar páginas sobre temas relacionados. A

capacidade de descobrir e analisar esses grupos permite a compreensão e auxilia a visualização da estrutura de redes (NEWMAN; GIRVAN, 2004).

A definição de comunidade sugere o uso de métodos de clusterização para sua detecção. De fato, segundo Porter, Onnela e Mucha (2009) os cientistas que estudam a detecção da comunidade e os que estudam clusterização estão olhando para a mesma moeda, os dois campos estão avançando em paralelo e que existem inúmeras ligações profundas entre eles.

1.2 Definição do problema

A detecção de comunidades não é um caso simples de particionamento de grafos, pois, em geral, não há informações a priori sobre o número e o tamanho das comunidades e a quantidade de arestas entre comunidades não precisa ser estritamente minimizadas, já que comunidades maiores admitem mais arestas entre comunidades que comunidades pequenas (NEWMAN; GIRVAN, 2004). A detecção de comunidades é a principal razão para o surgimento da clusterização em grafos.

Clusterização em grafos é um nome genérico para uma variedade de métodos matemáticos, estatísticos e de heurísticas que podem ser usados para reconhecer grupos naturais em um grafo. Em outras palavras, a clusterização em grafos é a tarefa de agrupar os vértices baseado na estrutura das arestas, de forma que os grupos induzem subgrafos cuja densidade é maior que a densidade do grafo original.

Note que a definição de comunidade em um grafo é subjetiva, isto é, não há numa definição formal universalmente aceita sobre a validade da caracterização de um conjunto de vértices como um grupo natural. No entanto, é possível afirmar que uma comunidade é um conjunto de vértices cujo subgrafo induzido é conexo.

Uma solução para um problema de clusterização em grafos é uma partição do conjunto de vértices do grafo. Considere o grafo G, uma solução de clusterização em grafos de G é um conjunto $C = \{C_1, C_2, ..., C_n\}$, de subconjuntos do conjunto de vértices, tal que cada vértice pertence a um e somente um C_i e todos os vértices de C_i são alcançáveis uns dos outros. Cada conjunto de vértices $C_i \in C$ é denominado cluster. Quando todos os clusteres pertencentes a C são grupos naturais, C é dita solução de clusterização em grafos natural.

O problema de clusterização em grafos pode ser definido como o problema de encontrar uma solução de clusterização natural em um grafo. Observe, entretanto, que esta definição é dependente da caracterização de grupos naturais. Logo, a descrição de clusterização em grafos herda a informalidade da noção de clusteres naturais. Ou seja, o problema de clusterização em grafos também não apresenta uma definição formal. Isto é, não há uma fundamentação teórica que defina uma padronização para os formatos das instâncias e das respostas do problema e que permita caracterizar o que seria uma solução válida para um determinado grafo. Por isso, existe uma variedade de formulações diferentes de grafo para o uso na clusterização em grafos. Essa diferença de notação entre autores torna o campo de clusterização em grafos confuso, impedindo a prova de algoritmos e a definição de medidas de qualidade universais.

O objetivo desse trabalho é apresentar uma definição formal rígida do problema de clusterização em grafos. Dada a natureza subjetiva do problema, não seria adequado defini-lo tendo como instância apenas o grafo em que se deseja detectar comunidades. Afinal, isso implicaria na necessidade de definir uma caracterização rígida do que seria uma comunidade, o que é impraticável, já que diferentes objetivos de pesquisa levam naturalmente a diferentes noções de comunidade.

Esse trabalho propõe que essa informalidade seja englobada na definição do problema de clusterização em grafos através de um mecanismo formal. O problema

de clusterização em grafos é descrito como um problema de encontrar uma solução de clusterização para um determinado grafo que satisfaça um conjunto de restrições e que minimize uma função objetivo. A escolha de qual função objetivo e conjunto de restrições utilizar depende das características desejadas na solução de clusterização natural. Isso permite que o mesmo problema seja utilizado com diferentes noções do que é uma comunidade em um grafo.

É proposta, também, uma solução geral para o problema de clusterização em grafos. Por fim, é analisado como as características do conjunto de restrições e da função objetivo podem ser combinadas a fim de reduzir o espaço de busca do algoritmo que é solução geral do problema. Isto é, como diminuir a quantidade de possíveis soluções de clusterização em grafos que deve ser inspecionada e, ainda assim, ter a garantia de encontrar a resposta correta para cada instância do problema.

Dentre as contribuições desse trabalho, vale destacar que essa definição formal do problema de clusterização em grafos permite:

- Aferir a validade de uma solução de clusterização em grafos.

- Reescrever alguns dos algoritmos existentes de forma rígida.

É importante ressaltar que apenas os problemas de clusterização que tem como solução uma partição do conjunto de vértices são tratados nessa dissertação. Existem soluções em que vértices não pertencem a nenhuma comunidade ou a mais de uma comunidade ao mesmo tempo, como as soluções difusas ou probabilística. Existem, também, algoritmos que por uma questão de escalabilidade optam por trabalhar apenas uma parte do grafo, os algoritmos locais. Entretanto, esse trabalho também não lida com essa possibilidade, estudando apenas os métodos globais de clusterização em grafos.

1.3 Apresentação da dissertação

O restante desse texto está organizado da seguinte forma:

- O Capítulo 2 apresenta alguns conceitos e definições importantes da teoria de grafos que serão utilizados nos capítulos subsequentes.
- O Capítulo 3 explica o conceito de clusterização em grafos, discute os problemas decorridos da natureza informal do problema. Adicionalmente, são apresentados alguns algoritmos, medidas de qualidade e medidas de centralidade existentes.
- O Capítulo 4 formaliza o problema de clusterização em grafos e mostra como as medidas de qualidade e de centralidade podem ser adaptadas para tornarem-se conjuntos de restrições e funções objetivos. Além disso, os algoritmos estudados no capítulo anterior são avaliados para decidir se a formalização proposta consegue descrever o problema específico resolvido pelo algoritmo.
- O Capítulo 5 apresenta uma solução geral para o problema de clusterização em grafos. Por fim, é analisado como combinar as características do conjunto de restrições e da função objetivo a fim de reduzir o espaço de busca da solução geral. Isto é, a fim de permitir que nem todas as possíveis soluções de clusterização em grafos sejam analisadas e, ainda assim, garantir que pelo menos uma das respostas exatas seja encontrada para cada uma das instâncias do problema de clusterização.
- O Capítulo 6 sumariza as conclusões desse trabalho e propõe alguns trabalhos futuros.

2 CONCEITOS DE TEORIA DOS GRAFOS

Esse capítulo apresenta alguns conceitos importantes para compreensão do desenvolvimento da formalização proposta para o problema da clusterização de grafos. Mais especificamente, o capítulo objetiva introduzir o conceito de grafos e explicar algumas propriedades e conceitos que serão utilizados nos capítulos subsequentes. A teoria de grafos pode ser estudada detalhadamente em West (2000).

DEFINIÇÃO 2.1: Um grafo G é um par (V, E), onde V é um conjunto finito cujos elementos são chamados vértices e E é o conjunto de arestas tal que $E \subseteq \binom{V}{2}$.

Onde $\binom{S}{2}$ representa o conjunto de todos os subconjuntos de S de cardinalidade 2. A definição anterior assume que todas as arestas são equivalentes, não havendo, a princípio, o que se falar de arestas mais importantes que outras. Em vários campos, entretanto, seria interessante indicar que dois vértices possuem uma ligação mais forte ou mais importante que outras ligações no grafo. Para tanto, foram introduzidos os grafos ponderados.

DEFINIÇÃO 2.2: Um grafo ponderado G é uma tripla (V, E, ω), onde (V, E) é um grafo simples e $\omega \colon E \to \mathbb{R}$ é uma função peso.

O peso de uma aresta e, $\omega(e)$, é utilizado para caracterizar a importância da aresta no grafo. Neste trabalho, os pesos serão sempre positivos, ou seja, $\omega \colon E \to \mathbb{R}^+$ e o fato de $\omega(e) > \omega(f)$ significa que a aresta e tem uma propriedade mais forte que a aresta f. Se as arestas indicarem similaridades entre os vértices, por exemplo, dois vértices pertencentes a e seriam mais similares que dois vértices pertencentes a f.

Não é necessário declarar explicitamente os conjuntos de vértices e de arestas e a função peso de um grafo ponderado G. Nesse caso, convenciona-se denotá-los por $V(G)$, $E(G)$ e ω_G, respectivamente. Essa convenção, assim como qualquer propriedade definida para grafos ponderados, pode ser utilizada com grafos simples. Afinal, todo grafo simples pode ser transformado em um grafo ponderado escolhendo ω_G de forma que todos os pesos sejam unitários. Ou seja, a versão ponderada do grafo simples G é $G' = (V(G), E(G), \omega)$, onde $\omega : E(G) \to \{1\}$.

Alguns grafos recebem nomes especiais. O grafo G é denominado vazio quando $|V(G)| = 0$ e G é dito trivial se $|V(G)| = 1$. Se $E(G) = \binom{V}{2}$, G é dito completo, isto é, não é possível adicionar nenhum elemento a $E(G)$ sem adicionar novos elementos a $V(G)$.

2.1 Representações gráfica e matricial de grafos

Os grafos podem ser representados graficamente por diagramas da seguinte forma:

i) Para cada vértice, é desenhado um ponto no plano. Embora geralmente a representação seja plana, pode-se utilizar qualquer espaço euclidiano $\mathbb{R}^n$.

ii) Para cada aresta, um segmento de curva é desenhado entre os pontos correspondentes aos seus vértices.

iii) Se o grafo é ponderado, os pesos de cada aresta são colocados próximos as arestas. Quando todas as arestas têm peso unitário, esse passo pode ser omitido.

Considere o grafo ponderado G_1 tal que $V(G_1) = \{a, b, c, d\}$,

$E(G_1) = \{\{a, b\}, \{a, c\}, \{b, c\}, \{c, d\}\}$ e

$$\omega_{G_1}(e) = \begin{cases} 2, & \text{se } e = \{a, b\} \\ 5, & \text{se } e = \{a, c\} \\ 3, & \text{se } e = \{b, c\} \\ 2, & \text{se } e = \{c, d\}. \end{cases}$$

Considere, também, o grafo simples G_2, tal que $V(G_2) = \{a, b, c, d, e, f, g, h, i, j\}$ e $E(G_2) = \{\{a, b\}, \{a, c\}, \{b, d\}, \{c, d\}, \{e, f\}, \{h, i\}, \{i, j\}\}$. As Figuras 2.3 e 2.4 mostram diagramas correspondentes aos grafos G_1 e G_2, respectivamente. Observe que, como os vértices podem ser dispostos de qualquer maneira, um mesmo grafo pode possuir várias representações. Entretanto, uma representação corresponde a um único grafo.

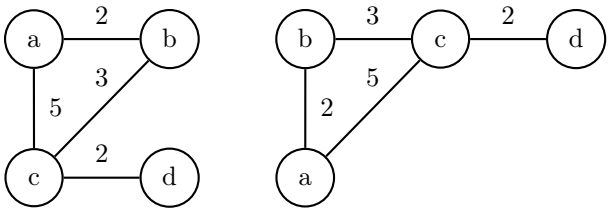

FIGURA 2.3 — Duas representações gráficas do grafo G_1.

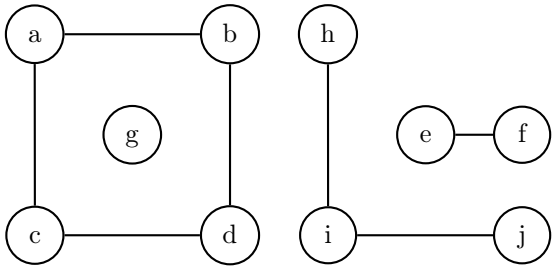

FIGURA 2.4 — Representação gráfica do grafo G_2.

Alternativamente, o grafo ponderado G pode ser representado Computacionalmente por uma matriz de adjacências A de dimensão $|V(G)| \times |V(G)|$ tal que

$$A(i,j) = \begin{cases} \omega_G(\{i,j\}), & \text{se } \{i,j\} \in E(G) \\ 0, & \text{caso contrário.} \end{cases}$$

Observe que a matriz A é simétrica. Isso decorre do fato da aresta ser um conjunto de vértices e não um par ordenado. O grafo G_1, por exemplo, pode ser representado pela matriz de adjacências

$$A = \begin{bmatrix} 0 & 2 & 5 & 0 \\ 2 & 0 & 3 & 0 \\ 5 & 3 & 0 & 2 \\ 0 & 0 & 2 & 0 \end{bmatrix}.$$

2.2 Passeios, caminhos e ciclos

Um passeio é uma sequência de vértices na qual existe uma aresta entre cada par de vértices consecutivos. Isto é, um passeio é uma sequência $P= (v_0, v_1, ..., v_k)$, tal que $v_i \in V(G), 0 \leqslant i \leqslant k$ e $\{v_i, v_{i+1}\} \in E(G), 0 \leqslant i < k$.

Caminhos e ciclos são passeios com propriedades especiais. Um caminho é um passeio sem repetição de vértices, isto é, $\forall i \neq j, v_i \neq v_j$. Um ciclo é um passeio sem repetições de vértices exceto pelos extremos, que são coincidentes, ou seja, $(v_0, v_1, ..., v_{k-1})$ é um caminho em G e $v_k = v_0$, $k \geqslant 3$.

Uma aresta $\{u, v\}$ é dita interna a um passeio P se os vértices u e v são consecutivos na sequência P em alguma ordem. O conjunto de arestas internas a P é notado por $E(P)$ e o tamanho de P é dado pela cardinalidade de $E(P)$, isto é, $|P| = |E(P)|$.

2.3 Relações entre arestas e vértices e propriedades de vértices em um grafo

Um vértice v é extremo de uma aresta e se $v \in e$, nesse caso, a aresta e é dita incidente em v. Duas arestas incidentes em um mesmo vértice são ditas adjacentes e dois vértices extremos de uma mesma aresta são denominados vizinhos ou adjacentes.

O conjunto de vértices adjacentes a v é denotado vizinhança de v, que é representada por $\Gamma(v)$. A soma dos pesos das arestas adjacentes a v é denominada grau do vértice v, que é notado por $d(v)$.

Se existe um passeio $P = (v_0, v_1, ..., v_k)$, então pode-se dizer que P é um passeio entre v_0 e v_k. Os vértices v_0 e v_k são extremos de P. Convenciona-se, ainda, dizer que v_k é alcançável por v_0.

Dados $u, v \in V(G)$, a distância entre u e v é definida pelo tamanho do menor caminho entre u e v em G e é notada por $dist(u, v)$. Caso não exista caminho entre u e v, é convencionado que $dist(u, v) = \infty$.

2.4 Classificações de grafos

Um grafo pode ser classificado segundo vários critérios. Nesse trabalho, serão importantes as classificações relativas à presença de ciclos e a conexidade. Se um grafo contém ciclos ele é dito cíclico, senão, é denominado grafo acíclico.

Um grafo G não vazio é conexo se existe caminho entre todos os vértices do grafo. O grafo vazio é trivialmente conexo.

2.5 Subgrafos

Um grafo H é subgrafo do grafo G se $V(H) \subseteq V(G)$ e $E(H) \subseteq E(G)$. O subgrafo induzido por $S \subseteq V(G)$ em G é

$$G[S] = \left(S, E(G) \cap \binom{S}{2}, \omega_G\right).$$

Se S induz um grafo completo em G, então S é dita uma clique. Por outro lado, se S induz um subgrafo tal que $E(G[S]) = \emptyset$, então S é chamado conjunto independente. Edachery et al. (1999) define uma generalização do conceito de clique: S é uma *distance-k clique* se existe um caminho de tamanho no máximo k entre cada par de vértices em S. Note que uma clique é uma *distance-k clique* para $k = 1$. Note que um conjunto unitário é uma *distance-k clique* para qualquer valor de k.

Um subgrafo conexo H de G é dito maximal se não existe algum subgrafo F de G, tal que H é subgrafo de F, F é conexo e $F \neq G$. Os subgrafos conexos maximais de um grafo são chamados componentes conexas do grafo. A Figura 2.5 representa as componentes conexas do grafo G_2. Cada componente conexa tem seus vértices representados por uma figura geométrica dentre círculos, quadrados, triângulos e losangos.

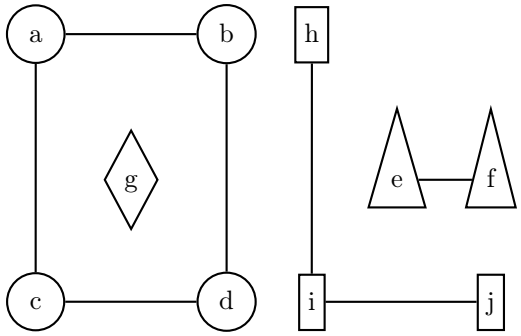

FIGURA 2.5 — Componentes conexas do grafo G_2. Cada componente conexa tem seus vértices representados por uma figura geométrica dentre círculos, quadrados, triângulos e losangos.

2.6 Propriedades de um grafo

Em um grafo conexo G, o tamanho do maior caminho é chamado diâmetro e é representado por $Diam(G)$. Convencionou-se que o diâmetro de um grafo desconexo é infinito. Analogamente, em um grafo cíclico G, o tamanho do menor ciclo é dito cintura, que é notada por $Cint(G)$. Em grafos acíclicos, diz-se que a cintura mede infinito.

O grau médio de um grafo G, $d(G)$, é a média aritmética entre o grau de todos os vértices de G. O grau mínimo e o grau máximo do grafo são, respectivamente, o menor e o maior grau dentre todos os vértices do grafo. O custo de um grafo G é representado por $\omega(G)$ e tem seu valor dado pela soma do peso de todas as arestas de G.

A densidade de um grafo G é a fração do número de arestas em G em relação ao número de arestas que G teria se fosse completo, isto é,

$$\delta(G) = \begin{cases} 1, & \text{se } |V(G)| \in \{0, 1\} \\ \dfrac{|E(G)|}{\left|\binom{V(G)}{2}\right|}, & \text{caso contrário.} \end{cases}$$

Note que a densidade de um grafo completo é 1.

2.7 Conexidade e cortes

A remoção de um conjunto S de vértices ou de um conjunto R de arestas de um grafo G resultam, respectivamente, em

$$G - S = (V(G)\backslash S, E(G)\backslash\{e : e \in E(G) \wedge e \cap S \neq \emptyset\}, \omega_G),$$

e

$$G - R = (V(G), E(G)\backslash R, \omega_G).$$

A conexidade de G, $k(G)$, é o tamanho do menor conjunto de vértices cuja remoção de G resulta em um grafo desconexo. Se G é um grafo completo, é convencionado que $k(G) = n-1$. Analogamente, a aresta-conexidade de G, $\lambda(G)$, é o tamanho do menor conjunto de arestas que deve ser removido de G de forma que o resultado seja um grafo desconexo. Se G é um grafo desconexo ou se é trivial, é convencionado que $k(G) = 0$. Se $k(G) = l$ ou $\lambda(G) = l$, então G é dito, respectivamente, l-conexo e l-aresta conexo.

Considere dois conjuntos $S, X \subseteq V(G)$, o conjunto de arestas que liga S a X é dado por

$$E(S, X) = \{\{u, v\} \in E(G) | u \in S \wedge v \in X\}.$$

O custo do conjunto de arestas que liga S a X é calculado pela soma dos pesos das arestas, isto é

$$\omega(S, X) = \sum_{\substack{u \in S \\ v \in X}} \omega(\{u, v\}).$$

Se $\{S, X\}$ é uma partição de $V(G)$, isto é, se $S \cap X = \emptyset$ e $S \cup X = V(G)$, então $E(S, X)$ é dito um corte em G. Geralmente, o corte pode ser definido apenas pelo conjunto S e, nesse caso, o corte é dado por $E(S, V(G)\backslash S)$.

A Figura 2.6 mostra o corte definido por $\{a, b\}$ no grafo G_1. Se $u \in S$ e $v \in V(G)\backslash S$, pode-se dizer que $E(S, V(G)\backslash S)$ é um corte entre u e v.

$E(S, V(G)\backslash S)$ é um corte-mínimo do grafo G se ele apresenta menor custo

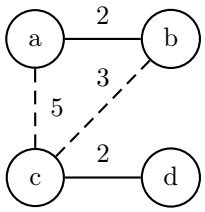

FIGURA 2.6 — Corte definido por $\{a, b\}$ em G_1. As arestas que pertencem ao corte encontram-se em linhas tracejadas.

dentre todos os cortes de G. Analogamente, um corte entre u e v é mínimo se ele apresenta o menor custo dentre todos os cortes existentes entre u e v. O custo de um corte também é chamado de tamanho do corte.

Hartuv e Shamir (2000) denota por altamente conectado o subgrafo H de G cuja conectividade, $k(H)$, é maior que a metade do número de vértices de H.

Segundo Matula (apud DONGEN, 2000), um subgrafo S de G é um:

- k-*bond* se o grau mínimo de S for k.
- k-componente se S for k-aresta-conexo.
- k-bloco se S for k-conexo.

Quando se deseja referir-se genericamente a k-*bond*, k-componente e k-bloco pode-se utilizar o termo k-objeto. Observe que cada k-objeto está contido em um $(k+1)$-objeto. Note, ainda, que, para um k específico, cada k-bloco é um subgrafo de um k-componente que é subgrafo de em k-*bond*. Ou seja, os k-objetos são refinamentos sucessivos.

2.8 Árvores, árvores geradoras e árvores de corte mínimo

Um grafo que é, ao mesmo tempo, conexo e acíclico é chamado árvore. Toda árvore T tem $|E(T)| = |V(T)| - 1$. Quaisquer duas dessas três propriedades definem

uma árvore (WEST, 2000).

Um vértice v é uma folha de T se $|\Gamma(v)| = 1$. Note que se $|V(T)| > 1$, então T contém ao menos uma folha.

Uma árvore geradora de G é uma árvore T, tal que, T é subgrafo de G e $V(G) = V(T)$. Todo grafo conexo tem ao menos uma árvore geradora. As árvores geradoras mínimas de G são àquelas cujo custo é mínimo. A Figura 2.7 mostra as árvores geradoras do grafo G_1.

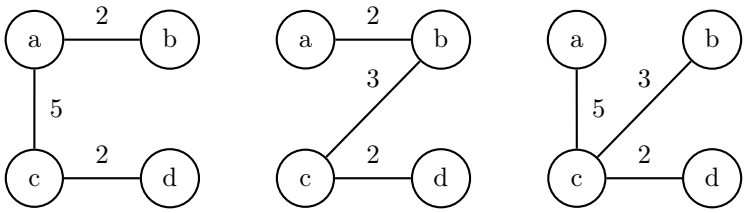

FIGURA 2.7 — Árvores geradoras do grafo G_1.

Dado um grafo G, Gomory e Hu (1961) definiu uma estrutura chamada árvore de cortes mínimos, representada por T_G. O corte mínimo entre qualquer par de vértices de G pode ser obtido através da inspeção do caminho entre esses vértices em T_G. A Figura 2.8 mostra um grafo e sua árvore de cortes mínimos. Observe que o menor peso dentre as arestas que pertencem ao caminho entre quaisquer pares de vértices é exatamente o corte mínimo entre esses vértices no grafo original.

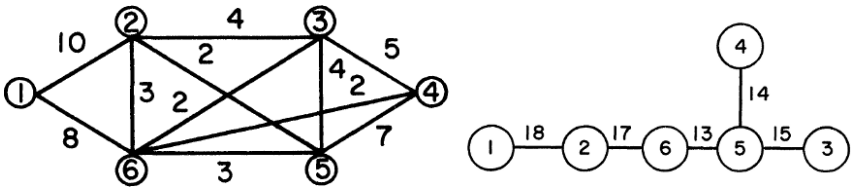

FIGURA 2.8 — Um grafo e sua árvore de cortes mínimos.

Segundo Gomory e Hu (1961), é possível construir uma árvore de cortes mínimos para qualquer grafo simples ou ponderado. O algoritmo para obtenção de uma árvore de cortes mínimos pode ser encontrado em Schwahn (2009).

2.9 Isomorfismo

Dois grafos G e H são denominados isomorfos quando existe uma função bijetora $f\colon V(G) \to V(H)$ chamada isomorfismo tal que

$$\forall \{u, v\} \in E(G), \{f(u), f(v)\} \in E(H).$$

2.10 Multigrafos

Um multigrafo G é uma tripla (V, E, ω), onde:

- V é um conjunto de vértices.
- E é um multiconjunto de pares não ordenados de vértices.
- $\omega\colon E \to \mathbb{R}$ é uma função peso.

Disso decorre que podem haver múltiplas arestas entre um mesmo par de vértices de G. É possível, ainda, a existência de arestas com extremos coincidentes, chamadas *loops*. A Figura 2.9 mostra um exemplo de multigrafo. Note que existem várias arestas paralelas entre a e b e uma aresta com extremos coincidentes b.

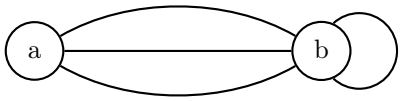

FIGURA 2.9 — Exemplo de multigrafo com loops.

3 CLUSTERIZAÇÃO EM GRAFOS

Muitas vezes, os dados que se deseja agrupar apresentam uma estrutura de grafos, como nos casos de redes sociais, circuitos elétricos e da rede citação em artigos. Em outros casos, é impossível ou impraticável calcular a similaridade entre todos os pares de elementos, mas é possível aferir o valor de uma similaridade indisponível através das similaridades desses elementos a um elemento em comum. Em ambas as situações, os dados podem ser modelados por um grafo cujos vértices correspondem aos elementos do conjunto de dados e as arestas indicam a similaridade ou a dissimilaridade entre os elementos.

Clusterização em grafos é um nome genérico para uma variedade de métodos matemáticos, estatísticos e de heurísticas que podem ser usados para reconhecer grupos naturais, também chamados de comunidades, clusteres naturais ou subgrafos coesivos. O processo de identificação dos clusteres naturais de um grafo é também denominado determinação da estrutura de comunidades do grafo.

A clusterização fornece uma abstração dos elementos para o cluster natural que os contém. Dessa forma, ajuda a descobrir a distribuição de padrões e correlações interessantes em grandes conjuntos de dados (HALKIDI; BATISTAKIS; VAZIRGIANNIS, 2001). No caso específico da clusterização em grafos, a clusterização facilita a interpretação de grafos com um conjunto grande de vértices, o que é útil na sumarização e compressão de dados, na geração e teste de hipóteses e na previsão de características de um elemento baseado nos elementos da comunidade que o contem.

Uma característica importante que ajuda a identificar se um método é de fato uma técnica de clusterização é se ele se baseia apenas em informações encontradas nos dados. Em outras palavras, não há clusteres naturais predefinidos nem definição do que seriam relações desejáveis entre os dados, por isso, a clusterização é

um processo não supervisionado (BERRY; LINOFF, 1997 apud HALKIDI; BATISTAKIS; VAZIRGIANNIS, 2001). Especificamente, a clusterização em grafos identifica a estrutura de comunidades de um grafo baseado apenas na estrutura das arestas.

A clusterização em grafos é a tarefa de agrupar os vértices em clusteres fortemente homogêneos e bem separados, isto é, elementos de uma mesma comunidade devem ser altamente similares, mas muito diferentes de elementos externos à comunidade. A ideia natural de clusterização em grafos é a separação em subgrafos densos esparsamente conectados. Note que, embora seja utilizado o termo subgrafo, uma comunidade na verdade seria o conjunto de vértices que induz esse subgrafo. As arestas que possuem extremos pertencentes a um mesmo cluster são denominadas arestas intra-cluster. As arestas com extremos situados em clusteres diferentes são ditas arestas inter-cluster.

A definição mais frouxa possível para uma comunidade é que ela seja um conjunto de vértices que induz um grafo conexo. A definição mais estrita é que seja uma clique maximal. Na maioria das vezes, os grupos naturais estão em algum lugar dentre esses extremos (SCHAEFFER, 2006). Como o grupo natural deve pelo menos induzir um grafo conexo, normalmente os algoritmos assumem que o grafo seja conexo. Nesses casos, para aplicar um método de clusterização em grafos em um grafo desconexo, basta aplicá-lo a cada componente conexa separadamente.

Essa intuição pode ser exemplificada pela Figura 3.1, que mostra um grafo com três clusteres candidatos a grupo natural destacados. As arestas intra-cluster também estão destacadas para facilitar a visualização dos subgrafos induzidos pelos clusteres elas. O cluster da esquerda é um forte candidato a grupo natural, pois é denso e tem poucas conexões com o restante do grafo. O cluster do meio também é relativamente denso, mas é muito conectado ao restante do grafo e o cluster da direita, apesar de possuir poucas arestas inter-cluster, tem densidade baixa. Por isso, os clusteres do meio e da direita não parecem ser bons candidatos a grupo natural.

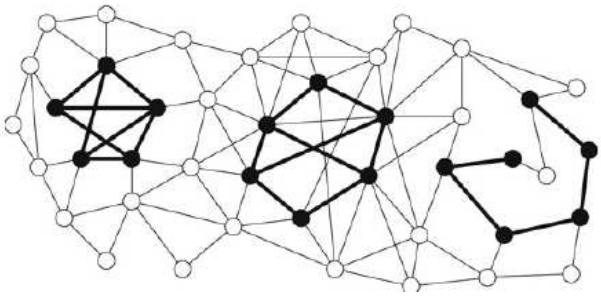

FIGURA 3.1 — Candidatos a grupo natural em um grafo. Imagem retirada de Schaeffer (2007).

Não existe uma definição de grupo natural universalmente aceita, mas em geral têm-se:

- Os clusteres devem ser conexos: deve haver pelo menos um e de preferência vários caminhos conectando cada par de vértices do cluster.
- Os caminhos devem ser internos ao cluster: o subgrafo induzido pelo cluster deve ser conexo.
- Os clusteres devem ser minimamente conectados: um passeio randômico tende a permanecer um longo período dentro de um mesmo cluster. Entretanto, o número de arestas entre comunidades não deve ser estritamente minimizado, já que comunidades maiores admitem mais arestas inter-cluster que comunidades pequenas (NEWMAN; GIRVAN, 2004).
- As arestas inter-cluster tendem a estar contidas em muitos dos caminhos mínimos entre os pares de vértices (DONGEN, 2000).
- A densidade interna do subgrafo induzido pelo cluster deve ser consideravelmente maior que a densidade do grafo (NEWMAN, 2004 apud SCHAEFFER, 2007).

A primeira vista, clusterização em grafos parece uma especificação de clusterização, mas é justamente o contrário. Qualquer problema de clusterização em que

uma medida de similaridade foi definida pode ser trivialmente reduzido a um problema de clusterização em grafos da seguinte forma (SCHAEFFER, 2006):

- Os elementos são representados por vértices.
- Uma aresta é inserida para cada par de elementos, ponderada por sua similaridade, ou por sua dissimilaridade.

A redução de um problema de clusterização em grafos para um problema de clusterização, por outro lado, só é possível quando o grafo é completo.

O restante do capítulo está dividido da seguinte forma:

- A Seção 3.1 discorre sobre alguns problemas com a informalidade da definição de clusterização em grafos.
- A Seção 3.2 define e classifica soluções de clusterização em grafos e apresenta algumas notações.
- A Seção 3.3 mostra algumas medidas de qualidade de clusteres isolados e de soluções de clusterização como um todo.
- As Seções 3.4 e 3.5 apresentam algumas medidas de centralidade de vértices e arestas.
- A Seção 3.6 mostra alguns algoritmos de clusterização em grafos.
- A Seção 3.7 expõe as discussões e considerações finais do capítulo.

3.1 Problemas com a informalidade da definição de clusterização em grafos

Existe uma variedade de formulações diferentes de grafo para o uso na clusterização em grafos. Alguns autores, como, por exemplo, Wu e Huberman (2004),

utilizam grafos simples e a presença de arestas indica uma similaridade entre os vértices e a ausência de arestas indica uma alta dissimilaridade entre eles. Outros autores utilizam grafos ponderados e o peso das arestas pode indicar tanto a similaridade entre os vértices quanto a dissimilaridade (SCHAEFFER, 2006). Outros ainda, como Bansal, Blum e Chawla (2002), usam pesos +1 para indicar uma alta similaridade e -1 para indicar uma alta dissimilaridade.

Essa diferença de notação muitas vezes causa confusão. Se o peso das arestas for definido como a similaridade, o número caminhos de comprimento grande em G é alto para os pares de vértices de um mesmo cluster e baixo para pares de vértices pertencentes a grupos diferentes. Se os pesos forem adotados como a dissimilaridade entre os vértices, entretanto, o objetivo é justamente o contrário.

Outro problema é a definição de comunidades, a maior parte dos autores, como Edachery et al. (1999), consideram que a comunidade é um grupo de vértices, entretanto, outros consideram que a comunidade é um subgrafo como, por exemplo, Hartuv e Shamir (2000). Esse texto assume a primeira definição.

3.2 Notação e definições

A clusterização em grafos determina a estrutura de comunidades do grafo. Como cada cluster natural é um subconjunto do conjunto de vértices, então um algoritmo de clusterização em grafos encontra o conjunto dos subconjuntos do conjunto de vértices que são comunidades.

DEFINIÇÃO 3.2: Uma solução de clusterização em grafos de um grafo G é um conjunto de clusteres $C = \{C_1, C_2, ..., C_n\}$, onde $\forall\ C_i \in C, C_i \subseteq V(G)$.

Impondo mais restrições à definição da solução de clusterização, é possível classificá-la em 3 eixos, referentes à sobreposição, à completude e à hierarquia (TAN; STEINBACH; KUMAR, 2005):

- **Exclusiva, sobreposta, difusa ou probabilística**

  Uma solução de clusterização é exclusiva se cada vértice pertencer a um e somente um cluster, isto é,

  $$\forall C_i, C_j \in C \wedge i \neq j, C_i \cap C_j = \emptyset,$$

  caso contrário, é dita sobreposta. Soluções de clusterização sobrepostas são usadas quando os vértices podem pertencer naturalmente a diversos grupos (quando os vértices representam objetos, por exemplo, um rádio-relógio pertence ao grupo de rádios e ao de relógios) ou quando o vértice está entre alguns grupos mas não pode ser razoavelmente atribuído a nenhum deles.

  Em uma solução de clusterização difusa, cada vértice pertence a todos os grupos, com um peso de adesão entre 0 (não pertence) e 1(pertence completamente). Em outras palavras, uma solução de clusterização é difusa se é um conjunto de conjuntos difusos. Uma solução de clusterização probabilística é uma solução difusa na qual a soma dos pesos de adesão de cada vértice é 1. O peso indica a probabilidade de o elemento pertencer ao cluster.

- **Completa ou parcial**

  Uma solução de clusterização é completa se todos os vértices do grafo pertencem a algum cluster, isto é, se $\bigcup_{C_i \in C} C_i = V(G)$. Caso contrário, é denominada parcial. Soluções de clusterização parciais são úteis quando há elementos que não possam ser razoavelmente atribuídos a nenhum dos clusteres.

- **Particional ou hierárquica**

A solução de clusterização particional é uma solução exclusiva em que não há qualquer relacionamento entre os clusteres. Tan, Steinbach e Kumar (2005) exige que, além de exclusiva, a solução seja completa, entretanto isso não é um consenso.

Uma solução de clusterização hierárquica é uma solução completa em que há um relacionamento hierárquico entre os clusteres. O nível mais alto da hierarquia contém apenas um cluster igual ao conjunto de vértices. Os clusteres dos níveis mais baixos são formados pela partição de um ou mais clusteres do nível imediatamente superior. Os clusteres do nível mais baixo da hierarquia são denominados folhas. Geralmente, as folhas são clusteres unitários, isto é, formados por um único vértice, mas não é necessário que o sejam.

Uma solução de clusterização hierárquica descreve uma hierarquia onde cada cluster, exceto pelas folhas, é formado pela união de subclusteres dos níveis inferiores. Isso pode ser representado por uma série de partições aninhadas ou por um diagrama de árvore chamado dendograma. A Figura 3.3 mostra como um (b) conjunto de partições aninhadas pode ser representada por um (a) dendograma.

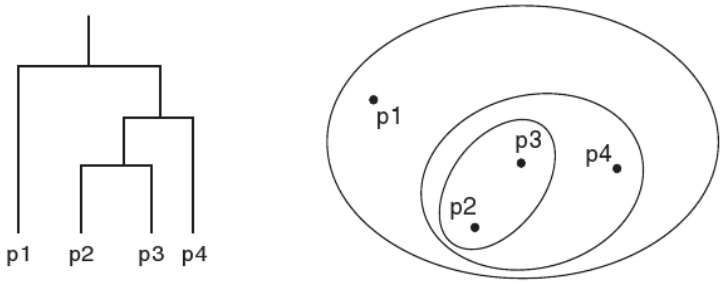

FIGURA 3.3 — (a) Dendograma e (b) conjunto de classes aninhadas referentes à solução de clusterização $\{\{p1, p2, p3, p4\}, \{p2, p3, p4\}, \{p2, p3\}\}$. Imagem retirada de Tan, Steinbach e Kumar (2005).

Note que solução de clusterização hierárquica pode ser vista, ainda, como a união de soluções particionais exclusivas completas, sendo que: (i) a primeira contém apenas um cluster contendo todos os elementos (ii) cada partição, exceto a primeira, é o resultado da divisão de um ou mais clusteres da partição

imediatamente anterior em sub-clusteres.

Soluções particionais podem ser obtidas do dendograma através de cortes horizontais em um nível específico. A Figura 3.4 mostra esse processo para um grafo com conjunto de vértices $\{1, ..., 23\}$. Observe as linhas tracejadas horizontais. Se o dendograma for cortado pela linha vermelha, a solução de clusterização apresentará os clusteres $\{1, ..., 14\}$ e $\{15, ..., 23\}$. Um corte na linha lilás, por outro lado, resulta nos clusteres $\{1, ..., 5\}$, $\{6, ..., 8\}$, $\{9\}$, $\{10, ..., 14\}$, $\{15, 16\}$, $\{17\}$, $\{18, ..., 20\}$ e $\{21, ..., 23\}$. Onde estão os clusteres mais relevantes em um dendograma? Grupos naturais menores e mais homogêneos tendem a permanecer inteiros até níveis baixos das folhas, enquanto grupos maiores e com maior diversidade tendem a ser separados em níveis mais altos. A escolha de "onde cortar" o dendograma é subjetiva, mas o mais comum é cortar o cluster em vários níveis de acordo com algum critério, como tamanho, densidade ou desvio padrão. Um exemplo de corte em vários níveis na Figura 3.4 seria a solução formada pelos clusteres $\{1, ..., 14\}$, $\{15, 16\}$, $\{17\}$, $\{18, ..., 20\}$ e $\{21, ..., 23\}$, cortando parte do dendograma na linha vermelha e parte na lilás.

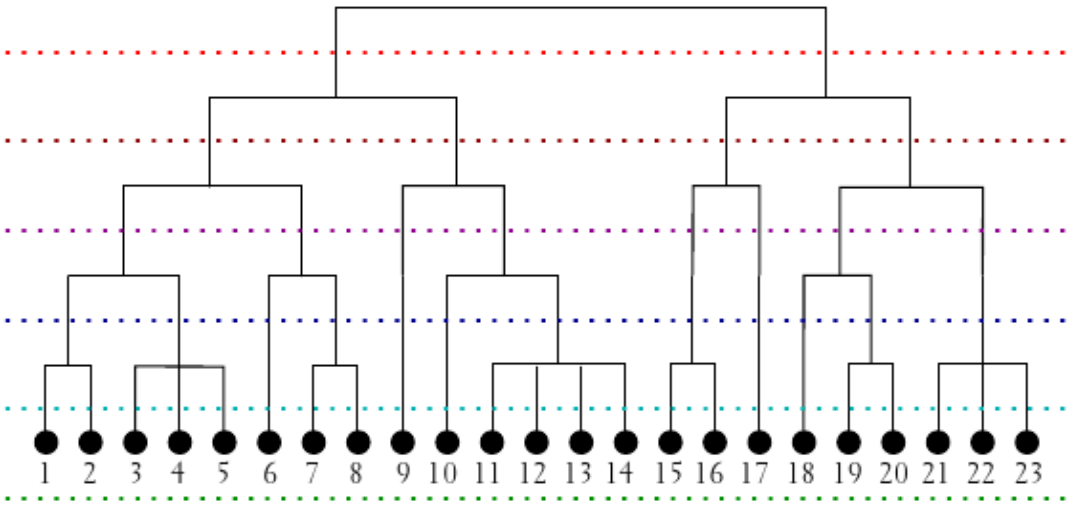

FIGURA 3.4 — Dendograma mostrando como obter soluções de clusterização particionais de uma solução hierárquica. Imagem retirada de Schaeffer (2006).

Neste trabalho, as soluções do problema de clusterização em grafos serão exclusivas e completas a não ser que seja dito expressamente o contrário. Ou seja, as soluções de clusterização são partições do conjunto de vértices do grafo. Ainda, como cada comunidade deve induzir um subgrafo conexo, então

$$\forall C_i \in C, G[C_i] \text{ é conexo.}$$

O conjunto de todas as soluções de clusterizações em grafos que são exclusivas, completas e que induzem subgrafos conexos no grafo G será denotado por $Clust(G)$. A solução de clusterização é formada por apenas um cluster contendo todos os vértices e a formada apenas por clusteres unitários são chamadas soluções triviais.

3.2.1 Propriedades de clusteres e de soluções de clusterização

As medidas de distância entre clusteres são importantes para avaliar se dois clusteres são bem separados. A distância entre os clusteres C_i e C_j é denotada por $dist(C_i, C_j)$, pode ser calculada como a maior, a menor ou a média das distâncias entre os vértices de C_i e C_j.

A distância entre vértices e clusteres pode ser utilizada para medir a tendência do vértice a pertencer a um cluster específico. A distância entre um vértice $v \notin C_j$ e um cluster C_j , $dist(v, C_j)$, é calculada como a distância máxima ou média entre esse vértice e os vértices do cluster.

Os graus interno e externo de um vértice em relação um cluster são utilizados para medir o quanto conectado um vértice é ao cluster em que está inserido e o quanto ele é conectado ao restante do grafo. O grau interno do vértice $v \in C_i$ em

relação ao cluster C_i, $d_{int}(v, C_i)$, é dado pela soma dos pesos das arestas intra-cluster com extremo em v. Analogamente, o grau externo de v em relação a C_i, $d_{ext}(v, C_i)$, é a soma dos pesos das arestas entre inter-cluster com extremo em v. Note que $d(v) = d_{int}(v, C_i) + d_{ext}(v, C_i)$.

Os conjuntos de arestas intra-cluster de um cluster C_i e de uma solução de clusterização C são dados por, respectivamente,

$$E(C_i) = \{\{v, w\} \in E | v, w \in C_i\}$$

e

$$E(C) = \bigcup_{C_i \in C} E(C_i).$$

Analogamente, os conjuntos de arestas inter-cluster de um cluster C_i e de uma solução de clusterização C são, respectivamente,

$$E'(C_i) = \{\{v, w\} \in E | v \in C_i \wedge w \notin C_i\}$$

e

$$E'(C) = \bigcup_{C_i \in C} E'(C_i).$$

Por fim, o conjunto das arestas inter-cluster com extremos em C_i e C_j é dado por

$$E(C_i, C_j) = \{\{v, w\} \in E : v \in C_i \wedge w \in C_j\}.$$

3.3 Medidas de Qualidade

A noção do que seria uma solução de clusterização em grafos natural ou, pelo menos, melhor dentre um conjunto de soluções de clusterização pode ser bem diversificada conforme o domínio do problema em questão e, até mesmo, conforme o pesquisador que o utiliza. Nesse sentido, a clusterização em grafos é análoga aos problemas de processamento de imagens, onde uma solução ótima muitas vezes é impossível, já que cada pesquisador tende a preferir uma imagem diferente - com mais ou menos contraste, mais ou menos cores.

Os algoritmos de clusterização em grafos encontram clusteres em qualquer que seja o grafo de entrada. Entretanto, isso só faz sentido em grafos com estrutura de comunidades. Se a estrutura do grafo é uniforme, a solução gerada será fortemente arbitrária (DONGEN, 2000). Considere por exemplo um grafo completo. A única divisão em grupos coesos e esparsamente conectados possíveis resulta em uma comunidade contendo todos os vértices.

É possível realizar uma avaliação dos dados a fim de aferir a tendência de agrupamento sem efetivamente realizar um processo de clusterização em grafos. Os dados são avaliados a fim de estimar se sua estrutura é aleatória ou se possui grupos naturais. Grafos que não apresentam estrutura de comunidades não deveriam ser processados. Entretanto, esse estudo nem sempre é realizado.

Mesmo em grafos que tenham uma estrutura de comunidades, alguns algoritmos podem encontrar soluções incorretas, devido a pressuposições incompatíveis com a sua estrutura. Por isso, é fundamental avaliar as soluções de clusterização a fim de validá-las, isto é, de decidir se os clusteres encontrados são significativos. Tan, Steinbach e Kumar (2005) define uma lista de questões importantes para a validação de uma solução.

- Determinar a tendência de agrupamento de um conjunto de dados, isto é, verificar se realmente existem grupos naturais.
- Determinar o número correto de clusteres.
- Avaliar os resultados de um algoritmo de clusterização.
- Comparar uma solução de clusterização com resultados conhecidos externamente.
- Comparar duas soluções de clusterização para determinar qual é a melhor.

As medidas de qualidade são tradicionalmente classificadas em três tipos a seguir (THEODORIDIS, 1999 apud HALKIDI; VAZIRGIANNIS, 2001) (TAN; STEINBACH; KUMAR, 2005):

- **Não supervisionadas ou interna**

  A qualidade de uma solução de clusterização pode ser medida como uma soma ponderada da qualidade dos clusteres. A qualidade de um cluster, por sua vez, baseia-se em medidas de coesão e de separação (BERRY; LINOFF, 1997 apud HALKIDI; BATISTAKIS; VAZIRGIANNIS, 2001). As medidas de coesão determinam o quanto o grafo induzido pelo cluster é conexo. As medidas de separação, por outro lado, caracteriza, o quanto o cluster é bem separado dos demais.

- **Supervisionadas ou externas**

  Medem o quanto a solução de clusterização corresponde a uma estrutura externa pré-definida. A ideia básica é testar se os dados são estruturados de forma não aleatória (TAN; STEINBACH; KUMAR, 2005), isto é, se realmente formam grupos naturais relevantes.

- **Relativas**

Comparam diferentes soluções de clusterização ou clusteres, a fim de identificar aquele que seja melhor de acordo com um critério pré-especificado.

Além de avaliar uma solução de clusterização, é possível aferir a qualidade de um cluster em específico, que pode ser usada para melhorar a qualidade global da solução. Um cluster não coeso, por exemplo, pode ser dividido, por outro lado, dois clusteres não muito separados podem ser unidos.

Diversas medidas de qualidade foram desenvolvidas para validação e aferição da qualidade de uma solução de clusterização em grafos. A escolha de quais medidas escolher para atestar a qualidade de uma solução de clusterização depende fundamentalmente do escopo da pesquisa. Os vários algoritmos de clusterização em grafos diferem justamente nessa definição do que significa a qualidade de uma solução, ou seja, em quais são os requerimentos de coesão e separação e como eles se combinam a fim de formar o critério de otimização dos algoritmos.

Apesar da variedade de medidas existentes, não se sabe precisamente o quão bem os índices medem a qualidade de uma solução de clusterização em grafos. Várias heurísticas e argumentos analíticos estão disponíveis, mas não existe qualquer teorema e mesmo a noção da estrutura de comunidade é em si baseada na metodologia escolhida para calculá-la (PORTER; ONNELA; MUCHA, 2009).

Segundo Porter, Onnela e Mucha (2009), ao analisar as redes construídas a partir de dados do mundo real, a melhor prática seria usar vários algoritmos disponíveis e confiar somente nas estruturas que são similares em vários métodos, a fim de ter certeza de que elas são propriedades dos dados reais, ao invés de produtos dos algoritmos utilizados para produzi-las.

O restante da seção descreve e compara algumas medidas de qualidade.

3.3.1 Medidas de coesão de um cluster

As medidas de coesão medem a conectividade interna de um cluster. Por isso, elas são aplicáveis apenas a clusteres isolados e não a soluções de clusterização como um todo. A densidade (BOUTIN; HASCOET, 2004) de um cluster C_i é calculada por

$$\delta(C_i) = \frac{|E(C_i)|}{|C_i|^2}.$$

Essa é a medida mais simples para aferir a qualidade de um cluster. Entretanto, dois grafos podem ter uma estrutura completamente diferente mesmo tendo o mesmo número de vértices e de arestas. Por isso, a densidade é uma medida de qualidade muito limitada. Observe na Figura 3.5 que os dois clusteres apresentam a mesma densidade embora sejam muito diferentes.

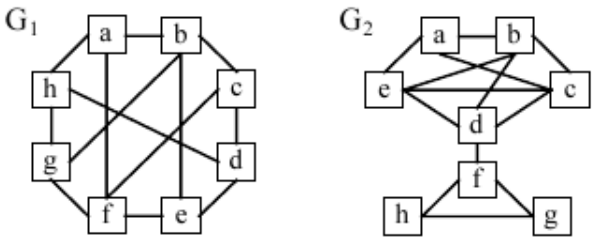

FIGURA 3.5 — Os dois grafos apresentam estruturas completamente diferentes, no entanto, a densidade do cluster $V(G)$ é a mesma nos dois grafos. Imagem retirada de Boutin e Hascoet (2004).

A compacidade (BOTAFOGO; RIVLIN; SHNEIDERMAN, 1992) foi desenvolvida para medir a coesão de uma comunidade, considerando a forma como as arestas estão distribuídas no subgrafo induzido. A compacidade Cp de um cluster C_i é dada por

$$Cp(C_i) = \frac{Max - \sum_{\substack{u,v \in C_i, \\ u \neq v}} dist(u,v)}{Max - Min}.$$

Max e Min são os valores que $\sum_{\substack{u,v \in C_i, \\ u \neq v}} dist(u,v)$ assumiria se todos os pares de vértices estivessem conectados, respectivamente, pela maior e pela menor distância no cluster, isto é

$$Max = \frac{|C_i|(|C_i|-1)Diam(G[C_i])}{2}$$

e

$$Min = \frac{|C_i|(|C_i|-1)}{2}.$$

A ideia dessa medida é que quanto mais compacto ou coeso um subgrafo induzido pelo cluster, menores as distâncias entre os pares de vértices. Nesse sentido, uma clique é o cluster mais compacto possível e um conjunto de vértices que induza um grafo desconexo não é considerado compacto.

Considere uma clique, então

$$\sum_{\substack{u,v \in C_i, \\ u \neq v}} dist(u,v) = Max = Min = \frac{|C_i|(|C_i|-1)}{2}.$$

Logo, a compacidade de uma clique é 1. Por outro lado, a distância entre dois vértices não alcançáveis pode ser mapeada por um valor inteiro muito superior à maior distância real entre dois vértices. Quanto maior o valor utilizado, mais aproximados os valores de Max e de $\sum_{\substack{u,v \in C_i, \\ u \neq v}} dist(u,v)$. Por isso, a compacidade de um conjunto de vértices que induz um subgrafo desconexo tende a 0. Os valores de compacidade variam entre 0 e 1.

Boutin e Hascoet (2004) propuseram um novo índice de compacidade normalizado que usa uma medida de similaridade. A similaridade entre dois vértices u e v, $sim(u,v)$ é dada pelo inverso da distância entre os vértices. Convenciona-se dizer que a similaridade entre dois vértices em componentes desconexas diferentes é 0. Por isso, não é necessário usar um valor alto arbitrário de distância para vértices não

alcançáveis.

A compacidade Cp^* de um cluster C_i é

$$Cp^*(C_i) = \frac{\sum_{\substack{u,v \in C_i, \\ u \neq v}} sim(u,v)}{\frac{|C_i|(|C_i|-1)}{2}}.$$

Com essa nova formulação, os valores de compacidade ainda variam de 0 a 1. Com $Cp^*(C_i) = 1$ para cliques e $Cp^*(C_i) = 0$ para conjuntos independentes.

Mesmo as medidas de compacidade ainda não são boas medidas de qualidade se usadas sozinhas, pois consideram apenas a coesão e não a separação. Considere por exemplo um grafo G completo com 10 vértices e um conjunto S de 3 vértices desse grafo. Como o grafo é completo, o subgrafo induzido $G[S]$ também o será. Logo, a densidade, a compacidade e a compacidade normalizada de S possuirão valor máximo. Entretanto, $|E(S)| = 3$ e $|E'(S)| = 21$. Assim, S é denso e coeso mas fortemente conectado ao restante do grafo, o que o torna um candidato muito ruim a ser uma comunidade em G.

3.3.2 Medidas baseadas em coesão e separação

Os índices baseados em coesão e separação são mais robustos que os índices baseados apenas na coesão. Além disso, eles podem ser utilizados para medir a qualidade de uma solução de clusterização como um todo.

3.3.2.1 Índices baseados em distâncias

O índice de Dunn (DUNN, 1974) objetiva identificar uma solução de clusterização em grafos compacta e bem separada. O índice de Dunn de uma solução C é definido como

$$D(C) = \frac{\min_{C_i, C_j \in C, \atop C_i \neq C_j} dist(C_i, C_j)}{\max_{C_k \in C} Diam(G[C_k])},$$

onde $dist(C_i, C_j)$ e $Diam(C_k)$ representam, respectivamente a menor distância inter-cluster e a maior distância intra-cluster.

Quanto maior a menor distância inter-cluster na solução e quanto menor a maior distância intra-cluster, maior o valor de D. Por isso, valores altos de D correspondem a boas soluções.

O índice de Dunn, além de requerer muito tempo em seu cálculo, é muito sensível à presença de ruído, já que isso pode acarretar num aumento do diâmetro (HALKIDI; BATISTAKIS; VAZIRGIANNIS, 2001). Isso se deve principalmente ao fato do índice de Dunn depender somente de uns poucos clusteres e ligações entre eles (BOUTIN; HASCOET, 2004).

O índice de Davies Bouldin (DAVIES; BOULDIN, 1979) visa identificar soluções que são compactas e bem separadas. O índice de Davies Bouldin para uma solução C é

$$DB(C) = \frac{1}{|C|} \sum_{C_j \in C} \max_{\substack{C_i \in C \\ C_i \neq C_j}} \frac{Diam(C_i) + Diam(C_j)}{dist(C_i, C_j)},$$

onde $\frac{Diam(C_i)+Diam(C_j)}{dist(C_i,C_j)}$ é uma medida de similaridade entre os clusteres C_i e C_j que assume valores menores quanto mais coesos e bem separados forem os clusteres.

Assim, o índice de Davies Bouldin é a média da similaridade de cada cluster com seu cluster mais similar. Portanto, quanto menor o valor do índice de Davies

Bouldin, melhor a solução.

O índice de Davies Bouldin é mais robusto que o índice de Dunn por considerar mais clusteres em seu cálculo e, com isso, é menos sensível a ruídos. Um problema comum aos índices de Dunn e Davies Bouldin, entretanto, é que eles consideram apenas as distâncias entre os vértices, mas não fazem nenhuma averiguação acerca das conectividades intra-cluster e inter-cluster.

3.3.2.2 Índices baseados na conectividade inter-cluster e intra-cluster

A densidade relativa (MIHAIL et al., 2002) de um cluster C_i é dada por

$$\rho(C_i) = \frac{|E(C_i)|}{|E(C_i)| + |E'(C_i)|}.$$

Isto é, a densidade relativa é a razão entre as arestas intra-cluster e o número total de arestas incidentes no cluster. Se o cluster for unitário ou um conjunto independente, é convencionado que $\rho(C_i) = 0$.

A cobertura (BRANDES; GAERTLER; WAGNER, 2003) é semelhante à densidade relativa, mas aplicável a uma solução. A cobertura de uma solução de clusterização C é dada por

$$cobertura(C) = \frac{|E(C)|}{|E(C)| + |E'(C)|} = \frac{|E(C)|}{|E(G)|}.$$

Quanto maior o valor de cobertura, melhor a qualidade de um agrupamento, já que mais arestas são internas ao cluster. No entanto, a simples maximização da cobertura forma uma solução trivial contendo todos os vértices do grafo. Por outro lado, se existe a restrição de haver pelo menos dois clusteres, o grafo seria dividido pelo corte mínimo.

Observe na Figura 3.6 que a divisão do grafo à esquerda no seu corte mínimo produz uma solução com qualidade maior que o original. Entretanto, se o grafo a direita for dividido no seu corte mínimo, a qualidade da solução piora. Isso ocorre pelo corte no grafo à esquerda criar grupos de tamanhos relativamente parecidos, enquanto o corte no segundo grafo levaria à criação de um grupo muito pequeno.

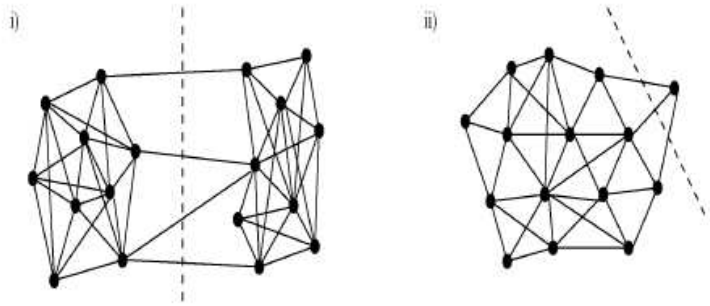

FIGURA 3.6 — Cortes mínimos em dois grafos diferentes. Imagem retirada de Kannan, Vempala e Veta (2000).

A expansão foi introduzida por Kannan, Vempala e Veta (2000) e se baseia justamente nessa ideia. Ao invés de medir simplesmente o corte mínimo, a expansão mede o tamanho relativo do corte aos grupos que cria.

A expansão de um corte definido por S no grafo G é dada por

$$\psi(S) = \frac{\sum_{\substack{u \in S, \\ v \in V(G) \setminus S}} \omega_G(\{u, v\})}{\min\{|S|, |V(G) \setminus S|\}}.$$

A expansão de um corte pode ser utilizada para caracterizar a qualidade de um grafo, de um cluster e de uma solução de clusterização da seguinte forma:

- A expansão do grafo G, $\psi(G)$, é dada pela menor expansão dentre os cortes em G.

- A expansão de um cluster C_i é calculada como a expansão do subgrafo induzido por C_i, isto é, $\psi(C_i) = \psi(G[C_i])$.

- A expansão de uma solução C, $\psi(C)$, é a menor expansão entre os clusteres de C.

Por isso, a expansão de uma solução serve como um limite inferior para a expansão de qualquer corte intra-cluster. Decorre que, soluções com expansão alta apresentam clusteres bem separados.

O problema da expansão é que ela á a mesma importância a todos os vértices, independente de como eles estão conectados ao grafo. Um vértice que tenha similaridade muito baixa com todos os outros vértices, por exemplo, ao ser inserido em qualquer grupo diminuiria muito a expansão mínima. Vértices que têm muitos vizinhos similares, por outro lado, deveriam ter importância maior que vértices com poucos vizinhos similares.

A condutância é uma medida introduzida por Kannan, Vempala e Veta (2000) que generaliza a expansão. Ao invés dos cortes serem ponderados inversamente pelo número de vértices eles são ponderados por uma função do peso das arestas no grupos, refletindo a importância dos vértices.

A condutância de um corte definido por S no grafo G é dada por

$$\phi(S) = \frac{\sum_{\substack{u \in S, \\ v \in V(G) \backslash S}} \omega_G(\{u, v\})}{\min\{\omega(S, V(G)), \omega(V(G) \backslash S, V(G))\}}.$$

Assim como na expansão, a condutância de um grafo G, $\phi(G)$, é dada pela menor condutância dentre os cortes em G.

Considere agora um cluster C_i em G. A condutância de um corte definido por S em C_i é dada por

$$\phi(S, C_i) = \frac{\sum_{\substack{u \in S, \\ v \in C_i \backslash S}} \omega(\{u, v\})}{\min\{c(S, V(G)), c(C_i \backslash S, V(G))\}},$$

Note que, diferentemente da expansão, a condutância de um corte definido por S em C_i é diferente da condutância de um corte definido por S no grafo induzido por C_i. Isso ocorre porque os denominadores são diferentes.

Quanto menor a condutância do corte, mais vantajoso é dividir o cluster por esse corte. A condutância de um cluster C_i, $\phi(C_i)$, é a menor condutância entre os cortes em C_i. Da mesma forma, a condutância de uma solução C, $\phi(C)$, é igual à menor condutância dentre todos os clusteres da solução. A condutância de uma solução, assim como a expansão, age como um limite inferior da condutância de qualquer corte intra-cluster. Soluções com condutância alta apresentam clusteres bem separados.

Tanto a expansão quanto condutância parecem ser medidas muito boas de qualidade para os agrupamentos, no entanto, são insuficientes como critério de agrupamento, já que não consideram o peso inter-cluster, nem o tamanho relativo dos clusteres (FLAKE; TARJAN; TSIOUTSIOULIKLIS, 2004).

Outro problema segundo Kannan, Vempala e Veta (2000) é que a estrutura de comunidades de uma rede pode consistir de diversas comunidades de alta qualidade e poucas comunidades de qualidade muito baixa, de modo que a solução tem uma qualidade global baixa. Medidas como a expansão e a condutância incentivam a criação de vários clusteres de qualidade relativamente baixa, já que a qualidade global da solução é mais alta.

A medida (α, ϵ) é uma medida bi-critério introduzida por Kannan, Vempala e Veta (2000) para contornar esse problema. Uma solução de clusterização C é dita (α, ϵ) se $\phi(C) \geqslant \alpha$ e $\frac{|E'(C)|}{|E(C)|+|E'(C)|} \leqslant \epsilon$. Alternativamente, pode-se utilizar a expansão ao invés de utilizar a condutância.

Note que α é um limite inferior para qualidade dos clusteres e ϵ é um limite superior para a fração do peso das arestas do grafo que são inter-cluster. Resultados empíricos sugerem que essa medida é adequada a uma variedade de aplicações (KANNAN; VEMPALA; VETA, 2000).

3.3.2.3 Índices baseados na comparação com modelos ideais ou aleatórios

A performance (BRANDES; GAERTLER; WAGNER, 2003) é a fração de pares de vértices corretamente interpretados em uma solução. Um par de vértices pertencentes a um mesmo cluster é corretamente interpretado se existe uma aresta entre eles. Por outro lado, um par de vértices pertencente a clusteres diferentes é corretamente interpretado se não há aresta entre eles.

A performance de uma solução C de um grafo G é dada por

$$performance(C) = \frac{E(C) + |\{\{u,v\}|C_i \neq C_j \wedge u \in C_i \wedge v \in C_j \wedge \{u,v\} \notin E(G)\}|}{\frac{|V(G)|(|V(G)|-1)}{2}}.$$

O fator de escala $\frac{|V(G)|(|V(G)|-1)}{2}$ garante que os valores de performance variem entre 0 e 1. Analogamente a performance pode ser calculada contando o número de vértices incorretamente interpretados, da seguinte forma

$$\begin{aligned} performance(C) &= 1 - \frac{E'(C) + \sum_{C_i \in C} \frac{|C_i|(|C_i|-1)}{2} - E(C_i)}{\frac{|V(G)|(|V(G)|-1)}{2}} \\ &= 1 - \frac{2|E(G)|(1 - 2cobertura(C)) + \sum_{C_i \in C} |C_i|(|C_i|-1)}{|V(G)|(|V(G)|-1)}. \end{aligned}$$

Os grafos cujas componentes conexas são cliques são os únicos cuja performance é máxima e é exatamente isso o que sugere a intuição (DONGEN; DONGEN, 2000). Pode-se considerar que a performance mede o quanto uma solução se afasta dessa solução ideal.

A modularidade, por outro lado, mede o quanto uma solução é melhor que um

grafo com as arestas distribuídas aleatoriamente.

Considere o grafo G. Como G tem $|E(G)|$ arestas, o número total de extremos de arestas em G é $2|E(G)|$. Considere todos os extremos de arestas que pertencem a um cluster C_i e cujo outro extremo pertence a C_j. A fração dos extremos de arestas que estão entre os clusteres C_i e C_j e que pertencem a C_i pode ser calculada por

$$e(C_i, C_j) = \begin{cases} \dfrac{|E(C_i, C_j)|}{2|E(G)|}, & C_i \neq C_j \\ \dfrac{|E(C_i)|}{|E(G)|}, & \text{caso contrário.} \end{cases}$$

Logo, a fração de extremos de arestas que pertencem a C_i pode ser obtida como

$$\begin{aligned} b_i &= \sum_{C_j \in C} e(C_i, C_j) \\ &= \frac{\sum_{v \in C_i} d(v)}{2|E(G)|}. \end{aligned}$$

Suponha, agora, que seja construído um novo grafo com o mesmo grupo de vértices mas que as extremidades das arestas sejam ligadas de forma aleatória. Note que os graus dos vértices no grafo original são respeitados no novo grafo. Nesse grafo, a fração das arestas intra-cluster no cluster C_i é b_i^2.

A modularidade mede a diferença entre a fração das arestas intra-cluster de uma solução e o valor esperado da fração das arestas intra-cluster em um grafo aleatório. A modularidade de uma solução C é dada por

$$
\begin{aligned}
Q(C) &= \sum_{C_i \in C} e(C_i, C_i) - b_i^2 \\
&= \frac{|E(C)|}{|E(G)|} - \sum_{C_i \in C} b_i^2 \\
&= cobertura(C) - \sum_{C_i \in C} b_i^2.
\end{aligned}
$$

Se C possui apenas um cluster, então $Q(C) = 0$, outros valores diferentes de 0 indicam desvios de aleatoriedade (NEWMAN, 2003a). Soluções com $Q(C)$ próximo de 1, indicam uma estrutura forte de comunidade. Os valores de $Q(C)$ para as redes reais geralmente estão na faixa de 0,3 a 0,7, valores mais elevados são raros (NEWMAN; GIRVAN, 2004).

A modularidade sofre de um limite de resolução (FORTUNATO; BARTHéLEMY, 2007 apud PORTER; ONNELA; MUCHA, 2009). Comunidades muito pequenas tendem a ser incorporadas em clusteres maiores, perdendo estruturas importantes.

3.3.3 Exemplo comparativo

A Figura 3.7 apresenta várias soluções para um mesmo grafo. Como o grafo é pequeno, é possível fazer uma avaliação visual das soluções. C_1 é muito ruim, pois é formada de um único cluster com conectividade intra-cluster baixa. C_2 é um pouco melhor, mas o cluster cujos vértices são representados por quadrados ainda apresenta baixa coesão. A solução C_3 é a melhor, pois apresenta clusteres relativamente coesos e bem separados. C_4, entretanto, parece piorar a solução, já que não apresenta um

aumento relevante na coesão, mas aumenta significativamente a conexão inter-cluster. A Tabela 3.8 apresenta os valores de algumas das medidas de qualidade para as soluções da Figura 3.7.

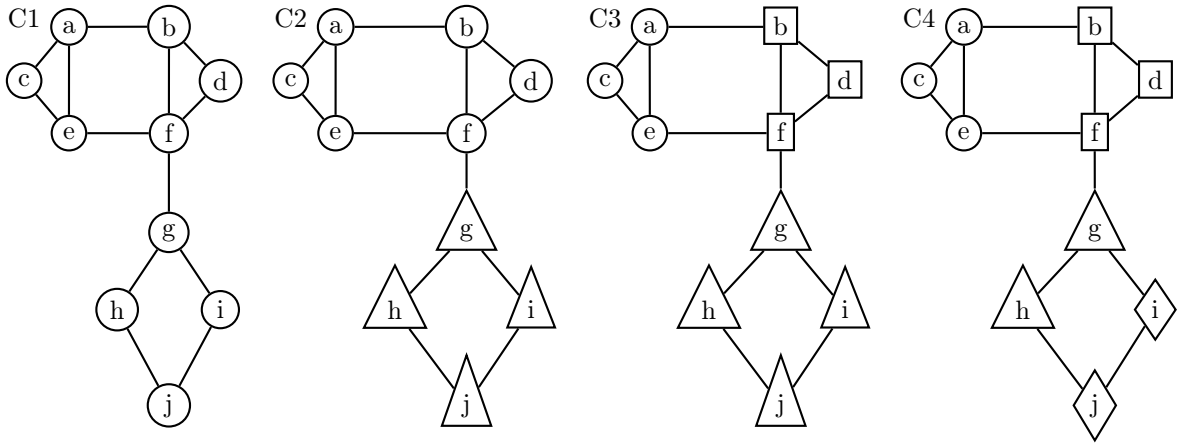

FIGURA 3.7 — Diferentes soluções de clusterização em grafos para um mesmo grafo. Cada cluster é representado nos grafos por uma figura geométrica diferente.

	Dunn	Davies Bouldin	Cobertura	Expansão	Condutância	Performance	Modularidade
C_1	0	—	1	$\frac{1}{4} = 0,25$	$\frac{1}{5} = 0,2$	$\frac{26}{90} \approx 0,29$	0
C_2	$\frac{1}{3} \approx 0,33$	5	$\frac{12}{13} \approx 0,92$	$\frac{2}{3} \approx 0,67$	$\frac{2}{5} = 0,4$	$\frac{70}{90} \approx 0,78$	$\frac{254}{676} \approx 0,38$
C_3	$\frac{1}{2} = 0,5$	$\frac{8}{3} \approx 2,67$	$\frac{10}{13} \approx 0,77$	1	$\frac{2}{3} \approx 0,67$	$\frac{80}{90} \approx 0,89$	$\frac{294}{676} \approx 0,43$
C_4	1	2	$\frac{8}{13} \approx 0,62$	1	$\frac{1}{2} = 0,5$	$\frac{80}{90} \approx 0,89$	$\frac{230}{676} \approx 0,34$

QUADRO 3.8 — Medidas de qualidade para as soluções da Figura 3.7.

Os índices de Dunn e Davies Bouldin e a Cobertura são muito limitados, uma vez que são baseados em poucas características. Por isso, não são capazes de identificar corretamente C_3 como a melhor solução. A expansão e a performance não conseguem identificar se a melhor solução é C_3 ou C_4, isso ocorre porque a expansão leva em conta apenas a conectividade interna do cluster e a performance pesa igualmente a falta de uma aresta dentro de um cluster como a presença de uma aresta inter-cluster, quando uma aresta inter-cluster deveria ter um peso um pouco maior. A condutância e a modularidade, por sua vez, identificaram corretamente a melhor solução.

3.4 Centralidade de vértices em uma rede

Bavelas (apud FREEMAN, 1977) foi o primeiro a discutir a existência de uma relação entre a posição de um vértice em uma rede e sua influência no grupo. Vértices localizados em posições centrais, seriam intermediários da comunicação entre vários pares de vértices e, por isso, teriam um potencial maior de influenciar o grupo.

Essa influência foi descrita de diversas formas. Segundo Bavelas (apud FREEMAN, 1977), vértices centrais podem reter ou distorcer a informação que passa por eles. Shaw (apud FREEMAN, 1977) fala do poder do intermediário de uma comunicação em se recusar a passar *requests* de informação. Para Shimbel (apud FREEMAN, 1977) vértices centrais teriam uma responsabilidade na manutenção da comunicação, por isso, a centralidade seria uma medida da pressão pela qual o vértice passa. Cohn e Marriott (apud FREEMAN, 1977) falam do potencial dos vértices centrais em coordenar as conectividades de outros pontos. A centralidade está relacionada, ainda, com a eficiência do grupo na resolução de problemas, percepção da liderança e da satisfação pessoal dos participantes (FREEMAN, 1979).

A centralidade pode ser interpretada, então, como a proeminência do vértice em uma estrutura social (BRANDES; GAERTLER; WAGNER, 2003). E os vértices mais centrais, dos quais os outros vértices dependem, são vistos como os atores mais poderosos do sistema (MARSDEN; LAUMANN, 1977 apud COOK; EMERSON; GILLMORE, 1983).

Segundo Freeman (1979), o vértice central de uma estrela é estruturalmente mais central que qualquer vértice em qualquer rede do mesmo tamanho. O problema é, no entanto, determinar a maneira em que esta posição é estruturalmente única. Freeman identificou três propriedades estruturais do centro de uma estrela: possui o maior grau possível, pertence aos caminhos mínimos entre todos os pares de vértices

e está localizado a uma distância mínima de todos os outros pontos. A centralidade de um vértice pode ser medida através do quanto ele se aproxima do centro de uma estrela em qualquer uma dessas propriedades. A Figura 3.9 mostra uma estrela de tamanho 4. Observe que o vértice b é o mais central segundo todas as propriedades.

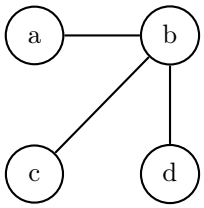

FIGURA 3.9 — O vértice localizado no centro da estrela possui o maior grau, está a uma distância mínima de todos os outros e está no número máximo de caminhos mínimos entre outros pares de vértices.

As seguintes medidas de centralidade são definidas para grafos não ponderados, embora a ideia possa ser estendida para grafos ponderados de forma simples.

3.4.1 Medidas de centralidade baseadas no grau

A ideia de usar o grau como índice de centralidade de um vértice foi introduzida por Shaw (apud FREEMAN, 1977). Um vértice com grau alto é visto como um grande canal de informação, por outro lado, um vértice de grau baixo é periférico e não tem uma participação muito ativa no processo de comunicação (FREEMAN, 1979).

Observe, novamente, a Figura 3.9. O grau do vértice b é 3. Se considerarmos qualquer rede com 4 vértices, um vértice nessa rede pode possuir no máximo uma aresta conectando a cada um dos outros vértices. Logo, o centro de uma estrela apresenta um grau de maior valor possível dentre todos os vértices de uma rede do mesmo tamanho.

Nieminen (apud FREEMAN, 1979) propôs uma medida simples de centra-

lidade baseada em grau, na qual a centralidade de um vértice v pode ser medida simplesmente por

$$C_D(v) = d(v).$$

Como o grau máximo de um vértice em uma rede com n vértice é $n-1$, C_D pode ser modificada para que seus valores variem entre 0 e 1 fazendo

$$C'_D(v) = \frac{C_D(v)}{n-1} = \frac{d(v)}{n-1}.$$

Segundo Cook, Emerson e Gillmore (1983), a maior fraqueza conceitual das medidas baseadas em grau é que elas são altamente localizadas, levando em consideração apenas ligações diretas, mas não efeitos indiretos ou caminhos.

3.4.2 Medidas de centralidade baseadas em proximidade

As medidas baseadas em proximidade medem o quanto um ponto particular está perto de todos os outros pontos (COOK; EMERSON; GILLMORE, 1983). Nesse sentido, um ponto central pode evitar o controle dos outros, não sendo dependente de intermediários na comunicação (LEAVITT, 1951 apud FREEMAN, 1979).

Na Figura 3.9, o vértice b está a uma distância unitária de cada um dos outros vértices. Como o caminho mínimo entre quaisquer vértices contém pelo menos uma aresta, qualquer vértice estará a pelo menos uma distância unitária de todos os outros vértices do grafo. Logo, o centro de uma estrela tem a maior proximidade a todos os outros vértices dentre todos os vértices de uma rede do mesmo tamanho.

Uma medida simples foi proposta por Sabidussi (apud FREEMAN, 1979), segundo a qual a centralidade do vértice v em um grafo G pode ser medida como

$$C_C^{-1}(v) = \sum_{u \in V(G)-v} dist(u, v).$$

Vértices com valores de C_C^{-1} menores são mais centrais. Em grafos desconexos, cada vértice está a uma distância infinita de pelo menos outro vértice, logo essa medida não é particularmente útil.

Beauchamp (1965) propôs uma medida normalizada cujos valores variam entre 0 e 1. A centralidade de um vértice v em um grafo G é

$$C'_C(v) = \frac{|V(G)| - 1}{\sum_{u \in V(G)-v} dist(u, v)}.$$

3.4.3 Medidas de centralidade baseadas em *betweenness*

As medidas de *betweenness* são baseadas na frequência em que um vértice pertence ao caminho entre os outros vértices do grafo. Quando há apenas um caminho entre um par de vértices, qualquer vértice interno a esse caminho consegue controlar completamente a comunicação, quando existe mais de um caminho mínimo, entretanto, um vértice interno a somente alguns desses caminhos tem um potencial de controle limitado (FREEMAN, 1977).

A Figura 3.10 ilustra essa limitação. Observe que qualquer caminho mínimo entre os vértices a e e possui o vértice d. Note que os vértices b e c, entretanto, são internos a apenas um desses caminhos. Por isso, d pode controlar toda a comunicação entre a e e, enquanto b e c só controlam as mensagens que trafegam pelo caminho ao qual eles são internos.

Na Figura 3.9, o vértice b é interno aos caminhos entre todos os outros vértices do grafo. Como o número de pares de vértices é fixo para grafos de mesmo tamanho,

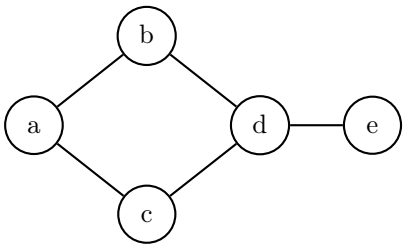

FIGURA 3.10 — O vértice d controla completamente a comunicação entre os vértices a e e, os vértices b e c, por outro lado, possuem apenas um controle parcial.

o centro de uma estrela tem o maior *betweenness* dentre todos os vértices de uma rede do mesmo tamanho.

Medidas diretas de *betweenness* foram desenvolvidas por Anthonisse (1971), com o nome de *rush*, e Freeman (1977). Sejam u, v e w três vértices de um grafo e $\sigma(u,v)$ o número de caminhos mínimos entre u e v. Supondo que uma mensagem sempre trafegue por caminhos mínimos e que dois vértices são indiferentes a qual caminho é utilizado na comunicação, a probabilidade que uma mensagem trafegue por um caminho em particular é $\frac{1}{\sigma(u,v)}$.

Denote por $\sigma(u,v,w)$ o número de caminhos mínimos entre u e v que contém w. O potencial que o vértice w controlar a comunicação entre u e v é o *betweenness* de w para u e v, que é calculado pela probabilidade que w seja interno a um caminho aleatoriamente selecionado entre u e v como

$$b(u,v,w) = \frac{\sigma(u,v,w)}{\sigma(u,v)}.$$

A centralidade global de w é a soma dos valores de *betweenness* de w para todos os pares de outros vértices no grafo e é dada por

$$C_B(w) = \sum_{\substack{u,v \in V(G)-w \\ u \neq v}} b(u,v,w).$$

Valores altos de C_B indicam vértices centrais. Segundo Freeman (1977), o valor máximo de C_B em uma rede de tamanho n é $\frac{n^2-3n+2}{2}$. Portanto, C_B pode ser

alterado de forma que seus valores variem dentre 0 e 1 fazendo

$$C'_B(w) = \frac{2C_B(w)}{n^2 - 3n + 2}.$$

Na maioria das redes, a comunicação não é feita apenas pelos caminhos mínimos (STEPHENSON; ZELEN, 1989) (FREEMAN; BORGATTI; WHITE, 1991), mas tendem a ser transmitida de uma forma mais aleatória. Observe a Figura 3.11 (a), todos os caminhos mínimos entre as comunidades tendem a passar pelos vértices A e B, por isso esses vértices terão valores de C_B altos e o vértice C um valor baixo. No entanto, nada impede que uma parcela considerável dos passeios aleatórios entre vértices dessas comunidades passe por C.

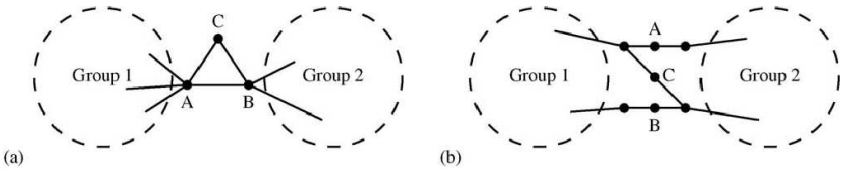

FIGURA 3.11 — (a) Os vértices A e B têm valores de *betweenness* altos, já o vértice C, não. (b) Os vértices A e B têm valores altos de *flow betweenness*, o vértice C, não. Imagem retirada de Newman (2003b).

Uma definição alternativa de *betweenness*, conhecida como *flow betweenness*, foi desenvolvida por Freeman, Borgatti e White (1991). Considere os vértices s, t e v. O *flow betweenness* de v em relação a s e t é o fluxo em v quando o maior fluxo possível é transmitido entre s e t. O *flow betweenness* global de um vértice v é a média dos *flow betweenness* de v em relação a todos os pares de vértices em $V(G)\backslash\{v\}$.

Entretanto, *flow betweenness* também apresenta alguns problemas. Considere a Figura 3.11 (b), o máximo fluxo entre os grupos é limitado em duas unidades, uma passando pelo vértice A e outra pelo B. O vértice C terá um *flow betweenness* baixo, mesmo que os caminhos através de C possam ser menores que os passando por A ou B.

Newman (2003b) considera que uma informação na rede é transmitida de forma completamente aleatória até a mensagem encontrar o destino. O *random-walk betweenness* é obtido calculando-se o número esperado de vezes que um passeio aleatório entre um determinado par de vértices passa por cada vértice e tirando a média entre as esperanças de todos os pares de vértices.

Essa medida inclui a contribuição de vários caminhos que não são ótimos em nenhum sentido, embora, caminhos menores contribuam mais, já que é improvável que um passeio aleatório fique muito longo antes de encontrar o destino. São incluídos alguns artifícios para impedir que um vértice obtenha um valor alto construindo um passeio aleatório que passe diversas vezes pelo vértice. Por exemplo, se um passeio qualquer passa por um vértice e depois passa novamente na direção oposta, não há nenhuma contribuição no *random-walk betweenness*.

A Figura 3.12 mostra uma rede de contatos sexuais. Quanto maior a representação do vértice, maior o seu valor de *random-walk betweenness*. Observe que os vértices mais periféricos possuem valores menores que os centrais, conforme o esperado. Os vértices destacados apresentam uma grande diferença entre seus valores de *betweenness* e *random-walk betweenness*. Não há nenhuma razão para supor que as doenças sexualmente transmissíveis se propagam nos caminhos mínimos (NEWMAN, 2003b). Assim, a utilização de uma medida de *betweenness* nesse caso, não consideraria a real importância desses vértices.

3.4.4 Exemplo ilustrativo das medidas de centralidade para vértices

A seguir será apresentado um exemplo ilustrativo das medidas de centralidade para vértices baseado em um exemplo retirado de Freeman (1979). A Figura 3.13

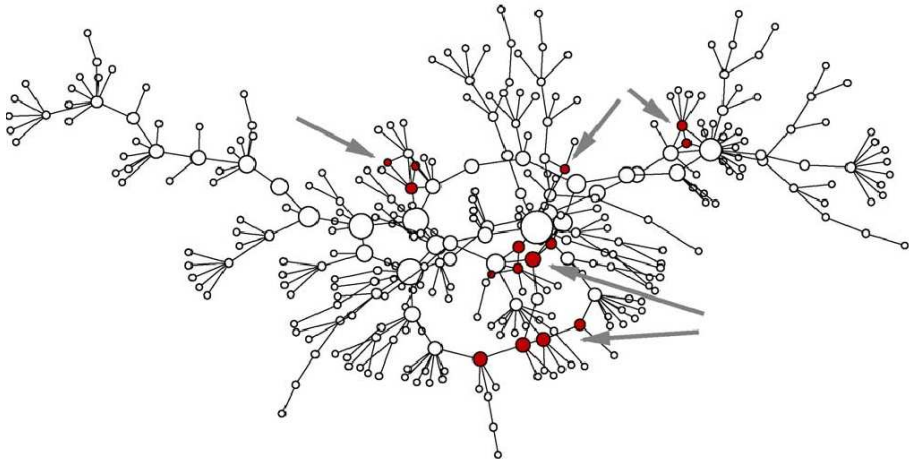

FIGURA 3.12 — Rede de contatos sexuais. Imagem retirada de Newman (2003b).

mostra todos os grafos não-isomorfos de tamanho quatro. A Tabela 3.14 apresenta os valores de centralidade para cada um dos vértices nas redes da Figura 3.13.

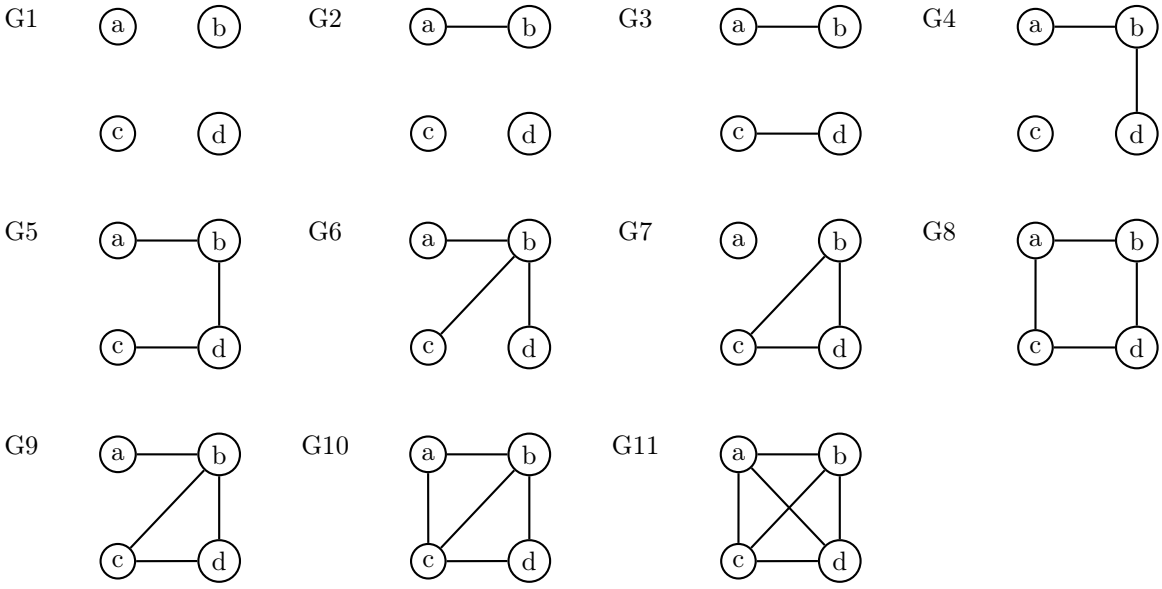

FIGURA 3.13 — Grafos não-isomorfos de tamanho quatro.

Note que as medidas de centralidade baseadas em conectividade não puderam ser aplicadas a grafos desconexos. Observe ainda que, conforme esperado, o centro de um grafo estrela (vértice b do grafo G_6) possui o maior valor possível de centralidade para todas as medidas.

	C'_D				C'_C				C'_B			
	a	b	c	d	a	b	c	d	a	b	c	d
G_1	0	0	0	0	—	—	—	—	0	0	0	0
G_2	$\frac{1}{3}$	$\frac{1}{3}$	0	0	—	—	—	—	0	0	0	0
G_3	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	—	—	—	—	0	0	0	0
G_4	$\frac{1}{3}$	$\frac{2}{3}$	0	$\frac{1}{3}$	—	—	—	—	0	$\frac{1}{3}$	0	0
G_5	$\frac{1}{3}$	$\frac{2}{3}$	$\frac{1}{3}$	$\frac{2}{3}$	$\frac{1}{2}$	$\frac{3}{4}$	$\frac{3}{4}$	$\frac{1}{2}$	0	1	1	0
G_6	$\frac{1}{3}$	1	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{3}{5}$	1	$\frac{3}{5}$	$\frac{3}{5}$	0	1	0	0
G_7	0	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{2}{3}$	—	—	—	—	0	0	0	0
G_8	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{3}{4}$	$\frac{3}{4}$	$\frac{3}{4}$	$\frac{3}{4}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$
G_9	$\frac{1}{3}$	1	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{3}{5}$	1	$\frac{3}{4}$	$\frac{3}{4}$	0	$\frac{2}{3}$	0	0
G_{10}	$\frac{2}{3}$	1	1	$\frac{2}{3}$	$\frac{3}{4}$	1	1	$\frac{3}{4}$	0	$\frac{1}{6}$	$\frac{1}{6}$	0
G_{11}	1	1	1	1	1	1	1	1	0	0	0	0

QUADRO 3.14 — Valores de centralidade C'_D, C'_D e C'_B para cada um dos vértices das redes da Figura 3.13

3.5 Centralidade de arestas em uma rede

O conceito de centralidade pode ser estendido para analisar a importância de uma determinada aresta em uma rede. As arestas mais importantes seriam as arestas que mais contribuem para a comunicação da rede.

Considere dois clusteres C_i e C_j em um grafo. Se esses clusteres são ligados por apenas uma aresta e, então toda a comunicação entre vértices de C_i e vértices de C_j contém a aresta e. Logo, e tem centralidade alta. Se os clusteres são ligados por mais de uma aresta, então pelo menos uma dessas arestas tem centralidade alta (GIRVAN; NEWMAN, 2002). Por isso, arestas de centralidade alta possuem alta probabilidade de ser uma aresta inter-cluster.

3.5.1 Centralidade da Informação

A eficiência (FORTUNATO; LATORA; MARCHIORI, 2004) (LATORA; MARCHIORI, 2003) (LATORA; MARCHIORI, 2001) com que dois vértices u e v se comunicam através de uma rede é dada por

$$\epsilon(u,v) = \frac{1}{dist(u,v)}.$$

A eficiência do grafo G mede a eficiência média com que um par de vértices se comunica no grafo e é denotada por S

$$\epsilon(G) = \frac{\sum_{\substack{u,v \in V(G), \\ u \neq v}} \epsilon(u,v)}{|V(G)|(|V(G)|-1)}.$$

A centralidade da informação (FORTUNATO; LATORA; MARCHIORI, 2004) (LATORA; MARCHIORI, 2007) mede a importância de uma aresta através da queda relativa na eficiência da rede causada pela remoção da aresta. A centralidade da informação de uma aresta e de G é

$$C^I(e) = \frac{\epsilon(G) - \epsilon(G-e)}{\epsilon(G)}.$$

3.5.2 Medidas de centralidade baseadas em *betweenness*

As medidas de centralidade baseadas em *betweenness* podem ser facilmente estendidas para medir a centralidade de arestas. Basta, ao invés de considerar os caminhos mínimos, passeios aleatórios, fluxo, ou qualquer outra medida que passa através de determinado vértice considerar a mesma medida passando por uma aresta

específica.

Por exemplo, Girvan e Newman (2002) estendeu a definição mais simples de *betweenness*. Considere os vértices $u, v \in V(G)$ e a aresta $e \in E(G)$. Denote por $\sigma(u, v, e)$ a quantidade desses caminhos que contém a aresta e. Então, o *betweenness* C_B de uma aresta e é

$$C_B(e) = \sum_{\substack{u,v \in V(G) \\ u \neq v}} \frac{\sigma(u, v, e)}{\sigma(u, v)}.$$

3.5.3 Exemplo ilustrativo das medidas de centralidade para arestas

Considere novamente a Figura 3.13, a Tabela 3.15 apresenta os valores de centralidade para cada uma das arestas desses grafos. Note que para a maior parte dos grafos a importância relativa de uma aresta medida pela centralidade da informação ou pelo *betweenness* é parecida. Nos grafos G_5 e G_{10}, entretanto isso não ocorre. A aresta $\{b, d\}$ de G_5 é considerada mais importante que as demais pela centralidade da informação, afinal, a sua distância a todos os outros vértices é menor, entretanto, seu *betweenness* é igual ao das demais arestas. No grafo G_{10}, por outro lado, apesar de $\{c, b\}$ ter *betweenness* mais baixo que as demais arestas, a sua remoção altera a eficiência da rede na mesma medida que a remoção de qualquer outra aresta e, por isso, sua centralidade de informação é igual a de todas as outras arestas do grafo.

	C^I						C_B					
	$\{a,b\}$	$\{a,c\}$	$\{a,d\}$	$\{b,c\}$	$\{b,d\}$	$\{c,d\}$	$\{a,b\}$	$\{a,c\}$	$\{a,d\}$	$\{b,c\}$	$\{b,d\}$	$\{c,d\}$
G_1	—	—	—	—	—	—	—	—	—	—	—	—
G_2	1	—	—	—	—	—	1	—	—	—	—	—
G_3	0,5	—	—	—	—	0,5	1	—	—	—	—	1
G_4	0,6	—	—	—	0,6	—	2	—	—	—	2	—
G_5	0,42	—	—	—	0,54	0,42	3	—	—	—	3	3
G_6	0,44	—	—	0,44	0,44	—	3	—	—	3	3	—
G_7	—	—	—	0,17	0,17	0,17	—	—	—	1	1	1
G_8	0,13	0,13	—	—	0,13	0,13	1,5	1,5	—	—	1,5	1,5
G_9	0,4	—	—	0,13	0,13	0,1	3	—	—	2	2	1
G_{10}	0,09	0,09	—	0,09	0,09	0,09	1,5	1,5	—	1	1,5	1,5
G_{11}	0,08	0,08	0,08	0,08	0,08	0,08	1	1	1	1	1	1

QUADRO 3.15 — Valores de centralidade da informação e de de *betweenness* simples (comunicação trafega pelos caminhos mínimos) para cada uma das arestas das redes da Figura 3.13.

3.6 Algoritmos

Diversos métodos foram desenvolvidos para detecção de comunidades em uma rede. Os métodos diferem, basicamente, na definição de cluster natural utilizada, nas técnicas de programação, nas pressuposições feitas (como, por exemplo, número, tamanho ou propriedades estruturais dos clusteres) e nos tipos de dados de entrada, que podem ser grafo simples, ponderados ou estruturas mais complexas. Entretanto, não há nenhum capaz de descobrir toda a variedade de estruturas presentes em conjuntos de dados multidimensionais (JAIN; MURTY; FLYNN, 1999). Muitos dos algoritmos populares, inclusive, são conhecidos por ter resultado ruim em vários tipos de conjuntos de dados (ZAïANE et al., 2002). Isso ocorre porque os algoritmos têm suposições implícitas sobre as características dos clusteres naturais e da solução de clusterização. Como consequência, eles devem apresentar comportamentos diferentes de acordo com as características do conjunto de dados e com os parâmetros de entrada (HALKIDI; VAZIRGIANNIS, 2001). Por isso, é fundamental que o usuário de um algoritmo conheça detalhes do processo de coleta de dados e tenha conhecimento

de domínio. Quanto mais informações tiver, o mais provável que tenha sucesso na obtenção da real estrutura (JAIN; DUBES, 1988 apud JAIN; MURTY; FLYNN, 1999).

Outra diferença entre os diferentes algoritmos está na localidade. Muitas vezes, o grafo de entrada é muito grande ou não está completamente disponível, como no caso da rede mundial de computadores, e é inviável aplicar o algoritmo no grafo todo. Nesses casos, pode-se utilizar uma abordagem local que analise só uma parte do grafo evitando o problema de escalabilidade. Isso permite, também, o uso de computação distribuída. Algoritmos locais podem ser estudados em Schaeffer (2006). Apenas algoritmos globais serão explicados nessa seção.

Os algoritmos mais comuns são os hierárquicos e os particionais. Algoritmos particionais geralmente recebem como parâmetro o número de clusteres naturais e tentam determinar iterativamente uma solução, que otimize um determinado critério, que pode enfatizar a estrutura local ou global dos dados (HALKIDI; BATISTAKIS; VAZIRGIANNIS, 2001). Essa limitação faz com que os algoritmos hierárquicos sejam mais versáteis e mais comumente utilizados que os algoritmos particionais.

Os algoritmos hierárquicos podem ser classificados em aglomerativos ou divisivos, conforme a forma de construir a solução: os aglomerativos iniciam com clusteres folha, geralmente unitários, e condensam os clusteres mais próximos a cada passo, já os divisivos iniciam com um cluster contendo todos os vértices do grafo e, a cada passo, dividem um ou mais clusteres segundo algum critério.

A seguir, são explicados alguns algoritmos de clusterização em grafos.

3.6.1 Algoritmos aglomerativos

Os métodos aglomerativos iniciam com uma solução inicial, normalmente formada por clusteres unitários e, a cada passo, dois ou mais clusteres são escolhidos para serem unidos segundo um critério específico. Segundo Newman e Girvan (2004), os métodos aglomerativos têm como problemas:

- Falham com alguma frequência em encontrar uma boa solução para redes onde a estrutura da comunidade é previamente conhecida.
- Tendem a localizar apenas os centros das comunidades e ignorar os vértices periféricos. A Figura 3.16 ilustra esse problema. O grafo é originalmente constituído das duas comunidades representadas pelos tracejados. Nos algoritmos aglomerativos, os vértices periféricos apenas são atribuídos a clusteres no final do processo, pois apresentam baixa conectividade. Por isso, quando o dendograma é cortado em um nível intermediário, são gerados dois clusteres maiores correspondentes ao centro denso das comunidades, indicados pelas arestas e vértices destacados, e vários clusteres unitários, ao invés da estrutura natural do grafo.

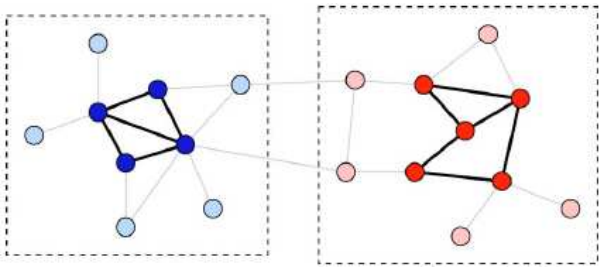

FIGURA 3.16 — Problema na detecção da periferia das comunidades por algoritmos aglomerativos. Imagem retirada de Newman e Girvan (2004).

A seguir, são mostrados alguns algoritmos aglomerativos. Inicialmente, são apresentados dois algoritmos aglomerativos baseados em medidas em qualidade, es-

pecificamente, na modularidade e na expansão. Posteriormente, é mostrado um algoritmo baseado em uma definição alternativa de comunidade, o *Distance-k clique*.

3.6.1.1 Clusterização em grafos baseada em modularidade

Newman (2003a) propôs um algoritmo aglomerativo que visa maximizar a modularidade. Como a otimização da modularidade é pelo menos exponencial segundo Newman (2003a), o Algoritmo 3.17 utiliza uma técnica gulosa.

ALGORITMO 3.17 — Clusterização em grafos baseado em modularidade

ENTRADA: Um grafo G
SAÍDA: Uma solução de clusterização em grafos de G

1: $C \leftarrow \emptyset$
2: PARA CADA $i \in V(G)$ FAÇA:
3: $C \leftarrow C \cup \{i\}$
4: FIM
5: PARA CADA $i, j \in V(G)$ FAÇA:
6: $e_{ij} \leftarrow \frac{A(i,j)}{2}$
7: FIM
8: ENQUANTO $|C| > 1$ FAÇA:
9: PARA CADA $C_i \in C$ FAÇA:
10: $a_i \leftarrow \sum_{C_j \in C} E(C_i, C_j)$
11: FIM
12: PARA CADA $C_i, C_j \in C$ FAÇA:
13: $\Delta Q_{ij} \leftarrow e_{ij} + e_{ji} - 2a_i a_j$
14: FIM
15: Encontre u,v tal que $\Delta Q_{uv} = \max_{i,j \in C} \Delta Q_{ij}$
16: Una C_u e C_v e atualize os e_{ij} somando as linhas e as colunas correspondentes aos clusteres unidos.
17: FIM

A maior parte das redes de interesse, no entanto, é esparsa e essa abordagem desperdiça muito tempo e espaço no armazenamento e na fusão de elementos da matriz com valor 0. Clauset, Newman e Moore (2004) propuseram um novo algoritmo para otimização de Q, que consiste em calcular quais as variações de Q quando pares

de clusteres são unidos e realizar a fusão naquela que resulte no maior acréscimo, ou menor decréscimo, de Q.

Uma forma direta de estender o Algoritmo 3.17 é pensar na rede como um multigrafo. Cada cluster é representado por um vértice, as arestas inter-cluster são mapeadas para arestas entre os vértices que correspondem aos clusteres e as arestas intra-cluster são representadas por auto-arestas. A matriz de adjacências do multigrafo tem elementos $A'(i,j) = 2|E(G)|e_{ij}$ e a união dos clusteres i e j corresponde à substituição das linhas e colunas i e j pela sua soma.

Esse novo algoritmo, ao invés de calcular ΔQ a cada passo, mantém explicitamente uma matriz ΔQ. Como a junção de clusteres desconexos não produz um aumento em Q, o elemento ΔQ_{ij} é armazenado apenas para pares i, j conexos.

Quando dois clusteres i e j são unidos, os elementos da matriz ΔQ são atualizados por

- Para k conectada i e j, $\Delta Q'_{jk} = \Delta Q_{ik} + \Delta Q_{jk}$.
- Para k conectada apenas a i, $\Delta Q'_{jk} = \Delta Q_{ik} - 2a_j a_k$.
- Para k conectada apenas a j, $\Delta Q'_{jk} = \Delta Q_{jk} - 2a_i a_k$.

Destas equações, decorre que Q tem um único valor máximo no algoritmo, já que depois que o maior ΔQ tornar-se negativo, todos os ΔQ só diminuem.

O Algoritmo 3.18 mostra o algoritmo resultante dessas mudanças.

3.6.1.2 Clusterização em grafos baseada em expansão

Flake, Tarjan e Tsioutsiouliklis (2004) desenvolveram um algoritmo de clusterização em grafos baseado em árvores de corte mínimos. Considere um grafo

ALGORITMO 3.18 — Novo algoritmo de clusterização em grafos baseado em modularidade

ENTRADA: Um grafo G
SAÍDA: Uma solução de clusterização em grafos de G
1: $C \leftarrow \emptyset$
2: PARA CADA $i \in V(G)$ FAÇA:
3: $C \leftarrow C \cup \{i\}$
4: $a_i \leftarrow \frac{d(i)}{2|E(G)|}$
5: FIM
6: PARA CADA $i, j \in V(G)$ FAÇA:
7: SE i, j são conexos ENTÃO,
8: $\Delta Q_{ij} \leftarrow \frac{1}{2|E(G)|} - \frac{d(i)d(j)}{|E(G)|^2}$
9: SENÃO,
10: $\Delta Q_{ij} \leftarrow 0$
11: FIM
12: FIM
13: ENQUANTO $|C| > 1$ FAÇA:
14: Selecione o maior ΔQ_{ij}
15: Junte os clusteres i e j
16: Atualize os valores de ΔQ_{ij} e a matriz a_i
17: Atualize as linha e coluna j e remova a linha e coluna i.
18: FIM

$G = (V, E, \omega)$ e construa o grafo G_α adicionando um vértice t a V e uma aresta com peso α ligando cada vértice de V a t. Ou seja,

$$G_\alpha = (V + \{t\}, E + \{\{v, t\} : v \in V\}, \omega_\alpha \colon E(G_\alpha) \to \mathbb{R}^+),$$

tal que,

$$\omega_\alpha(e) = \begin{cases} \alpha, & \text{se } t \in e \\ \omega(e), & \text{caso contrário.} \end{cases}$$

Flake, Tarjan e Tsioutsiouliklis (2004) definem que o cluster de $s \in V$ em G é S, onde $E(S, V \backslash S)$ é o corte mínimo entre s e t. Os autores provam, ainda, que α é um limite inferior para a expansão da solução de clusterização e que para todo $C_i \in C$

$$\frac{\omega(C_i, V \backslash C_i)}{|V \backslash C_i|} < \alpha.$$

Por isso, α é considerado um limite superior para a capacidade das arestas

inter-cluster e um limite inferior para capacidade das arestas intra-cluster e pode ser controlado para balancear esses critérios.

O Algoritmo 3.19 mostra os passos do algoritmo *Cut-Clustering*.

ALGORITMO 3.19 — *Cut-Clustering*

ENTRADA: Um grafo $G = (V, E, \omega)$, $\alpha \in \mathbb{R}^+$
SAÍDA: Uma solução de clusterização em grafos de G
1: $V' \leftarrow V \cup \{t\}$
2: $E' \leftarrow E$
3: PARA CADA $e \in E$ FAÇA:
4: $\quad \omega'(e) \leftarrow \omega(e)$
5: FIM
6: PARA CADA $v \in V$ FAÇA:
7: $\quad E' \leftarrow E' \cup \{v, t\}$
8: $\quad \omega'(\{v, t\}) \leftarrow k$
9: FIM
10: PARA CADA $i, j \in V(G)$ FAÇA:
11: $\quad e_{ij} \leftarrow \frac{A(i,j)}{2}$
12: FIM
13: $T \leftarrow$ árvore de cortes mínimos de (V', E', ω')
14: Retorne as componentes conexas de $G' - \{t\}$ como clusteres de G

Quando α tende a 0, o Algoritmo 3.19 produz uma solução trivial formado por apenas um cluster contendo todos os vértices. Quando α tende a ∞ o algoritmo produz clusteres unitários. Um valor intermediário de α leva a uma solução cuja expansão é limitada inferiormente por α.

É possível construir uma solução hierárquica através da reaplicação do algoritmo em um grafo construído da seguinte forma: cada cluster é representado por um vértice; as arestas intra-cluster são desconsideradas; e as arestas inter-cluster são representadas por uma única aresta com peso igual a soma dos pesos das arestas no grafo original. O Algoritmo 3.20 mostra os passos para construção dessa solução hierárquica.

ALGORITMO 3.20 — *Hierarchical CutClustering*

ENTRADA: Um grafo $G = (V, E, \omega)$, $\alpha \in \mathbb{R}^+$
SAÍDA: Uma solução de clusterização em grafos de G

1: $G^0 \leftarrow G$
2: PARA $i \leftarrow 0 \rightarrow \infty$ FAÇA:
3: SE $i = 0$ ENTÃO,
4: $\alpha^i \leftarrow \alpha$
5: SENÃO,
6: Escolha $\alpha^i < \alpha^{i-1}$
7: FIM
8: $C^i \leftarrow CutClustering(G^i, \alpha^i)$
9: SE a solução é trivial ou se um critério de parada qualquer for atingido, como tamanho ou número de clusteres ENTÃO,
10: DEVOLVA $\bigcup_{j=0}^{i} C^j$
11: SENÃO,
12: Produza G^{i+1}
13: FIM
14: FIM

3.6.1.3 *Distance-k clique*

Em Edachery et al. (1999) o problema de clusterização em grafos é traduzido na partição do conjunto de vértices no menor número de *distance-k cliques*. Note que esse problema é NP-Difícil, já que se reduz a encontrar cliques maximais para $k = 1$. Por isso, os autores desenvolveram algumas heurísticas para esse problema, todas variantes de um algoritmo base. Somente o algoritmo base será descrito aqui.

O Algoritmo 3.21 constrói uma solução inicial que será usada como base para o algoritmo de *Distance-k clique*. Todos os vértices iniciam não associados a nenhuma cluster. A cada passo, é escolhido o vértice com maior grau dentre os vértices ainda não assinalados a nenhum cluster. Esse vértice e todos os seus vizinhos ainda não associados são reunidos em um novo cluster. Esse processo é repetido até que todos os vértices pertençam a algum cluster. O algoritmo armazena a solução obtida por esse processo em um vetor indexado pelos vértices de forma que $v \in C_i, C[v] = i$.

ALGORITMO 3.21 — *InitialCluster*

ENTRADA: Um grafo G

SAÍDA: Uma solução inicial C de clusterização em grafos de G, um conjunto $ponte$ de vértices de ponte, uma lista CL de clusteres aos quais os vértices de ponte são conectados e uma lista $EstDia$ com o diâmetro de cada cluster.

1: $V' \leftarrow V(G)$
2: $i \leftarrow 0$
3: ENQUANTO $V' \neq \emptyset$ FAÇA:
4: $i \leftarrow i + 1$
5: $u \leftarrow$ vértice de maior grau em V'
6: $C[u] \leftarrow i$
7: $V' \leftarrow V' - \{u\}$
8: PARA CADA $\{u, v\} \in E(G)$ FAÇA:
9: SE $v \in V'$ ENTÃO,
10: $C[v] \leftarrow i$
11: $V' \leftarrow V' - \{v\}$
12: FIM
13: FIM
14: FIM
15: $ponte \leftarrow \emptyset$
16: PARA CADA $v \in V(G)$ FAÇA:
17: SE $\exists \{v, u\} \in E(G), C[v] \neq C[u]$ ENTÃO,
18: $ponte \leftarrow ponte \cup \{v\}$
19: $CL(v) \leftarrow \{C[u] | \{v, u\} \in E(G) \wedge C[v] \neq C[u]\}$
20: $|CL(v)| \leftarrow \sum_{j \in CL(i)} |\{v | v \in V(G) \wedge C[v] = j\}|$
21: $EstDia \leftarrow Diam(\{v | v \in V(G) \wedge C[v] = i\})$
22: FIM
23: FIM

Os vértices que estão nas bordas dos clusteres, isto é, vértices que pertencem a alguma aresta inter-cluster são denominados vértices de ponte. À cada vértice v de ponte é associada uma lista de clusteres $CL(v)$ aos quais ele é conectado por uma aresta inter-cluster. Além disso, o diâmetro de cada cluster inicial é calculado e armazenado no vetor $EstDia$ de forma que $EstDia(i)$ seja um limite superior para o diâmetro do cluster C_i.

O Algoritmo 3.22 descreve os passos e os critérios de aglomeração.A cada passo o vértice de ponte com grau mais alto m é escolhido para tentar formar um cluster maior. Inicialmente, o algoritmo tenta aglomerar todos os clusteres de $CL(u)$ com o próprio cluster $C[m]$ em um cluster

ALGORITMO 3.22 — *Distance-k clique*

ENTRADA: Um grafo G

SAÍDA: Uma solução de clusterização em grafos de G

1: Construa C, $ponte$ e CL pelo algoritmo *InitialCluster*.
2: ENQUANTO $\exists i, CL(i) \neq \emptyset$ FAÇA:
3: Encontre $m \in ponte$ tal que $|CL(m)|$ seja máximo
4: $Clist \leftarrow CL(m) \cup \{C[m])\}$
5: $ComClus \leftarrow \{v | C[v] = i \wedge i \in Clist$
6: $ComClusIndice \leftarrow$ novo índice ainda não usado para nenhum cluster
7: $LC_1 \leftarrow$ cluster de maior $EstDia$ em $Clist$
8: $LC_2 \leftarrow$ segundo cluster de maior $EstDia$ em $Clist$
9: SE $LC_1 = C[m] \vee LC_2 = C[m]$ ENTÃO,
10: $d \leftarrow 1$
11: SENÃO,
12: $d \leftarrow 2$
13: FIM
14: $EstDia(ComClusIndice) \leftarrow EstDia(LC_1) + EstDia(LC_2) + d$
15: SE $EstDia(ComClusIndice) \leqslant k$ ENTÃO,
16: Atualiza pelo algoritmo Atualiza
17: SENÃO,
18: SE $LC_1 \neq C[m]$ ENTÃO,
19: $RemClust \leftarrow LC_1$
20: SENÃO,
21: $RemClust \leftarrow LC_2$
22: FIM
23: $Clist \leftarrow Clist - RemCluster$
24: PARA CADA $u \in ComClus^C[u] = RemClust$ FAÇA:
25: $ComClus \leftarrow ComClus - u$
26: PARA CADA $i \in Clist$ FAÇA:
27: SE $i \in CL(u)$ ENTÃO,
28: $CL(u) \leftarrow CL(u) - i$
29: FIM
30: FIM
31: FIM
32: PARA CADA $i \in ComClus$ FAÇA:
33: $CL(i) \leftarrow CL(i) - RemClus$
34: FIM
35: Atualiza pelo algoritmo Atualiza
36: FIM
37: FIM

$$ComClust = \bigcup_{i \in CL(u) \vee C[u]=i} i.$$

Se o diâmetro estimado do novo cluster é menor ou igual a k dado como

entrada do algoritmo, então esse cluster é uma *distance-k clique*. Assim, a aglomeração é permanente e a lista de vértices pontes e as listas associadas a cada vértice de ponte são atualizadas através do Algoritmo 3.23. Caso contrário, o maior cluster $RemClus \neq i$ é removido de $ComClus$, de modo que o diâmetro seja reduzido e só então as listas são atualizadas. O processo finaliza quando as listas CL associadas a todos os vértices pontes é vazia.

ALGORITMO 3.23 — Atualiza

ENTRADA: Um novo cluster $ComClus$, seu índice $ComClusIndice$, a lista de clusteres $Clist$ unidos para formar o novo cluster, a lista C que indica o cluster de cada vértice, um conjunto $ponte$ de vértices de ponte, uma lista CL de clusteres aos quais os vértices de ponte são conectados e uma lista $EstDia$ com o diâmetro de cada cluster.

SAÍDA: Atualiza os valores de C, CL

1: PARA CADA $v \in ComClus$ FAÇA:
2: $C[v] \leftarrow ComClusIndice$
3: FIM
4: PARA $v \in ponte$ FAÇA:
5: PARA CADA $x \in Clist$ FAÇA:
6: SE $x \in CL(v)$ ENTÃO,
7: remova x de CL_v
8: FIM
9: FIM
10: SE $EstDia(ComClust) = k$ ENTÃO,
11: PARA CADA $i \in ComClus$ FAÇA:
12: $CL(i) \leftarrow \emptyset$
13: FIM
14: SENÃO, SE Pelo menos um cluster foi removido de $CL(v)$ ENTÃO,
15: $CL(v) \leftarrow CL(v) \cup ComClusIndice$
16: FIM
17: FIM

3.6.2 Algoritmos divisivos

Os métodos divisivos iniciam com uma solução $C = \{V(G)\}$ e iterativamente removem arestas até que restem, geralmente, apenas clusteres unitários. A cada

passo, os clusteres podem ser obtidos pelos conjuntos de vértice de cada componente conexa do grafo. Os vários algoritmos diferem na escolha de qual aresta remover a cada passo. Os métodos de centralidade podem ser utilizados para prever quais arestas são arestas inter-cluster. Altas centralidades indicam uma alta probabilidade da aresta conectar comunidades distintas.

A seguir, são mostrados alguns algoritmos divisivos. Inicialmente, são apresentados dois algoritmos baseados em centralidade de arestas, especificamente em centralidade da informação e em *betweenness*. Posteriormente, é mostrado um algoritmo baseado em uma definição alternativa de comunidade, o *Highly Connected Subgraphs*.

3.6.2.1 Algoritmo baseado em centralidade da informação

Fortunato, Latora e Marchiori (2004) apresentam o Algoritmo 3.24 que é baseado em centralidade da informação.

ALGORITMO 3.24 — Clusterização em grafos baseado em centralidade da informação

ENTRADA: Um grafo G
SAÍDA: Uma solução de clusterização em grafos de G

1: $E \leftarrow E(G)$
2: ENQUANTO $E \neq \emptyset$ FAÇA:
3: PARA CADA $e \in E$ FAÇA:
4: Calcule $C^I(e)$
5: FIM
6: $e' \leftarrow \max_{e \in E} C^I(e)$
7: $E \leftarrow E - \{e'\}$
8: Encontre as componentes conexas do grafo
9: FIM

O método consiste na remoção iterativa das arestas com maior centralidade da informação até que o grafo se constitua de um conjunto independente. O recálculo

das centralidades das arestas em cada passo é importante, pois algumas arestas inter-cluster podem apresentar, inicialmente, uma centralidade baixa.

3.6.2.2 Algoritmos baseado em *betweenness*

Girvan e Newman (2002) propuseram um algoritmo baseado em *betweenness* de arestas. Se uma rede é composta por comunidades conectadas por poucas arestas inter-cluster, todos os caminhos mínimos entre elementos de comunidades diferentes contém ao menos uma dessas arestas. Por isso, as arestas inter-cluster apresentam *betweenness* alto.

A remoção apenas das arestas inter-cluster separaria as comunidades, revelando a estrutura de comunidades da rede. O Algoritmo 3.25 remove gradualmente as arestas mais centrais de forma a separar o grafo em clusteres cada vez menores até que se chegue a uma solução composta por $|V(G)|$ clusteres unitários.

ALGORITMO 3.25 — Clusterização em grafos baseado em *betweenness* simples

ENTRADA: Um grafo G
SAÍDA: Uma solução de clusterização em grafos de G
1: $E \leftarrow E(G)$
2: ENQUANTO $E \neq \emptyset$ FAÇA:
3: PARA CADA $e \in E$ FAÇA:
4: Calcule $C_B(e)$
5: FIM
6: $e' \leftarrow \max_{e \in E} C_B(e)$
7: $E \leftarrow E - \{e'\}$
8: FIM

O recálculo dos *betweenness* após a remoção de cada aresta garante que pelo menos uma das arestas inter-cluster tenha uma centralidade alta.

Newman e Girvan (2004) propôs a extensão do Algoritmo 3.25 para o uso de outras medidas de centralidade: o *flow betweenness* e o *random-walk betweenness*

resultando no Algoritmo 3.26

ALGORITMO 3.26 — Clusterização em grafos baseado em medidas de *betweenness* genérica

ENTRADA: Um grafo G, uma medida de *betweenness* para arestas $B(e)$
SAÍDA: Uma solução de clusterização em grafos de G
1: $E \leftarrow E(G)$
2: ENQUANTO $E \neq \emptyset$ FAÇA:
3: PARA CADA $e \in E$ FAÇA:
4: Calcule $B(e)$
5: FIM
6: $e' \leftarrow \max_{e \in E} B(e)$
7: $E \leftarrow E - \{e'\}$
8: FIM

3.6.2.3 Algoritmo *Highly Connected Subgraphs*

O algoritmo *Highly Connected Subgraphs* (HCS) (HARTUV; SHAMIR, 2000) não usa a definição de cluster como conjunto de vértices, mas sim como os subgrafos induzidos por esses conjuntos. Hartuv e Shamir (2000) define que uma comunidade em um grafo G é um subgrafo H de G altamente conectado. O Algoritmo 3.27 descreve um algoritmo baseado nessa noção.

ALGORITMO 3.27 — HCS

ENTRADA: Um grafo G
SAÍDA: Uma solução de clusterização em grafos G
1: $(H, H') \leftarrow MINCUT(G)$
2: SE $k(G) > |V(G)|/2$ ENTÃO,
3: DEVOLVA G
4: SENÃO,
5: $HCS(H)$
6: $HCS(H')$
7: FIM

Se algoritmo HCS identifica que o grafo é altamente conectado, então ele é retornado como um cluster. Caso contrário, o algoritmo é chamado recursiva-

mente nos subgrafos construídos pelo procedimento $MINCUT(G)$. O procedimento $MINCUT(G)$ encontra o corte mínimo $E(S, V - S)$ de G e retorna os subgrafos $H = G[S]$ e $H' = G[V - S]$ resultantes da remoção das arestas do corte do grafo original.

Vértices simples não são considerados clusteres e são agrupadas em um conjunto de unitários. O conjunto de subgrafos retornado por HCS constitui a solução. A Figura 3.28 mostra um exemplo da aplicação do algoritmo HCS.

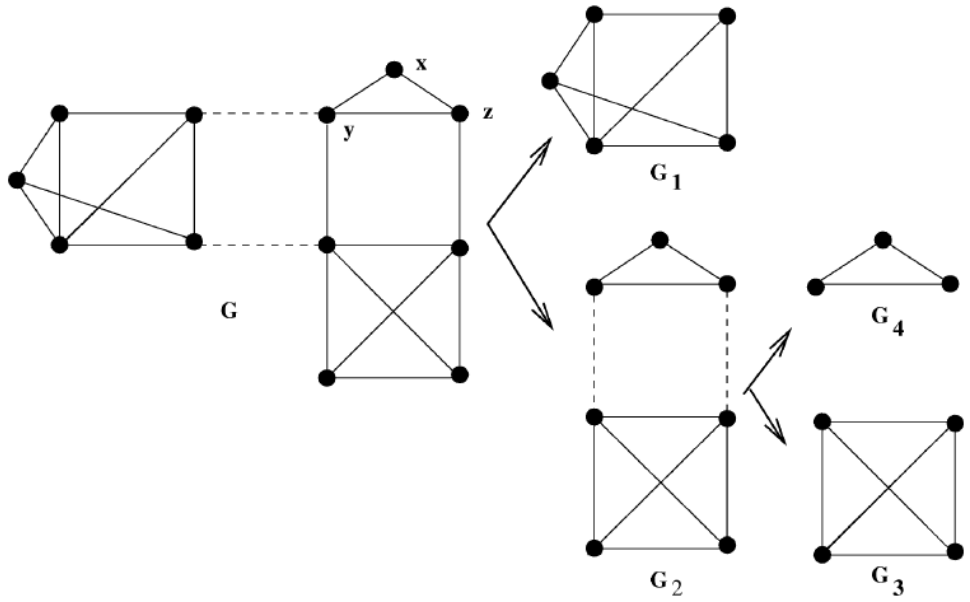

FIGURA 3.28 — Exemplo da aplicação do algoritmo HCS. Imagem retirada de Hartuv e Shamir (2000).

3.6.3 Algoritmos não hierárquicos

Os algoritmos não hierárquicos necessitam de alguma informação externa sobre o número de clusteres naturais do grafo. A seguir são descritos dois algoritmos não hierárquicos.

3.6.3.1 Clusterização em grafos baseada em k-objetos

Matula (apud DONGEN, 2000) considera que as noções de k-*bond*, k-componente e k-bloco podem ser usadas para definir uma comunidade. Algoritmos hierárquicos podem ser obtidos variando-se o k e, em cada nível, considerando que os clusteres são os k-objetos juntamente com os clusteres unitários formados por cada um dos vértices não assinalados a nenhum k-objeto.

Os métodos de clusterização em grafos baseados em k-objetos são muito restritos, pois exigem que k seja fixo em todo o grafo e, por isso, as variações locais em termos de conectividade têm um impacto grande na solução. Observe a Figura 3.29, a solução natural evidentemente é a que cada cluster induz uma componente conexa do grafo. Na solução de clusterização em grafos de k-blocos para $k = 3$, apenas a 3-clique seria corretamente identificada, a 4-clique formaria 4 3-blocos distintos. Para $k = 4$, apenas a 4-clique seria corretamente identificada, a 5-clique formaria 5 4-blocos distintos e a 3-clique formaria 3 clusteres unitários.A Figura 3.30 mostra um grafo e seus 3-blocos. Note que a solução dada por $k = 2$ conteria um cluster contendo todos os vértices do grafo e se $k = 3$ poucos vértices pertencem a comunidades e a grande maioria forma clusteres unitários.

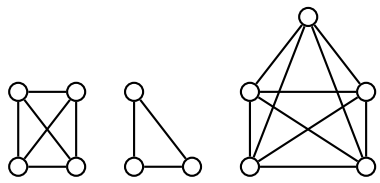

FIGURA 3.29 — Solução de clusterização em grafos ideal de um grafo cujas componentes conexas são cliques.

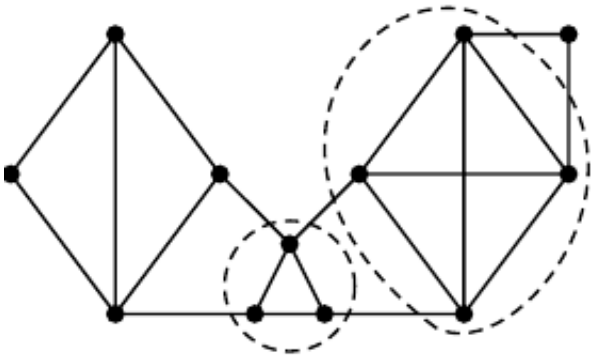

FIGURA 3.30 — Grafo e seu agrupamento com 3-blocos. Imagem retirada de (DONGEN, 2000).

3.6.3.2 Clusterização em grafos baseada em circuitos elétricos

Wu e Huberman (2004) propuseram um método para encontrar comunidades baseado em circuitos elétricos. A parte da teoria de circuitos elétricos necessária ao entendimento dessa seção pode ser encontrada em Nilsson e Riedel (2003).

Considere uma solução de clusterização em grafos composta por duas comunidades e dois vértices u e v pertencentes a comunidades diferentes. Imagine um circuito em que cada aresta seja um resistor e com uma bateria conectada entre u e v. u e v possuem o maior, e o menor potencial dentre todos os vértices do grafo. Os demais vértices possuem potenciais que podem ser obtidos pelas equações de Kirchhoff, que diz o fluxo de corrente que chega a um vértice é igual ao fluxo que sai do mesmo (EDMINISTER; NAHVI, 1985). Como os vértices em uma comunidade são densamente conectados, então, suas tensões tendem a ter valores próximos e a resistência local da comunidade é baixa. Como as comunidades são conectadas por poucos resistores, a resistência entre as comunidades é grande e há uma diferença grande de voltagem entre as duas comunidades como na Figura 3.31. Assim é possível prever a qual comunidade um determinado vértice pertence a partir do seu potencial.

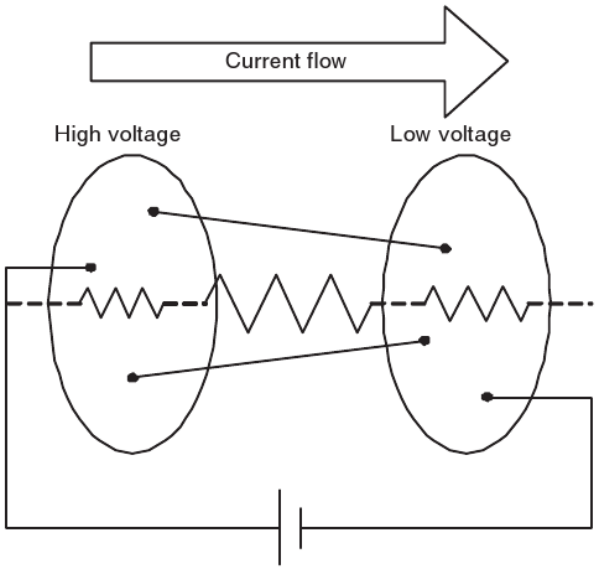

FIGURA 3.31 — Analogia de um grafo com um circuito elétrico.

Wu e Huberman (2004) exemplifica o porquê que essa analogia funciona através de alguns casos:

- Considere um vértice w tal que $\Gamma(w) = \{x\}$. Como w tem grau 1, w é um fio solto no circuito, por isso, não é possível haver corrente na aresta $\{w, x\}$. Assim, os potenciais V_x e V_w são iguais e, portanto, x e w pertencem à mesma comunidade, conforme o esperado.

- Considere um vértice w tal que $\Gamma(w) = \{x, y\}$. Como $\{w, x\}$ e $\{w, y\}$ têm a mesma resistência, $V_w = \frac{V_x + V_y}{2}$. Se x e y pertencem à mesma comunidade, então w também pertencerá. Do contrário, V_w pode estar perto do limite e ser relativamente difícil aferir a sua comunidade, mas isso é exatamente o esperado para vértices conectados com mais de uma comunidade.

- Considere um vértice w tal que $\Gamma(w) = \{z_1, ..., z_n\}$. Pelas equações de Kirchhoff, $V_w = \frac{\sum_{i=0}^{n} V_{z_i}}{n}$, isto é a média dos potenciais de seus vizinhos. Se a maior parte dos vizinhos de w pertencem a uma mesma comunidade, então V_w tende a pertencer à mesma comunidade.

Nos casos em que não se sabe a priori que dois vértices quaisquer pertençam a comunidades diferentes, pode-se escolher aleatoriamente dois vértices e aplicar o algoritmo muitas vezes. Cerca de metade dos resultados estariam corretos. Se os vértices escolhidos não forem vizinhos, então a maioria dos resultados estará correta. Assim, as comunidades podem ser determinadas pela maioria.

A extensão do método para detecção de várias comunidades é um tanto obscura. Wu e Huberman (2004) exemplifica para um caso de um grafo com 115 vértices que se sabe a priori ter 13 comunidades. Eles assumiram que cada comunidade deveria ter tamanho entre 4 e 13. Para fazer isso, a cada aplicação do método ordenaram os vértices em ordem crescente de potencial $V_1 \leqslant V_2 \leqslant ... \leqslant V_n$ e mediram as diferenças de potencial $V_{i+1} - V_i, i \in \{5, 6, ..., 13\}$. O i que resultasse na maior diferença de potencial indica uma comunidade $\{V_1, ..., V_i\}$. O mesmo procedimento é feito na outra extremidade do vetor ordenado de potenciais. As duas comunidades eram candidatos a comunidades naturais. Esse procedimento foi repetido 50 vezes. A comunidade de certo vértice é determinada dentre todas as comunidades que o contém através da comunidade que aparece mais vezes. O vértice inicial é escolhido como sendo o que aparece com maior frequência dentre os candidatos. O segundo vértice é o vértice do restante do grafo que aparece com mais frequência dentre os candidatos, e assim sucessivamente até que o procedimento seja repetido 13 vezes.

3.7 Discussões

A revisão da literatura mostrou que existem vários algoritmos de clusterização em grafos e diversas medidas de qualidade utilizadas para aferir se o resultado de um algoritmo é válido. Entretanto, não existe uma metodologia ou medida de qualidade

ótima que possa ser utilizada para toda a variedade de estrutura de comunidades existentes. Além disso, não há uma descrição formal do problema de clusterização em grafos como problema computacional.

O próximo capítulo apresentará uma proposta de formalização para o problema de clusterização em grafos. Assim como uma proposta de como adaptar esse novo problema formalizado para descrever algumas das medidas e algoritmos já vistas nesse capítulo.

4 FORMALIZAÇÃO DO PROBLEMA DE CLUSTERIZAÇÃO EM GRAFOS

Esse trabalho propõe-se a formalizar o problema de clusterização em grafos. Ou seja, defini-lo de forma rigorosa, bem definida e não ambígua. Dada a natureza informal do problema de clusterização em grafos, não é possível descrevê-lo tendo como entrada apenas o grafo, pois isso acarretaria em uma definição rígida do que caracteriza uma comunidade. Nesse trabalho, essa informalidade é abstraída descrevendo o problema como o problema de encontrar uma solução de clusterização em grafos que satisfaça um conjunto de restrições e minimize uma função objetivo.

A escolha de qual função objetivo e de qual conjunto de restrições utilizar depende das características desejáveis na solução de clusterização em grafos natural. As medidas de qualidade e as medidas de centralidade podem ser adaptadas para tornarem-se funções objetivo ou para indicar restrições na solução encontrada. Entretanto, essas adaptações não descrevem a totalidade de características desejáveis na solução de clusterização natural. Para ilustrar isso, os algoritmos de clusterização em grafos estudados no capítulo anterior são revistos. Para cada algoritmo, tentar-se-á propor uma função objetivo específica. Caso a adaptação não seja simples, será discutido o porquê de a formalização proposta não ser adequada para descrever o algoritmo em questão.

O restante desse capítulo está descrito da seguinte forma:

- A Seção 4.1 apresenta alguns conceitos iniciais necessários para o entendimento da formalização do problema. Especificamente, são estudadas as definições de problema computacional e de algoritmo.

- A Seção 4.2 propõe uma descrição formal para o problema de clusterização em grafos.

- A Seção 4.3 discorre sobre a robustez da descrição proposta para o problema de clusterização em grafos.

- A Seção 4.4 propõe como adaptar as medidas de qualidade e as medidas de centralidade de vértices e de arestas para descrever as características desejáveis de uma solução de clusterização natural.

- A Seção 4.5 analisa quais dos algoritmos estudados no Capítulo 3 podem ser descritos pela definição formal do problema de clusterização em grafos.

- A Seção 4.6 apresenta algumas discussões e considerações finais do capítulo.

4.1 Conceitos iniciais

Um problema computacional é um problema matemático definido de forma que a relação entre as variáveis do problema, denotadas por instâncias, e as respostas é não ambígua. Isto é, para cada uma das instâncias do problema, o conjunto de respostas é bem definido. Formalmente, tem-se que

DEFINIÇÃO 4.1: Um problema computacional P é uma tripla $P = (I(P), R(P), \Phi_P)$, tal que: $I(P)$ é o conjunto das instâncias de P; $R(P)$ é o conjunto das respostas de P; e $\Phi_P \colon I(P) \rightarrow 2^{R(P)}$ é uma função que associa cada instância em $I(P)$ às suas respostas,

onde 2^D denota o conjunto dos subconjuntos de D, também chamado de conjunto das partes. Essa definição foi adaptada da definição encontrada em (GAREY; JOHNSON, 1979).

Note que essa definição permite que uma instância arbitrária tenha respostas diferentes. Por isso, a primeira vista, poderia parecer que essa definição é ambígua. Entretanto, Φ_P especifica rigidamente quais são as respostas possíveis para cada instância, considerando que não há nenhuma preferência dentre as respostas. Logo, o problema poderia ser resolvido por um mecanismo que obtenha qualquer uma das respostas pertencentes ao conjunto definido por Φ_P. Isso pode ser alcançado através de algoritmos.

O conceito de algoritmo é semelhante aos conceitos de receita, método, processo, técnica, procedimento e rotina. No entanto, um algoritmo, além de ser um conjunto finito de regras que resultam em uma sequência de operações para resolver um tipo específico de problema, deve apresentar cinco características importantes: finitude, definitude, entradas, saídas e efetividade (KNUTH, 1997).

A propriedade finitude restringe que um algoritmo deve terminar após um número finito de passos. A característica definitude diz que cada passo do algoritmo deve ser bem definido. Ou seja, as ações a serem executadas devem ser especificadas rigorosamente e sem ambiguidades. Um algoritmo também deve, geralmente, apresentar efetividade. Isto é, as operações executadas pelos algoritmos devem ser suficientemente simples a ponto de serem executadas exatamente e em um tempo finito por um ser humano munido de lápis e papel. Além disso, um algoritmo tem zero ou mais entradas, que são valores dados para inicialização do algoritmo, e uma ou mais saídas, que são valores que tem uma relação específica com as entradas.

Baseado nessa definição, observe que a solução para o problema P é um algoritmo A que descreve uma sequência finita de instruções que, tendo como entrada $I \in I(P)$, resultam em $A(I) \in \Phi_P(I)$. Isto é, a solução para o problema P é um algoritmo A que para cada instância de P, encontra uma das suas respostas.

Observe, entretanto, que nem todo algoritmo é necessariamente a solução de algum problema computacional. As heurísticas, por exemplo, são algoritmos que, para

cada instância do problema, em geral, retornam saídas com qualidade satisfatória, mas sem nenhuma garantia de otimalidade ou aproximação da resposta teórica. Note, ainda, que a sequência de passos definido pelo algoritmo não é necessariamente determinística (MITZENMACHER; UPFAL, 2005).

4.2 Formalização do problema

Essa seção propõe-se a formalizar o problema de clusterização em grafos. Por formalização, entenda-se descrever o problema de forma rigorosa, bem definida e não ambígua. Ou seja, a descrição do problema deve ser suficientemente específica para que, qualquer leitor, sem a necessidade de recorrer a informações adicionais, possa entender especificamente quais as instâncias e as respostas do problema e qual a relação entre as instâncias e as respostas. Isso é alcançado descrevendo o problema como um problema computacional.

No Capítulo 3, no entanto, foi discutido que, devido à informalidade da definição do que seria uma solução de clusterização em grafos e à falta de consenso em quais características são desejáveis em uma solução de clusterização em grafos, não é possível descrever o problema de clusterização em grafos de forma bem definida se a instância for apenas o grafo de entrada. Afinal, isso implicaria em definir rigidamente o que é um cluster natural. Dessa forma, é necessário encapsular essa ambiguidade, que é própria da natureza do problema, como um dos parâmetros que compõem a instância do problema.

Isso pode ser obtido definindo o problema de clusterização em grafos como o problema de encontrar uma solução de clusterização que satisfaça um conjunto de restrições e minimize uma função objetivo. Formalmente, temos:

DEFINIÇÃO 4.2: O problema de clusterização em grafos $ClustGrafos$ é um problema computacional caracterizado por:
Instância: Um grafo conexo $G = (V, E, \omega)$, uma função $r\colon Clust(G) \rightarrow \{verdadeiro, falso\}$ que descreve um conjunto de restrições e uma função objetivo $f\colon Clust(G) \rightarrow \mathbb{R}$.
Resposta: Uma solução de clusterização $C \in Clust(G)$ tal que: $r(C)$ é verdadeiro e $f(C)$ é mínimo.

Note que, conforme explicado na Capítulo 3, cada cluster induz necessariamente um subgrafo conexo no grafo original. Portanto, para resolver o problema de clusterização em um grafo desconexo G, basta resolver o problema de clusterização para cada componente conexa de G.

TEOREMA 4.3: *O problema de clusterização em grafos descrito segundo a Definição 4.2 é um problema computacional.*

Demonstração:

Segundo a Definição 4.1, um problema P é um problema computacional se existe uma função $\Phi_P\colon I(P) \rightarrow 2^{R(P)}$ que relaciona cada instância de $I(P)$ com as suas respostas em $R(P)$.

Da Definição 4.2, decorre que

- O conjunto de instâncias é $I(ClustGrafos) = (G, r, f)$, onde: G é um grafo, $r\colon Clust(G) \rightarrow \{verdadeiro, falso\}$ e $f\colon Clust(G) \rightarrow \mathbb{R}$.
- O conjunto de respostas é $R(ClustGrafos) = Clust(G)$.

Para mostrar que $ClustGrafos$ é um problema computacional, basta demonstrar a existência de uma função Φ que mapeia cada instância para seu conjunto de respostas.

Considere a função $\Phi\colon I(ClustGrafos) \rightarrow 2^{R(ClustGrafos)}$ tal que

$$\Phi(I) = \{C \in Clust(G) | r(C) = verdadeiro \wedge (\forall C' \in Clust(G), (r(C') = falso \vee f(C') \geqslant f(C)))\}$$

Para mostrar que Φ mapeia cada entrada ao seu conjunto de respostas, basta demonstrar que, para uma instância arbitrária I_0, (1) não existe uma resposta que não pertence a $\Phi(I_0)$ e (2) não existe algum elemento em $\Phi(I_0)$ que não seja resposta.

(1): Suponha que exista uma resposta R_0 de I_0 tal que $R_0 \notin \Phi(I_0)$.

Se R_0 não pertence a $\Phi(I_0)$, então não é verdade que "$r(R_0)$ é verdadeiro e $\forall C' \in Clust(G), r(C')$ é falso $\vee f(C') \geqslant f(R_0)\}$". Logo, ou (i) $r(R_0)$ é falso ou (ii) $\exists C' \in Clust(G)$ tal que $r(C')$ é verdadeiro $\wedge f(C') < f(R_0)$.

(i) Se $r(R_0)$ é falso, então R_0 não é resposta de I_0. Mas, por hipótese, R_0 é resposta de I_0, o que configura um absurdo.

(ii) Se $\exists C' \in Clust(G)$ tal que $r(C')$ é verdadeiro $\wedge f(C') < f(R_0)$, então $f(R_0)$ não é mínimo. Logo R_0 não é resposta de I_0. Entretanto, como por hipótese, R_0 é resposta de I_0, isso configura um absurdo.

Logo, não existe uma resposta que não pertença a $\Phi(I_0)$.

(2): Suponha que exista $C \in \Phi(I_0)$ que não seja resposta de I_0.

Se existe $C \in \Phi(I_0)$ que não é resposta de I_0, então não é verdade $r(C)$ é verdadeiro e $f(C)$ é mínimo. Logo, ou (i) $r(C)$ é falso ou (ii) $f(C)$ não mínimo.

(i) Se $r(C)$ é falso, então $C \notin \Phi(I_0)$. Mas, por hipótese, $C \in \Phi(I_0)$, o que configura um absurdo.

(ii) Se $f(C)$ não mínimo, então existe $C' \in Clust(G)$ tal que $r(C')$ é verdadeiro $\wedge f(C') < f(C)$. Logo $C' \notin \Phi(I_0)$. No entanto, como por hipótese, $C \in \Phi(I_0)$, isso configura um absurdo.

Logo, se C está em $\Phi(I_0)$, então C não é resposta.

Logo, o problema de clusterização em grafos descrito segundo a Definição 4.2 é um problema computacional.

□

A partir desse momento, o problema de clusterização em grafos descrito segundo a Definição 4.2, será denominado simplesmente de problema de clusterização em grafos.

4.3 Robustez da descrição formal do problema de clusterização em grafos

Até esse momento foi afirmado que a Definição 4.2 descreve o problema de clusterização em grafos. Entretanto, ainda é preciso analisar se essa definição seria robusta o suficiente para descrever o problema resolvido por cada um dos algoritmos de clusterização em grafos. Para facilitar a leitura, quando a definição proposta permite descrever o problema resolvido por um determinado algoritmo, diz-se que a definição abrange esse algoritmo.

Como não há nenhum consenso geral sobre o que caracteriza uma solução de clusterização em grafos natural, não é possível decidir de forma absoluta se a Definição 4.2 abrange todos os algoritmos de clusterização em grafos. Alternativamente, pode-se analisar cada um dos algoritmos e decidir se o problema resolvido pelo algoritmo poderia ser descrito por essa definição. No entanto, essa abordagem é

impraticável, afinal, existem diversos algoritmos e técnicas de clusterização em grafos espalhados em trabalhos de diversas áreas do conhecimento.

Além disso, devido à forma com que muitos desses algoritmos e técnicas são propostos, muitas vezes não é possível descobrir qual é o problema que o algoritmo resolve. Isto é, nem sempre está claro qual é exatamente a propriedade desejável da solução de clusterização em grafos natural. Na prática, muitas vezes o autor apenas define algum critério para escolha de qual par de clusteres unir ou de qual cluster dividir em um procedimento hierárquico. Em outros trabalhos, entretanto, os algoritmos são criados de uma forma tão complexa que não é possível inferir nenhuma informação de interesse sobre qual as propriedades comuns aos resultados da clusterização de diferentes grafos.

Observe, ainda, que nem sempre o algoritmo é construído de forma a efetivamente otimizar uma propriedade desejável na solução de clusterização natural, esse é o caso, por exemplo, quando se utiliza heurísticas ou outras técnicas aproximadas. Nessa situação, apesar do algoritmo não resolver exatamente um problema, geralmente é possível inferir qual o problema que o autor pretendia resolver. Para simplificar a leitura desse texto, esse problema será chamado de problema resolvido pelo algoritmo, embora isso não ocorra de fato.

Devido a todas essas particularidades do campo de clusterização em grafos, não é possível decidir de forma absoluta se o problema descrito segundo a Definição 4.2 é suficientemente robusto para descrever todos os algoritmos de clusterização existentes. No entanto, pode-se analisar alguns algoritmos de clusterização em grafos e decidir quais deles podem ser descritos segundo essa definição. Assim, é possível decidir se um algoritmo não pode ser descrito pela definição proposta devido a uma particularidade na forma como o algoritmo foi concebido ou se isso ocorre devido à definição proposta não ser robusta o suficiente.

Esse procedimento será realizado posteriormente através da análise dos al-

goritmos estudados na Seção 3.6. No entanto, como alguns desses algoritmos se baseiam em medidas de qualidade e em medidas de centralidade, antes de efetivamente examinar os algoritmos, será analisado como adaptar essas medidas para serem utilizadas como função objetivo ou como conjunto de restrições.

4.4 Utilização de medidas de qualidade e de centralidade de vértices e arestas para descrição das características desejáveis de uma solução de clusterização ótima

A escolha de qual função objetivo e conjunto de restrições que devem ser utilizados como parte da instância de um problema de clusterização em grafos depende das características desejáveis na solução de clusterização em grafos natural. Observando os algoritmos de clusterização em grafos apresentados na Seção 3.6, essas características poderiam ser descritas a partir das medidas de qualidade e das medidas de centralidade de vértices e de arestas.

As medidas de qualidade, por indicarem a qualidade de um cluster isolado ou de uma solução de clusterização em grafos como um todo, podem ser adaptadas para tornarem-se funções objetivo. As medidas de centralidade, por outro lado, como indicam propriedades de vértices ou de arestas isoladas são melhor utilizadas para indicar restrições na solução.

4.4.1 Funções objetivo baseadas nas medidas de qualidade

As medidas de qualidade são utilizadas para medir a qualidade de uma solução de clusterização em grafos ou de um cluster isolado. Ou seja, as medidas de qualidade representam propriedades desejáveis globais de uma solução de clusterização em grafos ou de um cluster específico. Por isso, as medidas de qualidade são candidatas naturais a função objetivo do problema de clusterização em grafos.

As medidas que se referem à qualidade de uma clusterização em grafos podem ser utilizadas diretamente, já as medidas que são realizadas em clusteres isolados, por outro lado, devem ser adaptadas para referir-se à solução de clusterização como um todo. A seguir, é apresentada como as medidas de qualidade apresentadas na Seção 3.3 poderiam ser utilizadas como funções objetivo do problema de clusterização em grafos.

4.4.1.1 Funções objetivo baseadas em coesão

As medidas de coesão consideram apenas a qualidade interna de um cluster e, por isso, são utilizadas apenas para aferir a qualidade de um cluster isolado, através da qualidade do subgrafo induzido por ele. A extensão das medidas baseadas em coesão para medir a qualidade de uma solução de clusterização em grafos pode ser obtida relacionando a qualidade da solução com a qualidade de todos os clusteres.

Essa relação pode ser determinada em termos de propriedades estatísticas no conjunto das qualidades de todos os clusteres pertencentes à solução. Como exemplo, a coesão de uma solução de clusterização, poderia ser caracterizada pela maior

ou pela menor coesão dentre o conjunto de coesões dos clusteres, caracterizando um limite superior ou inferior para a qualidade dos mesmos. Outras propriedades utilizadas poderiam ser, dentre outras, a média aritmética, a média geométrica, a mediana ou a moda. A utilização dessas propriedades seria uma tentativa de aferir uma qualidade média dos clusteres, permitindo que uma parcela deles tenha qualidade muito além ou aquém do desejado.

Note que, tanto a densidade, quanto as medidas de compacidade C_p e C_p^* apresentam valores mais altos para clusteres mais compactos. Por isso, devem ser modificadas para que valores menores representem bons clusteres, já que, segundo a definição do problema de clusterização em grafos, a função objetivo é minimizada. Isso pode ser obtido facilmente multiplicando essas medidas por uma constante negativa, tipicamente -1, ou invertendo o valor da medida. Entretanto, uma inversão do valor das medidas não seria adequada, já que essas medidas podem assumir valor 0, como por exemplo, no caso da densidade de clusteres que são conjuntos independentes no grafo.

Considerando que seja escolhido como critério de extensão que a qualidade de uma solução de clusterização seja um limite inferior para qualidade dos clusteres, tem-se como exemplo de função objetivo:

- Utilizando densidade,

$$f(C) = \min_{C_i \in C} -\delta(C_i).$$

- Utilizando compacidade C_P,

$$f(C) = \min_{C_i \in C} -Cp(C_i).$$

- Utilizando compacidade C_P^*,

$$f(C) = \min_{C_i \in C} -Cp^*(C_i).$$

Observe, ainda, que a densidade e as compacidades C_p e C_p^*, são máximas para um cluster que induz um subgrafo completo no grafo clusterizado. Ou seja, a minimização irrestrita de uma função objetivo baseada em medidas de coesão para obtenção da solução de clusterização de um grafo G levaria à solução com todos os clusteres iguais a cliques no grafo original, afinal, nesse caso todos os clusteres teriam coesão máxima e qualquer propriedade estatística aplicada no conjunto de qualidades dos clusteres levaria também à qualidade máxima.

Por isso, para a utilização das medidas de coesão é necessário que exista alguma restrição adicional que proíba que as soluções indesejadas sejam consideradas válidas. Exemplos simples seriam uma restrição baseada no tamanho ou no número de clusteres ou, ainda, no número de arestas inter-cluster.

4.4.1.2 Funções objetivo baseadas em coesão e separação

As medidas de qualidade baseadas em coesão e separação, em geral, já são definidas para estimar a qualidade de uma solução de clusterização em grafos. Por isso, essas medidas necessitam nenhuma ou pouca modificação para ser utilizadas como função objetivo.

4.4.1.2.1 Funções objetivo baseadas em distâncias

O índice de Dunn e o índice de Davies Bouldin utilizam medidas de distância para identificar uma solução de clusterização em grafos compacta e bem separada. Valores altos do índice de Dunn e valores baixos do índice de Davies Bouldin correspondem a soluções com qualidade elevada. Por isso, uma função objetivo baseada no índice de Davies Bouldin seria o próprio índice, enquanto o índice de Dunn deve ser modificado para que valores baixos indiquem boas soluções, como por exemplo, pela multiplicação por uma constante negativa. Ou seja,

- Utilizando o índice de Dunn,

$$f(C) = -D(C).$$

- Utilizando o índice de Davies Bouldin,

$$f(C) = DB(C).$$

Por ser muito sensível a ruídos, a minimização de uma função objetivo baseada no índice de Dunn pode levar a uma solução muito desviada da desejada. Por ruídos, entenda-se qualquer diferença entre o grafo que representa o conjunto de dados e a estrutura real dos dados, isto é, o grafo que representaria fielmente o conjunto de dados. Como exemplo de ruídos, pode-se destacar, ausência de arestas no grafo que estão presentes na estrutura real dos dados, presença de arestas que não estão na estrutura real e pesos com valores incorretos indicando ligações mais fortes ou mais fracas que o real. O índice de Davies Bouldin é mais robusto que o índice de Dunn e, portanto, menos sensível a ruídos e mais adequado para ser utilizado como função objetivo a ser minimizada.

Como os índices de Dunn e Davies Bouldin não fazem nenhuma averiguação acerca das conectividades intra-cluster e inter-cluster, uma função objetivo baseada nesses índices levará a uma solução de clusterização com conectividades intra-cluster e inter-cluster relativamente arbitrárias.

4.4.1.2.2 Funções objetivo baseadas na conectividade inter-cluster e intra-cluster

A densidade relativa, assim como as medidas baseadas apenas em coesão, afere a qualidade de um cluster isolado e valores altos indicam um cluster de boa qualidade. Por isso, a adaptação para ser utilizada como uma função objetivo que meça a qualidade da solução de clusterização deve ser feita por um processo análogo ao utilizado na extensão das medidas baseadas em coesão descrito anteriormente.

Novamente, considerando que seja escolhido como critério de extensão que a qualidade de uma solução de clusterização seja um limite inferior para qualidade dos clusteres, então um exemplo de função objetivo baseada em densidade relativa poderia ser

$$f(C) = \min_{C_i \in C} -\rho(C_i).$$

A cobertura é utilizada para medir a qualidade de uma solução de clusterização. Quanto maior o valor da cobertura, menor o número de arestas inter-cluster e melhor a qualidade da solução. Novamente, como valores altos de cobertura indicam uma boa solução, a cobertura deve ser multiplicada por uma constante negativa para ser utilizada como função objetivo. Ou seja,

$$f(C) = cobertura(C).$$

No entanto, a minimização irrestrita da cobertura forma uma solução trivial formada por um único cluster igual ao conjunto de vértices do grafo. Por isso, uma função objetivo baseada na cobertura não deve ser minimizada irrestritamente.

A expansão e a condutância são definidas tanto para cortes e clusteres isolados como para uma solução de clusterização em grafos. A expansão e a condutância de uma solução servem como limites inferiores para a expansão e a condutância de qualquer corte intra-cluster. Por isso, valores altos dessas medidas indicam uma solução de qualidade alta. Portanto, a expansão e a condutância devem ser multiplicadas por uma constante negativa a fim de ser utilizadas como função objetivo. Ou seja,

- Utilizando expansão,

$$f(C) = -\psi(C).$$

- Utilizando condutância,

$$f(C) = -\phi(C).$$

Observe que, conforme discutido na Seção 3.3, a expansão e a condutância são insuficientes como critério de agrupamento, portanto, sua utilização como função objetivo requer que seja aliada a utilização de um conjunto de restrições que englobe as deficiências dessas medidas, ou seja, que considere o peso das arestas inter-cluster e o tamanho relativo dos clusteres.

A medida (α, ϵ) é uma medida bi-critério criada para tornar a expansão e a condutância suficientes para ser utilizadas como critério de agrupamento. Conforme apresentado anteriormente, uma solução C é (α, ϵ) se

$$\phi(C) \geqslant \alpha,$$

e

$$\frac{E'(C)}{E(C)+E'(C)} \leqslant \epsilon.$$

Fazendo uma manipulação simples, pode-se dizer que uma solução C é (α, ϵ) se

$$-\phi(C) \leqslant -\alpha,$$

e

$$\frac{E'(C)}{E(C)+E'(C)} \leqslant \epsilon.$$

Observe que não é possível definir uma função objetivo baseada na medida (α, ϵ) sem definir também uma função que englobe restrições. As restrições são definidas trivialmente pela definição modificada de solução (α, ϵ), fazendo:

$$r(C) = \begin{cases} verdadeiro, & \text{se } \frac{E'(C)}{E(C)+E'(C)} \leqslant \epsilon \wedge -\phi(C) \leqslant -\alpha \\ falso, & \text{caso contrário} \end{cases}$$

Uma função objetivo para ser utilizada com essa medida teria que fazer uma minimização bi-variável em $-\alpha$ e ϵ.

4.4.1.2.3 Funções objetivo baseadas na comparação com modelos ideais ou aleatórios

A performance mede o quanto uma solução de clusterização em grafos se afasta de uma solução ideal em que o grafo induzido fosse construído de tal forma que cada cluster induzisse um grafo completo e não houvesse nenhuma aresta inter-cluster. A modularidade, por outro lado, mede o quanto uma solução é melhor que

um grafo com as arestas distribuídas aleatoriamente, mas respeitando os graus dos vértices. Valores altos de performance e modularidade indicam uma solução de clusterização em grafos de qualidade alta. Por isso, funções objetivo podem ser obtidas multiplicando-se esses índices por uma constante negativa. Isto é,

- Utilizando performance,

$$f(C) = -performance(C).$$

- Utilizando modularidade,

$$f(C) = -Q(C).$$

4.4.2 Restrições baseadas nas medidas de centralidade

As medidas de centralidade não são facilmente extensíveis para medir alguma qualidade global de uma solução de clusterização. Entretanto, essas medidas podem ser trivialmente incorporadas como restrições em vértices e arestas da solução de clusterização.

Vértices localizados em posições centrais são fortes candidatos a ser vértices de fronteira, isto é, vértices que são extremos de uma aresta inter-cluster. Disso, deriva que é possível restringir a centralidade dos vértices internos dos clusteres, aqueles que não são extremos de nenhuma aresta inter-cluster, a um valor máximo arbitrário ou relativo à centralidade máxima de qualquer vértice na rede.

Analogamente, arestas com centralidade alta possuem alta probabilidade de ser uma aresta inter-cluster. Pelo mesmo princípio, é possível restringir a centralidade

de uma aresta intra-cluster a um valor máximo arbitrário ou relativo à centralidade máxima de aresta na rede.

4.5 Descrição formal dos problemas resolvidos pelos algoritmos do Capítulo 3

Essa seção analisa cada um dos algoritmos estudados na Seção 3.6 para decidir se é possível descrevê-los pela Definição 4.2. Quando isso é possível, o problema será reescrito conforme a definição formal proposta, do contrário, será discutido o porquê da formalização proposta não ser adequada para descrever o algoritmo em questão.

Como esses algoritmos tem como entrada apenas um grafo, para descrever o problema resolvido por cada um desses algoritmos pela Definição 4.2, basta escolher adequadamente uma função objetivo f e uma função r que represente um conjunto de restrições que serão fixos. Assim, o problema resolvido por um algoritmo pode ser reduzido ao problema de clusterização em grafos associando r e f a cada grafo G que é uma possível entrada do algoritmo.

4.5.1 Clusterização em grafos baseada em modularidade

O algoritmo proposto por Newman (2003a) visa maximizar a modularidade. Por isso, a função objetivo pode ser definida conforme a adaptação já analisada na Seção 4.4, fazendo:

$$f(C) = -Q(C).$$

Note, ainda que, como em princípio não há nenhuma restrição na solução de clusterização natural, todas as soluções de $Clust(G)$ são consideradas válidas, logo:

$$r(C) = verdadeiro.$$

4.5.2 Clusterização em grafos baseada em expansão

Flake, Tarjan e Tsioutsiouliklis (2004), por outro lado, desenvolveram um algoritmo que otimiza a expansão. Portanto, a função objetivo pode ser definida conforme a adaptação já analisada na Seção 4.4, fazendo:

$$f(C) = -\psi(C).$$

No entanto, essa otimização não é deixada irrestrita. O problema de clusterização em grafos baseada em expansão tem como instância, além de um grafo, um valor α que é um limite inferior para a expansão da solução de clusterização e um limite superior para

$$\frac{\omega(C_i, V \backslash C_i)}{|V \backslash C_i|} < \alpha.$$

Isso pode ser englobado no problema de clusterização em grafos como uma restrição. Dessa forma, para cada $C_i \in C$ tem-se

$$r(C) = \begin{cases} verdadeiro, & \text{se } \psi(C) > \alpha \wedge \left(\forall C_i \in C, \frac{\omega(C_i, V \backslash C_i)}{|V \backslash C_i|} < \alpha \right) \\ falso, & \text{caso contrário} \end{cases}$$

4.5.3 *Distance-k clique*

Em Edachery et al. (1999) o problema de clusterização em grafos é traduzido na partição do conjunto de vértices no menor número de *distance-k cliques*. Disso, decorre que a descrição desse problema pela Definição 4.2 é dada fazendo:

$$f(C) = |C|$$

e

$$r(C) = \begin{cases} verdadeiro, & \text{se } \forall C_i \in C, C_i \text{ é uma } \textit{distance-}k \textit{ clique} \\ falso, & \text{caso contrário} \end{cases}$$

4.5.4 Algoritmos baseados em centralidade da informação e em *betweenness*

Nos algoritmos descritos nas Seções 3.6.2.2 e 3.6.2.3, não está claro qual é exatamente a propriedade desejável da solução de clusterização em grafos natural. Os autores definiram apenas dois critérios diferentes para escolha de qual aresta remover em um procedimento divisivo. Por isso, estes algoritmos não puderam ser

abrangidos pela Definição 4.2. No entanto, estudos posteriores podem encontrar alguma forma de resolver este problema.

4.5.5 Algoritmo *Highly Connected Subgraphs*

Hartuv e Shamir (2000) definem que as comunidades em um grafo G são subgrafos de G altamente conectados. A primeira providência para descrever esse problema pela Definição 4.2 é considerar que uma comunidade é o conjunto de vértices que induz um subgrafo altamente conectado.

Note que, pela forma como o algoritmo é construído, um cluster C_i só é considerado uma comunidade se ele é altamente conectado e se não existe $S \subseteq V(G)$ tal que $G[C_i]$ é subgrafo de $G[S]$ e $G[S]$ é altamente conectado. Ou seja, não existe $S \subseteq V(G)$ tal que $C_i \subset S$ e $G[S]$ é altamente conectado. Por isso, o conjunto de restrições pode ser definido como

$$r(C) = \begin{cases} verdadeiro, & \text{se } \forall C_i \in C, G[C_i] \text{ é altamente conectado } \wedge \\ & (\forall S \subseteq V(G), C_i \not\subset S \vee G[S] \text{ não é altamente conectado}) \\ falso, & \text{caso contrário} \end{cases}$$

Note que não há nenhuma preferência dentre as soluções de clusterização com essas características, por isso, a única função objetivo que faz algum sentido é uma função constante, como por exemplo,

$$f(C) = 1.$$

4.5.6 Clusterização em grafos baseada em k-objetos

Matula (apud DONGEN, 2000) considera que as noções de k-*bond*, k-componente e k-bloco podem ser usadas para definir uma comunidade. Algoritmos hierárquicos poderiam ser obtidos variando-se o k e, em cada nível, considerando que os clusteres são os k-objetos juntamente com os clusteres unitários formados por cada um dos vértices não assinalados a nenhum k-objeto.

Novamente, a primeira providência para descrever esse problema pela Definição 4.2 é considerar que uma comunidade é um conjunto unitário ou um conjunto de vértices que induz um k-objeto.

Note que esse algoritmo não é particional, pois um vértice pode pertencer a mais de um k-objeto. Por isso esse algoritmo deve ser adaptado para ser descrito pela definição formal proposta para o problema de clusterização em grafos. Uma adaptação possível é considerar que uma solução de clusterização em grafos é natural se ela é constituída apenas por k-objetos e clusteres unitários. Podem ser definidas três conjuntos de restrições conforme o tipo de k-objeto utilizado:

- Se o k-objeto for um k-*bond*

$$r(C) = \begin{cases} verdadeiro, & \text{se } \forall C_i \in C, |C_i| = 1 \vee G[C_i] \text{ é um } k\text{-bond} \\ falso, & \text{caso contrário} \end{cases}$$

- Se o k-objeto for um k-componente

$$r(C) = \begin{cases} verdadeiro, & \text{se } \forall C_i \in C, |C_i| = 1 \vee G[C_i] \text{ é um } k\text{-componente} \\ falso, & \text{caso contrário} \end{cases}$$

- Se o k-objeto for um k-bloco

$$r(C) = \begin{cases} verdadeiro, & \text{se } \forall C_i \in C, |C_i| = 1 \vee G[C_i] \text{ é um } k\text{-bloco} \\ falso, & \text{caso contrário} \end{cases}$$

Como não há nenhuma preferência dentre as soluções de clusterização naturais, a única função objetivo que faz algum sentido é uma função constante, como por exemplo,

$$f(C) = 1.$$

Observe que essa definição permite que uma solução composta apenas por conjuntos unitários seja considerada natural, mesmo que o grafo contenha k-objetos. Outra possível adaptação desse algoritmo é considerar que uma solução de clusterização em grafos é natural se ela contém o menor número de clusteres dentre todas as soluções constituídas apenas por k-objetos e clusteres unitários.

Note que nessa nova adaptação os conjuntos de restrições são os mesmos. Entretanto, a função objetivo se tornaria

$$f(C) = |C|.$$

Essa nova definição não permite mais que uma solução composta apenas por conjuntos unitários seja considerada natural, mesmo que o grafo contenha k-objetos. Afinal, se algum k-objeto puder ser formado através da união de clusteres unitários pertencentes a solução C, então existe ao menos uma tal que $f(C') < f(C)$ e, portanto, C' não seria considerada uma solução natural.

4.5.7 Clusterização em grafos baseada em circuitos elétricos

Wu e Huberman (2004) propôs um método para encontrar comunidades baseado em circuitos elétricos. Na forma como esse trabalho foi proposto, não é possível inferir qual é o problema que o autor desejava resolver ou alguma característica que é comum a soluções de clusterização de vários grafos. Por isso, não é possível abranger esse algoritmo na Definição 4.2.

4.6 Discussões

Esse capítulo propôs uma formalização para o problema de clusterização em grafos. Nessa formalização, o problema de clusterização em grafos é definido como o problema de encontrar uma solução de clusterização em grafos que satisfaça um conjunto de restrições e minimize uma função objetivo. Foi observado que a função objetivo e o conjunto de restrições podem ser definidos com base nas medidas estudadas. Especificamente, foi analisado como as medidas de qualidade podem ser adaptadas para tornarem-se funções objetivo e como utilizar as e as medidas de centralidade para indicar restrições na solução encontrada.

Com a finalidade de analisar a robustez da formalização proposta para definir o problema de clusterização em grafos, tentou-se reescrever todos os problemas resolvidos pelos algoritmos apresentados no capítulo anterior. Entretanto, isso não foi possível, pois nem sempre o autor deixa claro qual o problema que o algoritmo resolve. Por isso, foi constatado que essa definição não é robusta o suficiente para descrever todos os algoritmos existentes de clusterização em grafos. No entanto, den-

tre os algoritmos estudados, observou-se que a definição é robusta o suficiente para abranger todos os algoritmos para os quais é possível identificar qual é o problema resolvido.

5 SOLUÇÃO PARA O PROBLEMA DE CLUSTERIZAÇÃO EM GRAFOS

Esse capítulo propõe um mecanismo exaustivo para resolver o problema de clusterização em grafos, isto é, um algoritmo que examina todas as soluções de clusterização em grafo possíveis para o grafo de entrada. Quando há mais informações sobre as características do conjunto de restrições e da função objetivo, no entanto, é possível que a solução geral possa ser melhorada, de forma a reduzir o espaço de busca, isto é, a quantidade de soluções de clusterização em grafos inspecionadas.

Para averiguar essa possibilidade, inicialmente, são estudadas as características do conjunto das possíveis soluções de clusterização em grafos. Essas propriedades serão a base a partir da qual as características da função objetivo e do conjunto de restrições serão analisadas. Quatro maneiras diferentes da combinação de restrições e função objetivo são propostas para permitir que nem todas soluções de clusterização sejam inspecionadas e ainda assim haja a garantia de encontrar ao menos uma das respostas corretas para cada instância do problema de clusterização.

O restante desse capítulo está descrito da seguinte forma:

- A Seção 5.1 propõe uma solução geral para o problema de clusterização em grafos.

- A Seção 5.2 descreve algumas propriedades do conjunto de soluções de clusterização em grafos $Clust(G)$.

- A Seção 5.3 analisa as características do conjunto de restrições e da função objetivo que permitem reduzir o espaço de busca do problema de clusterização em grafos.

- A Seção 5.4 apresenta algumas discussões e considerações finais do capítulo.

5.1 Solução geral para o problema de clusterização em grafos

A solução geral do problema de clusterização em grafos é um algoritmo que resolve o problema independentemente de informações adicionais sobre as características da função objetivo f, do conjunto de restrições r ou do grafo G. Nesse caso, a única solução possível é fazer uma busca exaustiva no conjunto de soluções de clusterização. Isto é, examinar todos os elementos do conjunto de soluções de clusterização em grafos para o grafo de entrada e encontrar a resposta. O Algoritmo 5.1 mostra essa solução.

ALGORITMO 5.1 — Solução geral do problema de clusterização em grafos

ENTRADA: Um grafo $G = (V, E, \omega)$, uma função objetivo f e um conjunto de restrições r
SAÍDA: Uma solução de clusterização $C_{min} \in Clust(G)$ que satisfaz r e minimiza f

```
1: C_min ← uma solução arbitrária ∈ Clust(G)
2: PARA CADA C ∈ Clust(G) FAÇA:
3:    SE r(C) = verdadeiro ENTÃO,
4:       SE f(C) < f(C_min) ENTÃO,
5:          C_min ← C
6:       FIM
7:    FIM
8: FIM
```

Quando há mais informações sobre as características de r e f, é possível que o Algoritmo 5.1 possa ser melhorado para reduzir o espaço de busca, isto é, para que nem todas as possíveis soluções de clusterização em grafos de G tenham que ser inspecionadas. Note que essa modificação não se refere à utilização de heurísticas ou de outros algoritmos aproximados, mas a combinações das características da função objetivo e do conjunto de restrições que permitam que uma parte do espaço de busca seja efetivamente descartada e a resposta correta seja encontrada. Em outras palavras, para que, em se identificando que as características da função objetivo e do conjunto de restrições obedecem um certo padrão, seja possível identificar, para cada instância I do problema, um subconjunto $S(I) \in Clust(G)$ tal que pelo menos

uma das respostas $R \in \Phi_{ClustGrafos}(I)$ esteja contido em no subconjunto $S(I)$, ou seja $\Phi_{ClustGrafos}(I) \cap S(I) \neq \emptyset$.

Antes de discutir as características da função objetivo e do conjunto de restrições que permitem a redução do espaço de busca do problema, entretanto, é necessário entender algumas propriedades do conjunto $Clust(G)$. Essas propriedades serão a base a partir da qual as características da função objetivo e do conjunto de restrições serão analisadas. A próxima seção discute algumas propriedades de $Clust(G)$.

5.2 Propriedades do conjunto $Clust(G)$

Segundo a definição, $Clust(G)$ é o conjunto de todas as soluções de clusterização em grafos de G que são exclusivas, completas e que induzem subgrafos conexos em G. Isto é, $Clust(G)$ é um subconjunto do conjunto de partições de $V(G)$ tal que, para cada $C \in Clust(G)$, tem-se

$$\forall C_i \in C, G[C_i] \text{ é conexo.}$$

No caso particular de grafos completos, tem-se que para todo $S \subseteq V(G)$, $G[S]$ é conexo. Portanto $Clust(G)$, nesse caso, é exatamente igual ao conjunto de partições de $V(G)$. Devido a essa particularidade, antes de analisar as propriedades de $Clust(G)$ para um grafo G qualquer, é interessante estudar as características específicas de $Clust(G)$ para um grafo G completo.

5.2.1 Análise das características de $Clust(G)$ para um grafo G completo

Quando G é um grafo completo, o conjunto $Clust(G)$ é exatamente igual ao conjunto de partições de $V(G)$. Por isso, as propriedades de $Clust(G)$ para qualquer grafo completo G podem ser analisadas diretamente através das propriedades do conjunto de partições de um conjunto qualquer.

Considere um conjunto D. Uma partição de D é um conjunto X de subconjuntos não vazios de D, chamados blocos, tal que cada elemento de D pertence a um e somente um bloco $X_i \in X$. Ou seja, se X é uma partição de D, então tem-se que

$$\forall X_i \in X, X_i \subseteq D \wedge X_i \neq \emptyset,$$

$$\forall X_i, X_j \in X \wedge i \neq j, X_i \cap X_j = \emptyset,$$

e

$$D = \bigcup_{X_i \in X} X_i.$$

Denote o conjunto de partições de D por $Part(D)$. Defina a relação $Refinamento \subseteq Part(D) \times Part(D)$ tal que $(X, Y) \in Refinamento$ se e somente se cada bloco de X é subconjunto de algum bloco de Y, isto é se

$$\forall X_i \in X, (\exists Y_i \in Y, X_i \subseteq Y_i).$$

Nesse caso, a partição X é dita um refinamento da partição Y. Dadas duas partições Z e W, tal que $(Z, W) \in Refinamento$, diz-se que W cobre Z se

$$\nexists U \in Part(D) \backslash \{Z, W\}, (Z, U) \in Refinamento \wedge (U, W) \in Refinamento.$$

TEOREMA 5.2: *A relação $Refinamento$ é uma relação de ordem no conjunto $Part(D)$.*

Demonstração:

Para provar que a relação $Refinamento$ é uma relação de ordem basta mostrar que $Refinamento$ é uma relação (i) reflexiva, (ii) transitiva e (iii) antissimétrica (LIPSCHUTZ, 2004).

(i) Se $Refinamento$ é uma relação reflexiva, então

$$\forall X \in Part(D), (X, X) \in Refinamento.$$

Da definição de $Refinamento$ tem-se que $(X, X) \in Refinamento$ se, para cada bloco $X_i \in X$, existe um bloco $X_j \in X$ tal que $X_i \subseteq X_j$.

Suponha a existência de um bloco $X_i \in X$ tal que X_i não seja subconjunto de nenhum bloco de X. Mas $X_i \subseteq X_i$, o que configura um absurdo. Logo, $Refinamento$ é uma relação reflexiva.

(ii) Se $Refinamento$ é uma relação transitiva, então

$$(X, Y) \in Refinamento \wedge (Y, Z) \in Refinamento \Longrightarrow (X, Z) \in Refinamento.$$

Suponha que $(X, Y) \in Refinamento$ e $(Y, Z) \in Refinamento$. Considere um bloco $X_i \in X$. Como $(X, Y) \in Refinamento$, existe um bloco $Y_i \in Y$ tal que $X_i \subseteq Y_i$. Analogamente, como $(Y, Z) \in Refinamento$, então também existe um bloco $Z_i \in Z$ tal que $Y_i \subseteq Z_i$. Como a relação estar contido é uma relação transitiva, então $X_i \subseteq Z_i$. Logo, $(X, Z) \in Refinamento$ e, portanto $Refinamento$ é uma relação transitiva.

(iii) Se $Refinamento$ é uma relação antissimétrica, então

$$(X, Y) \in Refinamento \wedge (Y, X) \in Refinamento \Longrightarrow X = Y.$$

Assuma que existam duas partições $X \neq Y$ tal que $(X, Y) \in Refinamento$ e $(Y, X) \in Refinamento$. Como $(X, Y) \in Refinamento$, para todo bloco $X_i \in X$, existe um bloco $Y_i \in Y$ tal que $X_i \subseteq Y_i$. Analogamente, como $(Y, X) \in Refinamento$, então existe um bloco $X_j \in X$ tal que $Y_j \subseteq X_j$.

Como os blocos são subconjuntos não vazios de D, considere um elemento $x \in Xi$, como X_i é subconjunto de Y_i, então $x \in Y_i$. Da mesma forma, como Y_i é subconjunto de X_j, então $x \in X_j$. Como X é uma partição, se x pertence a X_i e a X_j, então $X_i = X_j$. Logo, tem-se que $X_i \subseteq Y_i \subseteq X_i$. Como a relação estar contido é uma relação antissimétrica, então $Y_i = X_i$. Aplicando essa propriedade a todos os blocos de X, tem-se que $X = Y$. Isso contraria a hipótese inicial. Logo $Refinamento$ é uma relação antissimétrica.

Logo, a relação $Refinamento$ é uma relação de ordem no conjunto $Part(D)$.

□

Como $Refinamento$ é uma relação de ordem então $(Part(D), Refinamento)$ é dito um conjunto parcialmente ordenado (LIPSCHUTZ, 2004). De uma forma mais simples, o conjunto $Part(D)$ também é chamado um conjunto parcialmente ordenado. Para simplificar a notação, $(X, Y) \in Refinamento$ será representado simplesmente por $X \preceq Y$.

Se um conjunto é parcialmente ordenado, ele é linearmente ordenado se, dados dois elementos X e Y do conjunto, então ou $X \preceq Y$ ou $Y \preceq X$ (LIPSCHUTZ, 2004). Nesse caso X e Y são ditas comparáveis, caso contrário, X e Y são ditas incomparáveis.

Note que o conjunto $Part(D)$ não é linearmente ordenado. Isso pode ser ilustrado através de um exemplo simples. Considere o conjunto $\{1, 2, 3, 4\}$ e duas partições $\{\{1, 2, 3\}, \{4\}\}$ e $\{\{1, 3, 4\}, \{2\}\}$ desse conjunto. Observe que essas duas soluções não são associadas pela relação $Refinamento$.

Os conjuntos parcialmente ordenados podem ser representados por um diagrama de Hasse (LIPSCHUTZ, 2004). No diagrama de Hasse o conjunto parcialmente ordenado é representado por um grafo, onde

- O conjunto de vértices é o conjunto parcialmente ordenado $Part(D)$.
- Há aresta entre duas partições X e Y se e somente se X cobre Y ou Y cobre X.

No diagrama de Hasse, esse grafo é desenhado de forma que os vértices são dispostos em níveis e, se $X \preceq Y$ então Y está situado em um nível superior ao nível de Y. Nesse diagrama, é simples averiguar se duas partições estão relacionadas pela relação de ordem, pois $Z \preceq W$ se e somente se existe caminho ascendente entre Z e W. A Figura 5.3 ilustra o diagrama de Hasse para o conjunto de partições de $\{1, 2, 3, 4\}$.

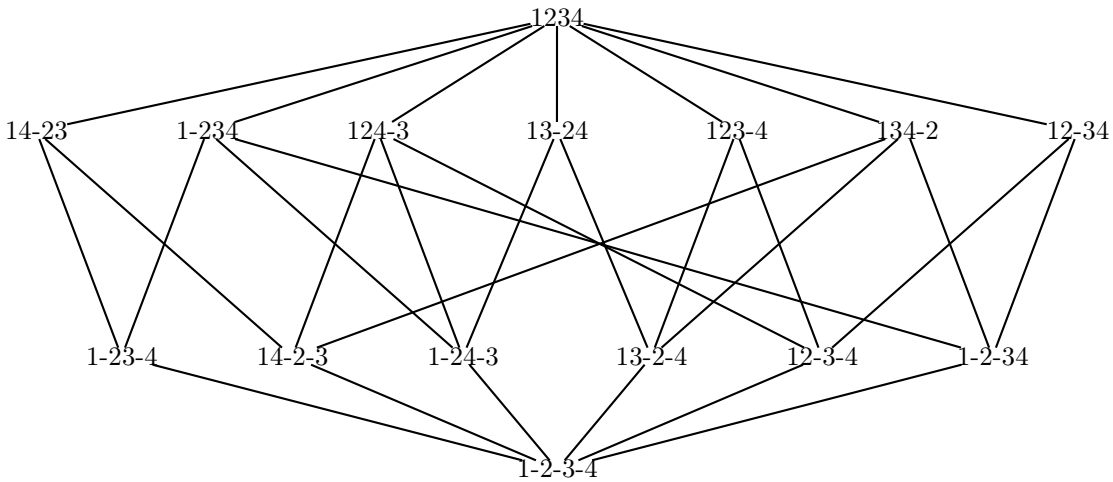

FIGURA 5.3 — Diagrama de Hasse que representa o conjunto de partições de $\{1, 2, 3, 4\}$. Cada partição é representada de forma que os clusteres são separados por hifens.

Observe que existe exatamente um elemento máximo e um elemento mínimo em $Part(D)$. Por conveniência, as partições que são elemento máximo e mínimo em $Part(D)$ receberão nomes especiais. A partição composta por um único bloco igual ao conjunto D é chamada partição raiz e será representada por $Raiz = \{D\}$. Analogamente, a partição composta por $|D|$ conjuntos unitários é chamada partição folha e é simbolizada por $Folhas = \{\{d\}|d \in D\}$.

Como o elemento X é mínimo se $X \preceq Z$ para todo Z no conjunto parcialmente ordenado, para provar que a partição $Folhas$ é mínimo em $Part(D)$ basta mostrar que $Folhas \preceq Z$ para todo $Z \in Part(D)$. Analogamente, como o elemento Y é máximo se $Z \preceq Y$ para todo Z no conjunto parcialmente ordenado, para provar que $Raiz$ é máximo em $Part(D)$ basta mostrar que $Z \preceq Raiz$ para todo $Z \in Part(D)$.

TEOREMA 5.4: *Todos os elementos de $Part(D)$ são refinamentos da partição $Raiz$.*

Demonstração:

A prova desse teorema será realizada por indução finita. Como qualquer partição em $Part(D)$ tem pelo menos um bloco, basta mostrar que: (base) todas as partições com um bloco são refinamentos de $Raiz$; e (passo indutivo) assumindo todas as partições com k blocos são refinamentos de $Raiz$, então todas as partições com $k + 1$ blocos também são refinamentos de $Raiz$.

Base: Da definição de partição, tem-se que a única partição com um bloco é a partição $Raiz$. Como $Refinamento$ é uma relação reflexiva, então, $Raiz \preceq Raiz$. Logo, todas as soluções com um cluster são refinamentos da solução $Raiz$.

Passo indutivo: Suponha que todas as partições com k blocos, $k \geqslant 1$, sejam um refinamento de $Raiz$.

Considere uma partição X com $k + 1$ blocos e dois blocos quaisquer X_i e X_j nessa partição. Una X_i e X_j formando o bloco

$$X_{i+j} = X_i \cup X_j.$$

Construa uma nova partição

$$X' = (X \backslash \{X_i, X_j\}) \cup X_{i+j}.$$

Observe que X' tem k blocos. Note também que, devido a forma como X' é construída, $X \preceq X'$. Por hipótese, $X' \preceq Raiz$. Como $Refinamento$ é uma relação transitiva, então $X \preceq Raiz$.

Logo, todos os elementos de $Part(D)$ são refinamentos da partição $Raiz$.

□

TEOREMA 5.5: *A partição $Folhas$ é refinamento de todos os elementos de $Part(D)$.*

Demonstração:

A prova desse teorema será realizada por indução finita. Como qualquer partição em $Part(D)$ tem no máximo $|D|$ blocos, basta mostrar que: (base) $Folhas$ é um refinamento de todas as partições com $|D|$ blocos; e (passo indutivo) assumindo $Folhas$ seja um refinamento de todas as partições com k blocos, então $Folhas$ também é um refinamento de todas as partições com $k-1$ blocos.

Base: Da definição de partição, tem-se que a única partição com $|D|$ blocos é a partição $Folhas$. Como $Refinamento$ é uma relação reflexiva, então, $Folha \preceq Folhas$. Logo, $Folhas$ é refinamento de todas as soluções com $|D|$ blocos.

Passo indutivo: Suponha que $Folhas$ seja refinamento de todas as partições com k blocos, $k \leqslant |D|$.

Considere uma partição X com $k-1$ blocos. Observe que existe pelo menos um bloco $X_i \in X$ formado por mais de um elemento, isto é, tal que $|X_i| \geqslant 2$. Escolha um elemento $d \in X_i$. Como $|X_i| \geqslant 2$ então $X_i \backslash \{d\}$ é um subconjunto não vazio de D. Construa uma partição

$$X' = (X \backslash X_i) \cup \{\{d\}, X_i \backslash \{d\}\}.$$

Observe que X' tem k blocos. Por hipótese, $Folhas \preceq X'$. Como $Refinamenro$ é uma relação transitiva e $X' \preceq X$, então $Folhas \preceq X'$.

Logo, a partição $Folhas$ é refinamento de todos os elementos de $Part(D)$.

□

Observe ainda, que os elementos de $Part(D)$ são distribuídos no diagrama de Hasse de forma que em cada nível a partição tem exatamente um bloco a mais que as partições no nível imediatamente superior e existem arestas apenas entre vértices em níveis consecutivos. Isso é decorrente do fato que uma partição é coberta apenas por partições situadas nos níveis imediatamente superiores. Para mostrar que essa distribuição é aplicável a $Part(D)$, qualquer que seja o conjunto D, basta provar que uma partição X^k cobre apenas elementos X^{k+1} com exatamente um bloco a mais que X^k. Isto é, basta mostrar que (i) toda partição com $k+1$ blocos é refinamento de pelo menos uma partição com k blocos e que (ii) se uma partição X^l com l blocos é refinamento de uma partição X^k com k blocos, $l > k+1$, então existe uma partição X^{k+1} com $k+1$ blocos, tal que $X^l \preceq X^{k+1} \preceq X^k$.

TEOREMA 5.6: *Toda partição com $k+1$ blocos, $k \geqslant 1$, é refinamento de pelo menos uma partição X^k com k blocos.*

Demonstração:

Considere uma partição X com $k+1$ blocos. Como $k \geqslant 1$, então existem pelo menos duas partições em X. Considere dois blocos X_i e X_j em X. Una X_i e X_j formando o bloco

$$X_{i+j} = X_i \cup X_j.$$

Construa a partição

$$X' = (X \backslash \{X_i, X_j\}) \cup \{X_{i+j}\}.$$

Observe que X' tem k blocos. Portanto, qualquer partição com $k+1$ blocos pode ser obtida pelo refinamento de alguma partição com k blocos.

□

TEOREMA 5.7: *Se uma partição X^l com l blocos é refinamento de uma partição X^k com k blocos, $l > k+1$, então existe uma partição X^{k+1} com $k+1$ blocos, tal que $X^l \preceq X^{k+1} \preceq X^k$.*

Demonstração:

A prova desse teorema será realizada por indução finita. Basta mostrar que: (base) se uma partição X^{k+2} com $k+2$ blocos é refinamento de uma partição X^k com k blocos, então existe uma partição X^{k+1} com $k+1$ blocos tal que $X^{k+2} \preceq X^{k+1} \preceq X^k$; e (passo indutivo) assumindo que para toda partição X^l com l blocos que é refinamento de uma partição X^k com k blocos exista uma partição X^{k+1} com $k+1$ blocos, tal que $X^l \preceq X^{k+1} \preceq X^k$, então para todas as

partições X^{l+1} com $l+1$ blocos que são refinamentos de uma partição X^k com k blocos existe uma partição X^{k+1} com $k+1$ blocos, tal que $X^{l+1} \preceq X^{k+1} \preceq X^k$.

Base: Considere uma partição X^{k+2} com $k+2$ blocos que seja refinamento de X^k. Considere um bloco $X_i^k \in X^k$ tal que $X_i^k \notin X^{k+2}$. Note que existem pelo menos dois blocos de X^{k+2}, X_i^{k+2} e X_j^{k+2} que são subconjuntos de X_i^k. Una X_i^{k+2} e X_j^{k+2} formando um novo bloco

$$X_{i+j}^{k+2} = X_i^{k+2} \cup X_j^{k+2}.$$

Construa uma nova partição

$$X^{k+1} = (X^{k+2} \backslash \{X_i^{k+2}, X_j^{k+2}\}) \cup \{X_{i+j}^{k+2}\}.$$

Observe que X_{k+1} tem $k+1$ blocos e que $X_{k+2} \preceq X_{k+1}$. Note que, devido à maneira que X_i^{k+2} e X_j^{k+2} foram escolhidos, $X_{k+1} \preceq X_k$. Logo, para todas as partições X^{k+2} com $k+2$ blocos que são refinamentos de uma partição X^k com k blocos existe uma partição X^{k+1} com $k+1$ blocos, tal que X^{k+1} é um refinamento de X^k e X^{k+2} é um refinamento de X^{k+1}.

Passo indutivo: Assuma para toda partição X^l com l blocos que é refinamento de uma partição X^k com k blocos existe uma partição X^{k+1} com $k+1$ blocos, tal que $X^l \preceq X^{k+1} \preceq X^k$.

Considere uma partição X^{l+1} com $l+1$ blocos que é refinamento de X^k com k blocos. Considere uma partição $X_i^k \in X^k$ tal que $X_i^k \notin X^{l+1}$. Note que existem pelo menos dois blocos de X^{l+1}, X_i^{l+1} e X_j^{l+1} que são subconjuntos de X_i^k. Una os blocos X_i^{l+1} e X_j^{l+1} formando um novo bloco

$$X_{i+j}^{l+1} = X_i^{l+1} \cup X_j^{l+1}.$$

Construa uma nova partição

$$X^l = (X^{l+1} \backslash \{X_i^{l+1}, X_j^{l+1}\}) \cup \{X_{i+j}^{l+1}\}.$$

Observe que X_l tem l blocos e que $X_{l+1} \preceq X_l$. Note que, devido à maneira que X_i^{l+1} e X_j^{l+1} foram escolhidos, $X_l \preceq X_k$.

Por hipótese, existe uma partição X^{k+1} com $k+1$ blocos, tal que $X^l \preceq X^{k+1} \preceq X^k$. Como $Refinamento$ é uma relação transitiva, então $X^{l+1} \preceq X_{k+1}$. Logo, para todas as partições X^{l+1} com $l+1$ blocos que são refinamentos de uma partição X^k com k blocos existe uma partição X^{k+1} com $k+1$ blocos, tal que $X^{l+1} \preceq X^{k+1} \preceq X^k$.

□

5.2.2 Análise das características de $Clust(G)$ para um grafo G conexo

Quando o grafo G não é completo, algumas partições de $V(G)$ apresentam clusteres que não induzem grafos conexos em G. Por isso, nem todas as partições de $V(G)$ pertencem ao conjunto $Clust(G)$.

Considere, por exemplo, o grafo G construído de forma que $V(G) = \{1, 2, 3, 4\}$ e $E(G) = \{\{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}\}$. A Figura 5.8 mostra as soluções de clusterização $C \in Clust(G)$ dispostas em um diagrama de Hasse. Note que todas as partições que induze grafos desconexos em G e, portanto, não são soluções de clusterização em grafos foram propositalmente omitidas do diagrama.

Será que é sempre possível representar todas as possíveis soluções de clus-

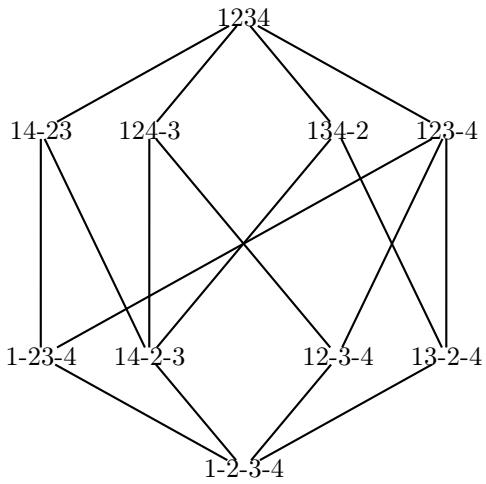

FIGURA 5.8 — Diagrama de Hasse que representa $Clust(G)$ para o grafo definido por $V(G) = \{1, 2, 3, 4\}$ e $E(G) = \{\{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}\}$. Cada partição é representada de forma que os clusteres são separados por hifens.

terização $C \in Clust(G)$ em um diagrama de Hasse? Note que, como $Clust(G)$ é um subconjunto de $Part(V(G))$, então $Clust(G)$ também é um conjunto parcialmente ordenado. Logo, o diagrama de Hasse representa todas as possíveis soluções de clusterização $C \in Clust(G)$.

Note que, assim como o conjunto $Part(V(G))$, $Clust(G)$ também possui um elemento máximo e um elemento mínimo e todas as arestas estão situadas entre níveis consecutivos no diagrama. Para demonstrar essas propriedades, os Teoremas 5.4 a 5.7 devem ser adaptados para referir-se ao conjunto $Clust(G)$.

Antes de provar os teoremas adaptados, é importante observar que as soluções de clusterização em grafos correspondente às partições $Folhas$ e $Raiz$ definidas anteriormente pertencem a $Clust(G)$. Isso ocorre devido ao fato de G ser assumido ser sempre conexo.

TEOREMA 5.9: *Todas as soluções de clusterização $C \in Clust(G)$ são refinamentos de $Raiz$.*

Demonstração:

Pelo Teorema 5.4, todos os elementos de $Part(V(G))$ são refinamentos de $Raiz$. Como $Clust(G) \subseteq Part(V(G))$, então todos os elementos de $Clust(G)$ também são refinamentos de $Raiz$.

□

TEOREMA 5.10: *$Folhas$ é um refinamento de todas as soluções de clusterização $C \in Clust(G)$.*

Demonstração:

Pelo Teorema 5.5, $Folhas$ é refinamento de todos os elementos de $Part(V(G))$. Como $Clust(G) \subseteq Part(V(G))$, então $Folhas$ também é refinamento de todos os elementos de $Clust(G)$.

□

TEOREMA 5.11: *Qualquer solução de clusterização com k clusteres, $k \geqslant 1$, pode ser obtida pelo refinamento de alguma solução de clusterização com $k-1$ clusteres.*

Demonstração:

Considere uma solução de clusterização C^k com k clusteres, $k \geqslant 2$. Considere um cluster qualquer $C_i^k \in C^k$. Como G, por definição, é conexo, então existe $C_j^k \in C^k$, $C_i^k \neq C_j^k$, tal que $E'(C_i^k, C_j^k) \neq \emptyset$.

Logo, $C_{i+j}^k = C_i^k \cup C_j^k$ induz um grafo conexo em G e por isso é um cluster em G. Considere, agora, a solução de clusterização construída de forma que

$$C^{k-1} = \left(C^k \backslash \{C_i^k, C_j^k\}\right) \cup \{C_{i+j}^k.$$

Note que C^{k-1} é uma solução de clusterização com $k-1$ clusteres e $C^k \preceq C^{k-1}$. Portanto, qualquer solução de clusterização com k clusteres, $k > 1$, pode ser obtida pelo refinamento de alguma solução de clusterização com $k-1$ clusteres.

□

TEOREMA 5.12: *Se uma solução de clusterização C^l com l clusteres é refinamento de uma solução de clusterização C^k com k clusteres, $l > k+1$, então existe uma solução de clusterização C^{k+1} com $k+1$ clusteres, tal que $C^l \preceq C^{k+1} \preceq C^k$.*

Demonstração:

A prova desse teorema será realizada por indução finita. Basta mostrar que: (base) se uma solução de clusterização C^{k+2} com $k+2$ clusteres é refinamento de uma solução de clusterização C^k com k clusteres, então existe uma solução C^{k+1} com $k+1$ clusteres tal que $C^{k+2} \preceq C^{k+1} \preceq C^k$; e (passo indutivo) assumindo que para toda solução de clusterização X^l com l clusteres, $l \geqslant k+2$, que é refinamento de uma solução de clusterização C^k com k clusteres exista uma solução C^{k+1} com $k+1$ clusteres, tal que $C^l \preceq C^{k+1} \preceq C^k$, então para todas as soluções C^{l+1} com $l+1$ clusteres que são refinamentos de uma solução C^k com k clusteres existe uma solução C^{k+1} com $k+1$ clusteres, tal que $C^{l+1} \preceq C^{k+1} \preceq C^k$.

Base: Considere uma solução C^{k+2} com $k+2$ clusteres que seja refinamento de C^k. Considere um cluster $C_i^k \in C^k$ tal que $C_i^k \notin C^{k+2}$. Então, existem pelo menos dois clusteres em C^{k+2} que são subconjunto de C_i^k.

Como C_i^k induz um subgrafo conexo em G, existe pelo menos uma aresta inter-

cluster em $E'(C^{k+2})$ que é uma aresta interna do cluster C_i^k. Tome uma aresta e tal que $e \in E'(C^{k+2}) \wedge e \in E(C_i^k)$. Denote por C_i^{k+2} e C_j^{k+2} os dois clusteres para os quais $e \in E'(C_i^{k+2})$ e $e \in E'(C_j^{k+2})$.

Una C_i^{k+2} e C_j^{k+2} formando um novo cluster

$$C_{i+j}^{k+2} = C_i^{k+2} \cup C_j^{k+2}.$$

Note que C_{i+j}^{k+2} induz um subgrafo conexo em G. Construa uma nova solução

$$C^{k+1} = (C^{k+2} \backslash \{C_i^{k+2}, C_j^{k+2}\}) \cup \{C_{i+j}^{k+2}\}.$$

Observe que C_{k+1} tem $k+1$ clusteres e $C_{k+2} \preceq C_{k+1}$. Note que, devido à maneira que C_i^{k+2} e C_j^{k+2} foram escolhidos, $C^{k+1} \preceq C_k$. Logo, para todas as soluções de clusterização C^{k+2} com $k+2$ clusteres que são refinamentos de uma solução C^k com k clusteres existe uma solução C^{k+1} com $k+1$ clusteres, tal que $C^{k+2} \preceq C^{k+1} \preceq C^k$.

Passo indutivo: Assuma para toda solução C^l com l clusteres que é refinamento de uma solução C^k com k clusteres existe uma solução C^{k+1} com $k+1$ clusteres, tal que $C^l \preceq C^{k+1} \preceq C^k$.

Considere uma solução C^{l+1} com $l+1$ clusteres que é refinamento de C^k com k clusteres. Considere um cluster $C_i^k \in C^k$ tal que $C_i^k \notin C^{l+1}$. Então, existem pelo menos dois clusteres em C^{l+1} que são subconjunto de C_i^k.

Como C_i^k induz um subgrafo conexo em G, existe pelo menos uma aresta inter-cluster em $E'(C^{l+1})$ que é uma aresta interna do cluster C_i^k. Tome uma aresta e tal que $e \in E'(C^{l+1}) \wedge e \in E(C_i^k)$. Denote por C_i^{l+1} e C_j^{l+1} os dois clusteres para os quais $e \in E'(C_i^{l+1})$ e $e \in E'(C_j^{l+1})$.

Una C_i^{l+1} e C_j^{l+1} formando um novo cluster

$$C_{i+j}^{l+1} = C_i^{l+1} \cup C_j^{l+1}.$$

Construa uma nova solução

$$C^l = (C^{l+1} \backslash \{C_i^{l+1}, C_j^{l+1}\}) \cup \{C_{i+j}^{l+1}\}.$$

Observe que C_l tem l clusteres e que $C_{l+1} \preceq C_l$. Note que, devido à maneira que C_i^{l+1} e C_j^{l+1} foram escolhidos, $C_l \preceq C_k$.

Por hipótese, existe uma solução C^{k+1} com $k+1$ clusteres, tal que $C^l \preceq C^{k+1} \preceq C^k$. Como $Refinamento$ é uma relação transitiva, então $C^{l+1} \preceq C_{k+1}$. Logo, para todas as soluções C^{l+1} com $l+1$ clusteres que são refinamentos de uma solução C^k com k clusteres existe uma solução C^{k+1} com $k+1$ clusteres, tal que $C^{l+1} \preceq C^{k+1} \preceq C^k$.

□

5.2.3 Inspeção de todas as partições de $Clust(G)$ através do diagrama de Hasse que o representa

Existem diversas formas de inspecionar todos os elementos de um conjunto finito. Uma maneira simples, por exemplo, seria selecionar sucessivamente um elemento do conjunto de forma aleatória até que todos eles tenham sido selecionados. Inspecionar o conjunto com esse procedimento, entretanto, tem a desvantagem de necessitar que todo os elementos do conjunto estejam disponíveis ao mesmo tempo. Por isso, essa técnica é inadequada quando se considera um conjunto com cardinalidade

muito alta. Esse é o caso, por exemplo, o conjunto de soluções de clusterização em grafos $Clust(G)$, afinal, o número de elementos de $Clust(G)$ cresce exponencialmente com o número de vértices de G.

Seria interessante, então, ter uma maneira de inspecionar todos os elementos de um conjunto sem a necessidade de ter todo o conjunto disponível de uma só vez. Isto é, uma forma de gerar os elementos de $Clust(G)$ ao mesmo tempo em que se inspeciona os elementos, evitando o desperdício no espaço de armazenamento. No caso específico de um conjunto parcialmente ordenado, em que todos os elementos podem ser representados em um grafo conexo, outra maneira de inspecionar todos os elementos seria realizar um procedimento de busca no diagrama de Hasse que representa o conjunto.

Busca é um nome que se refere a algoritmos que examinam todos os vértices e todas as arestas de um grafo (BOSS, 2010). Os algoritmos de busca em grafos possuem como entrada um grafo e um vértice inicial nesse grafo e tem como saída uma enumeração do conjunto das arestas, que indica a ordem em que as arestas são percorridas. Como o diagrama de Hasse representa um grafo conexo, um passeio que visite todas as arestas de um grafo necessariamente contém todos os vértices do grafo.

Quando se considera o conjunto de soluções de clusterização $Clust(G)$, as arestas do diagrama de Hasse que o representa possuem um significado físico importante: duas partições estão ligadas por uma aresta se e somente se são relacionadas pela relação $Refinamento$ e se uma das partições cobre a outra. Segundo o Teorema 5.12, isso significa que uma dessas partições tem exatamente um cluster a mais ou a menos que a outra.

Disso decorre que, considerando uma partição C qualquer, é possível gerar todos os elementos de $Clust(G)$ que são ligados a C por uma aresta nos níveis imediatamente superior e inferior do diagrama de Hasse. Isso é obtido, respectivamente,

gerando todas as partições C' que cobrem C e gerando todas as partições C'' que são cobertas por C. Essa possibilidade é justamente o que torna possível gerar todos os elementos de $Clust(G)$ ao mesmo tempo em que se realiza a busca.

Especificamente no caso de $Clust(G)$, duas buscas são importantes e por isso serão chamadas pelos nomes especiais ascendente e descendente. Uma busca descendente inicia na solução $Raiz$. A cada passo, a solução é inspecionada e todos os seus vizinhos no nível imediatamente inferior do diagrama de Hasse são gerados. Isto é feito gerando todas as soluções de clusterização em grafos que são cobertas pela solução. Cada uma dessas novas soluções é marcada para ser visitada nos passos seguintes.

A escolha de qual das soluções de clusterização em grafo marcadas será selecionada para ser inspecionada nos próximos passos não é relevante para a análise que será feita nesse trabalho. No entanto, observe que essa escolha é importante em termos práticos. Considerando que seja escolhido um mecanismo de busca em largura o número de soluções marcadas para ser analisada nas próximas iterações pode ser muito grande, enquanto se for utilizado um procedimento de busca em profundidade esse número é limitado a ser no máximo $|V|$. Por isso, um mecanismo de busca em profundidade é interessante no sentido de permitir que a quantidade de soluções efetivamente armazenadas na memória em um dado momento seja relativamente pequena.

Analogamente, uma busca ascendente começa na solução de clusterização $Folhas$ e, a cada passo, gera todos os vizinhos no nível imediatamente superior do diagrama de Hasse e os marca para serem visitados nos passos seguintes.

Note que, se não houver nenhum controle de quais soluções já foram analisadas, uma mesma solução poderia ser inspecionada por várias vezes, já que uma solução pode ser vizinha de várias outras partições em níveis superiores e inferiores. Dependendo das características do conjunto de restrições e da função objetivo a ser

minimizada, entretanto, algumas partes do conjunto $Clust(G)$ poderiam ser eliminadas do procedimento de busca. Por exemplo, se houvesse a garantia que nenhum refinamento de uma certa solução fosse solução do problema de clusterização em grafos, essas soluções poderiam ser ignoradas do processo de busca.

A próxima seção analisa algumas combinações das características que podem ser utilizadas de forma a reduzir o espaço de busca do algoritmo que resolva o problema de clusterização em grafos.

5.3 Análise das características do conjunto de restrições e da função objetivo que permitem reduzir o espaço de busca do problema de clusterização em grafos

Boa parte dos algoritmos de clusterização em grafos, inclusive dos apresentados na Capítulo 3, não resolvem realmente o problema de clusterização em grafos. Isto é, não há garantia que a resposta encontrada para uma dada instância seja realmente uma solução de clusterização ótima nas propriedades que aquele algoritmo em particular pretende otimizar. Isso ocorre devido ao fato que muitos algoritmos utilizam heurísticas para, ao invés de examinar todas as possíveis soluções de clusterização para o grafo de entrada, encontrar uma solução relativamente boa em um espaço de busca menor. No entanto, como nesse trabalho, o foco principal é formalizar o problema e viabilizar a construção de algoritmos que sejam uma solução exata do problema de clusterização, heurísticas e soluções aproximadas não são de interesse.

O Algoritmo 5.1 resolve o problema de clusterização em grafos através da inspeção de todas as possíveis soluções de clusterização em $Clust(G)$. Entretanto, como o número de elementos de $Clust(G)$ cresce exponencialmente com o número de vértices de G, sua utilização é inviável para grafos com um grande número de vértices.

Por isso, apesar desse algoritmo servir ao objetivo primário desse trabalho, seria interessante estudar qual a combinação das características do conjunto de restrições e da função objetivo que permite reduzir o espaço de busca do problema de clusterização em grafos e, mesmo assim, ainda garantir encontrar uma das respostas corretas para cada instância do problema. Ou seja, analisar a combinação de restrições e função objetivo que permitem que nem todas soluções de clusterização em $Clust(G)$ sejam inspecionadas e ainda assim haja a garantia de encontrar uma das respostas corretas para cada instância do problema de clusterização.

Devido às propriedades de $Clust(G)$, é intuitivo imaginar que seria possível reduzir o espaço de busca do problema de clusterização em grafos auxiliado pelo diagrama de Hasse que o representa. Imagine, por exemplo, um conjunto de restrições construído de forma que se as restrições não são satisfeitas em uma solução elas também não são satisfeita em todos os seus refinamentos. Nesse caso, um mecanismo de busca descendente no grafo correspondente ao diagrama de Hasse poderia ser utilizado para que assim que uma solução fosse considerada inválida todos os seus refinamentos não precisassem ser inspecionados.

Por isso, é interessante analisar as características do conjunto de restrições e da função objetivo que tornam o seu comportamento interessante no sentido ascendente ou descendente do diagrama de Hasse que representa $Clust(G)$. Considere, por exemplo, uma função objetivo f_c tal que, se C_{ref} é um refinamento de C, então

$$f_c(C_{ref}) \geqslant f_c(C).$$

Nesse caso, f_c será denominada crescente no sentido dos refinamentos. Analogamente, tome, agora, uma função objetivo f_d que se comporta de maneira que, se C_{ref} é um refinamento de C, então

$$f_d(C_{ref}) \leqslant f_d(C).$$

Nessa situação, f_d será dita decrescente no sentido dos refinamentos.

Note que $Folhas$ terá valor mínimo de f_c dentre todas as partições do conjunto $Clust(G)$ e que $Raiz$, por sua vez, tem valor mínimo de f_d em $Clust(G)$. Por isso, a minimização irrestrita de uma função crescente ou decrescente no sentido dos refinamentos leva às soluções triviais.

Considere, agora, um conjunto de restrições r_c tal que, se C_{ref} é um refinamento de C, tem-se

$$(r_c(C_{ref}) = falso) \Longrightarrow (r_c(C) = falso)\,.$$

Um conjunto de restrições que satisfaça essas propriedades será denominado restritivo no sentido contrário ao dos refinamentos. Alternativamente, considere um conjunto de restrições r_d que se comporta de maneira que, se C_{ref} é um refinamento de C, então

$$(r_d(C) = falso) \Longrightarrow (r_d(C_{ref}) = falso)\,.$$

Um conjunto de restrições que tenha esse comportamento será dito restritivo no sentido dos refinamentos.

Se um conjunto de restrições restritivo no sentido dos refinamentos é $verdadeiro$ em $Folhas$, então ele é válido em todos os elementos de $Clust(G)$. Da mesma forma, se um conjunto de restrições restritivo no sentido contrário ao dos refinamentos é $verdadeiro$ em $Raiz$, então ele também é válido em todos os elementos de $Clust(G)$.

Casos mais interessantes, e que permitem uma redução do espaço de busca, são os casos em que o conjunto de restrições é válido apenas para parte do conjunto de restrições. Quatro combinações entre as características do conjunto de restrições e da função objetivo parecem ser interessante: (i) conjunto de restrições restritivo no sentido dos refinamentos e função objetivo qualquer; (ii) conjunto de restrições restritivo no sentido contrário ao dos refinamentos e função objetivo qualquer; (iii) conjunto

de restrições restritivo no sentido dos refinamentos e função objetivo decrescente no sentido dos refinamentos; (iv) conjunto de restrições restritivo no sentido dos refinamentos e função objetivo crescente no sentido dos refinamentos. O restante da seção analisa essas quatro situações.

(i) Se o conjunto de restrições r é restritivo no sentido dos refinamentos, então se r é falso para uma solução, todos os seus refinamentos também o serão. Por isso, a inspeção deve ser feita no sentido descendente. A única adaptação necessária para utilizar o mecanismo geral de busca descendente é que a cada passo, deverá ser analisado se o conjunto de restrições é válido. Se o conjunto de restrições é falso na partição, então todos os refinamentos da partição serão inválidos, logo, os vizinhos no nível imediatamente inferior do diagrama de Hasse não são marcados para serem inspecionados nos próximos passos.

(ii) Da mesma forma, se o conjunto de restrições r é restritivo no sentido contrário ao dos refinamentos, então se r é verdadeiro para uma solução, todos os seus refinamentos também o serão. Por isso, a inspeção deve ser feita no sentido ascendente. Novamente, a única adaptação necessária para utilizar o mecanismo geral de busca ascendente é que a cada passo, deverá ser analisado se o conjunto de restrições é válido. Se o conjunto de restrições é falso na partição, então todas as soluções das quais a solução que está sendo inspecionada é refinamento também serão inválidos. Por isso, os vizinhos no nível imediatamente superior do diagrama de Hasse também não são marcados para serem inspecionados nos próximos passos.

(iii) Quando o conjunto de restrições é restritivo no sentido dos refinamentos e a função objetivo é decrescente no sentido dos refinamento, não é necessário avaliar o valor da função objetivo em todos as soluções de clusterização em grafos válidas. Isso é particularmente útil quando a função objetiva é muito custosa. Para aproveitar essa possibilidade, para utilizar o mecanismo geral de busca descendente deve ser novamente modificado. A cada passo, deverá ser analisado se algum dos vizinhos no

nível imediatamente abaixo do diagrama de Hasse é válido. Caso nenhum deles o seja, o valor da função objetivo naquela partição deve ser calculado. Senão, os seus vizinhos válidos no nível inferior são marcados para serem analisados em passos subsequentes.

(iv) Analogamente, quando o conjunto de restrições é restritivo no sentido contrário ao dos refinamentos e a função objetivo é crescente no sentido dos refinamento, também não é necessário avaliar o valor da função objetivo em todos as soluções de clusterização em grafos válidas. Para aproveitar essa possibilidade o mecanismo geral de busca ascendente deve ser modificado de uma forma análoga à modificação feita para o caso anterior. A cada passo, deverá ser analisado se algum dos vizinhos no nível imediatamente acima no diagrama de Hasse é válido. Caso nenhum deles o seja, o valor da função objetivo naquela partição deve ser calculado. Senão, os seus vizinhos válidos no nível imediatamente superior são marcados para serem analisados em passos subsequentes.

5.4 Discussões

Esse capítulo apresentou uma solução geral para o problema de clusterização em grafos. Foi observado que essa solução é muito custosa, no entanto, dependendo das características do conjunto de restrições e da função objetivo, essa solução poderia ser melhorada, diminuindo a quantidade de soluções de clusterização a ser inspecionadas.

Antes de analisar as características do conjunto de restrições e da função objetivo que permitem reduzir o espaço de busca do algoritmo que é solução geral do problema de clusterização, foi necessário estudar as características do conjunto de

possíveis soluções de clusterização em grafos para um grafo arbitrário G. Observou-se que esse conjunto é um subconjunto do conjunto de partições de $V(G)$. Por isso, $Clust(G)$ é um conjunto parcialmente ordenado e, portanto, pode ser representado por um diagrama de Hasse. Foi observado, ainda, que esse conjunto tem exatamente um máximo e um mínimo. Essa característica permite que todos os elementos de $Clust(G)$ sejam inspecionados por um mecanismo de busca ascendente iniciado no elemento mínimo ou por uma busca descendente iniciada no máximo.

Tendo em vista essas particularidades, foram propostas quatro combinações de características do conjunto de restrições e da função objetivo que permitem que soluções sejam eliminadas do processo de busca pela solução natural.

6 CONCLUSÃO E TRABALHOS FUTUROS

Esse trabalho desenvolveu um estudo aprofundado de clusterização em grafos, buscando identificar as principais falhas conceituais e as soluções já desenvolvidas para o problema. O primeiro problema observado foi que as notações, as definições e os conceitos utilizados variam bastante conforme o autor. Por isso, foi desenvolvida uma unificação da notação que permitiu adaptar as medidas e os algoritmos estudados de forma a facilitar que o leitor tenha uma visão ampla e clara dos diferentes trabalhos apresentados na revisão bibliográfica.

Por meio desse estudo, observou-se que não existe um consenso no que caracteriza um cluster ou uma solução de clusterização em grafos natural. Por isso, um mesmo grafo pode apresentar soluções de clusterização diferentes conforme o método utilizado para detecção de comunidades. Disso, decorre que não é possível predizer qual seria o resultado da clusterização de um grafo sem analisar as características específicas do algoritmo utilizado.

Para resolver essa situação, foi proposta uma descrição formal para o problema de clusterização em grafos. Essa formalização foi feita por meio de uma descrição parametrizada, em que, além do grafo de entrada, o problema tenha como instância uma função objetivo que meça a qualidade da solução do problema de clusterização em grafos e um conjunto de restrições que diga quais as soluções dentre todas as soluções possíveis são soluções válidas. A existência dessa função objetivo e do conjunto de restrições permite que o mesmo problema seja utilizado para descrever problemas com diferentes noções do que é uma comunidade em um grafo.

Foi apresentado, também, como adaptar as medidas de qualidade para serem utilizadas como funções objetivo e como as medidas de centralidade de vértices e arestas podem ser utilizadas para indicar restrições na solução de clusterização em

grafos. Posteriormente, a formalização proposta foi analisada para identificar quais dos trabalhos, dentre os estudados, podem ser descritos pelo problema desenvolvido. Observou-se que os trabalhos que não puderam ser descritos tinham em comum o fato que o problema resolvido pelo algoritmo não estava claro, mas apenas o procedimento utilizado para obter a solução.

Por fim, foi apresentada uma solução geral exaustiva para o problema de clusterização em grafos. Observou-se que essa solução, por analisar todas as possíveis soluções de clusterização de um grafo, é muito custosa. Por isso, foi analisado como as características da função objetivo e do conjunto de restrições podem ser combinadas a fim de que nem todas as possíveis soluções de clusterização devam ser inspecionadas e mesmo assim haja garantia que seja encontrada uma resposta correta para cada instância.

6.1 Impactos desse trabalho

Os resultados obtidos nesse trabalham atingem os objetivos definidos na proposta de dissertação aprovada durante o exame de qualificação. Inclusive, algumas partes superam os objetivos iniciais, como por exemplo o desenvolvimento de uma análise das características de $Clust(G)$ e de como diminuir o espaço de busca do algoritmo exaustivo que é solução geral para o problema.

O desenvolvimento de uma definição formal para o problema de clusterização em grafos é muito importante do ponto de vista teórico. Afinal, essa formalização permite que as análises realizadas nos trabalhos da área não sejam meramente qualitativas e comparativas, como é usual nos trabalhos encontrados na área. Dentre os impactos esperados no campo, destaca-se que essa definição formal do problema

viabiliza aferir a validade de soluções de clusterização em grafos.

Essas possibilidades evoluem o campo de clusterização em grafos, ao passo que permite que a clusterização em grafos não seja mais vista apenas como um nome genérico para uma variedade de métodos matemáticos, estatísticos e de heurísticas que podem ser usados para reconhecer grupos naturais em um grafo, mas como um problema computacional bem definido para o qual existem diversos algoritmos construídos.

6.2 Principais deficiências e trabalhos futuros

A principal deficiência notada na formalização proposta é que alguns algoritmos muito utilizados, como o algoritmo de clusterização em grafos baseado em *betweenness*, não podem ser abrangidos por essa solução. Isso ocorre devido ao fato que o problema resolvido por esses algoritmos não está claro, mas apenas o procedimento utilizado para obter a solução. Outra deficiência é que esse problema presume que a solução de clusterização seja uma partição do conjunto de vértices do grafo e, por isso, algoritmos que não utilizem essa definição, como por exemplo os probabilísticos, também não são abrangidos por essa definição.

Dessas deficiências decorre que dois trabalhos futuros se mostram muito importantes:

- Estender a definição formal proposta para o problema de clusterização em grafos para abranger o problema definido por aqueles problemas que definem apenas algum critério de aglomeração ou de divisão para ser utilizado em um procedimento hierárquico.
- Estender a definição formal proposta para o problema de clusterização em grafos

para ser utilizada nos casos em que a solução desejada não seja uma partição com conjunto de vértices do grafo.

REFERÊNCIAS

ANTHONISSE, J. M. *The Rush In A Directed Graph*. [S.l.], 1971. 1 – 10 p. (Stichting Mathematisch Centrum. Mathematische Besliskunde, Stichting Mathematisch Centrum. Mathematische Besliskunde-BN 9/71). Disponível em: <http://oai.cwi.nl/oai-/asset/9791/9791A.pdf>.

BANSAL, N.; BLUM, A.; CHAWLA, S. Correlation clustering. *Foundations of Computer Science, Annual IEEE Symposium on*, IEEE Computer Society, Los Alamitos, CA, USA, v. 0, p. 238, 2002. ISSN 0272-5428.

BAVELAS, A. A mathematical model of Group Structure. *Human Organizations*, v. 7, p. 16–30, 1948.

BEAUCHAMP, M. A. An improved index of centrality. *Behavioral Science*, State University of New York at Buffalo, v. 10, n. 2, p. 161–163, 1965. ISSN 1099-1743. Disponível em: <http://dx.doi.org/10.1002/bs.3830100205>.

BERRY, M. J.; LINOFF, G. *Data Mining Techniques: For Marketing, Sales, and Customer Support*. New York, NY, USA: John Wiley & Sons, Inc., 1997. ISBN 0471179809.

BOSS, S. L. B. *Caracterizações de buscas em hipermultigrafos*. Dissertação (Mestrado) — Universidade Federal do Paraná, 2010.

BOTAFOGO, R. A.; RIVLIN, E.; SHNEIDERMAN, B. Structural analysis of hypertexts: identifying hierarchies and useful metrics. *ACM Trans. Inf. Syst.*, ACM, New York, NY, USA, v. 10, p. 142–180, abr. 1992. ISSN 1046-8188. Disponível em: <http://doi.acm.org/10.1145/146802.146826>.

BOUTIN, F.; HASCOET, M. Cluster validity indices for graph partitioning. In: *Proceedings of the Information Visualisation, Eighth International Conference*. Washington, DC, USA: IEEE Computer Society, 2004. p. 376–381. ISBN 0-7695-2177-0. Disponível em: <http://portal.acm.org/citation.cfm?id=1018435.1021645>.

BRANDES, U.; GAERTLER, M.; WAGNER, D. Experiments on graph clustering algorithms. In: *In 11th Europ. Symp. Algorithms*. [S.l.]: Springer-Verlag, 2003. p. 568–579.

CARVALHO, A. X. Y.; MATA, D. D.; RESENDE, G. M. Clusterização dos municípios brasileiros. Brasília, IPEA, p. 181–208, 2007.

CLAUSET, A.; NEWMAN, M. E. J.; MOORE, C. Finding community structure in very large networks. *Physical Review E*, American Physical Society, v. 70, n. 6, dez. 2004. Disponível em: <http://dx.doi.org/10.1103/PhysRevE.70% -.066111>.

COHN, B. S.; MARRIOTT, M. Networks and centres of integration in indian civilization. *Journal of Social Research*, v. 1, p. 1–9, 1958.

COOK, K. S.; EMERSON, R. M.; GILLMORE, M. R. The Distribution of Power in Exchange Networks: Theory and Experimental Results. *The American Journal of Sociology*, v. 89, n. 2, p. 275–305, 1983. Disponível em: <http://www.jstor.org/stable-/2779142>.

DAVIES, D. L.; BOULDIN, D. W. A Cluster Separation Measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1, n. 2, p. 224–227, abr. 1979. ISSN 0162-8828. Disponível em: <http://dx.doi.org/10.1109/TPAMI.1979.4766909>.

DONGEN, S. V.; DONGEN, S. V. *Performance Criteria for Graph Clustering and Markov Cluster Experiments*. [S.l.], 2000.

DONGEN, S. van. *Graph Clustering by Flow Simulation*. Tese (Doutorado) — University of Utrecht, Utrecht, maio 2000.

DUNN, J. C. Well separated clusters and optimal fuzzy-partitions. *Journal of Cybernetics*, v. 4, p. 95–104, 1974.

EDACHERY, J. et al. Graph clustering using distance-k cliques. In: *in Proc. of Graph Drawing*. [S.l.]: Springer-Verlag, 1999. p. 98–106.

EDMINISTER, J.; NAHVI, J. M. *Circuitos eletricos*. BOOKMAN COMPANHIA ED, 1985. ISBN 9788536305516. Disponível em: <http://books.google.com.br-/books?id=stDABhaQqGAC>.

EVERITT, B. S. *Cluster Analysis*. 3rd. ed. A Hodder Arnold Publication, 1993. Hardcover. ISBN 0340584793. Disponível em: <http://www.amazon.com/exec/obidos-/redirect?tag=citeulike07-20\&path=ASIN/0340584793>.

FLAKE, G. W.; TARJAN, R. E.; TSIOUTSIOULIKLIS, K. Graph Clustering and Minimum Cut Trees. *Internet Mathematics*, v. 1, n. 4, p. 385–408, 2004.

FORTUNATO, S.; BARTHéLEMY, M. Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, v. 104, n. 1, p. 36–41, jan. 2007. Disponível em: <http://dx.doi.org/10.1073/pnas.0605965104>.

FORTUNATO, S.; LATORA, V.; MARCHIORI, M. A Method to Find Community Structures Based on Information Centrality. *Physical Review E*, American Physical Society, v. 70, n. 5, p. 056104+, ago. 2004. Disponível em: <http://dx.doi.org/10.1103-/PhysRevE.70% -.056104>.

FREEMAN, L. C. A Set of Measures of Centrality Based on Betweenness. *Sociometry*, American Sociological Association, v. 40, n. 1, p. 35–41, mar. 1977.

FREEMAN, L. C. Centrality in social networks: Conceptual clarification. *Social Networks*, v. 1, p. 215– 239, 1979.

FREEMAN, L. C.; BORGATTI, S. P.; WHITE, D. R. Centrality in valued graphs: A measure of betweenness based on network flow. *Social Networks*, North-Holland, v. 13, n. 2, p. 141–154, jun. 1991.

GAREY, M.; JOHNSON, D. *Computers and intractability*. [S.l.]: Freeman San Francisco, 1979.

GIRVAN, M.; NEWMAN, M. E. J. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences of the United States of America*, Santa Fe Institute, 1399 Hyde Park Road, Santa Fe, NM 87501, USA. girvan@santafe.edu, v. 99, n. 12, p. 7821–7826, jun. 2002. ISSN 0027-8424. Disponível em: <http://dx.doi.org/10.1073/pnas.122653799>.

GOMORY, R. E.; HU, T. C. Multi-Terminal Network Flows. *Journal of the Society for Industrial and Applied Mathematics*, Society for Industrial and Applied Mathematics, v. 9, n. 4, p. 551–570, 1961. ISSN 03684245. Disponível em: <http://dx.doi.org/10-.2307/2098881>.

GRANOVETTER, M. S. The Strength of Weak Ties. *The American Journal of Sociology*, JSTOR, v. 78, n. 6, p. 1360–1380, 1973.

HALKIDI, M.; BATISTAKIS, Y.; VAZIRGIANNIS, M. On clustering validation techniques. *J. Intell. Inf. Syst.*, Kluwer Academic Publishers, Hingham, MA, USA, v. 17, p. 107–145, dez. 2001. ISSN 0925-9902. Disponível em: <http://portal.acm.org/citation-.cfm?id=607585.607609>.

HALKIDI, M.; VAZIRGIANNIS, M. A data set oriented approach for clustering algorithm selection. In: *Proceedings of the 5th European Conference on Principles of Data Mining and Knowledge Discovery*. London, UK: Springer-Verlag, 2001. (PKDD '01), p. 165–179. ISBN 3-540-42534-9. Disponível em: <http://portal.acm.org/citation-.cfm?id=645805.669997>.

HARTUV, E.; SHAMIR, R. A clustering algorithm based on graph connectivity. *Inf. Process. Lett.*, Elsevier North-Holland, Inc., Amsterdam, The Netherlands, The Netherlands, v. 76, p. 175–181, dez. 2000. ISSN 0020-0190. Disponível em: <http://portal.acm.org/citation.cfm?id=364456.364469>.

JAIN, A. K.; DUBES, R. C. *Algorithms for clustering data*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1988. ISBN 0-13-022278-X.

JAIN, A. K.; MURTY, M. N.; FLYNN, P. J. *Data Clustering: A Review*. 1999.

KANNAN, R.; VEMPALA, S.; VETA, A. On clusterings-good, bad and spectral. In: *FOCS '00: Proceedings of the 41st Annual Symposium on Foundations of Computer Science*. Washington, DC, USA: IEEE Computer Society, 2000. p. 367. ISBN 0-7695-0850-2. Disponível em: <http://portal.acm.org/citation.cfm?id=796585>.

KIRA, K.; RENDELL, L. A. The Feature Selection Problem: Traditional Methods and a New Algorithm. In: *AAAI*. Cambridge, MA, USA: AAAI Press and MIT Press, 1992. p. 129 – 134.

KNUTH, D. E. *Art of Computer Programming, Volume 1: Fundamental Algorithms (3rd Edition)*. 3. ed. [S.l.]: Addison-Wesley Professional, 1997.

LATORA, V.; MARCHIORI, M. Efficient Behavior of Small-World Networks. out. 2001. Disponível em: <http://arxiv.org/abs/cond-mat/0101396>.

LATORA, V.; MARCHIORI, M. Economic small-world behavior in weighted networks. *The European Physical Journal B - Condensed Matter and Complex Systems*, Springer Berlin / Heidelberg, v. 32, p. 249–263, 2003. ISSN 1434-6028. Disponível em: <http://dx.doi.org/10.1140/epjb/e2003-00095-5>.

LATORA, V.; MARCHIORI, M. A measure of centrality based on the network efficiency. *New Journal of Physics*, IOP Publishing, v. 9, n. 6, p. 188, 2007.

LEAVITT, H. J. Some effects of certain communication patterns on group performance. *Journal of Abnormal and Social Psychology*, v. 46, p. 38–50, 1951.

LESOT, M.-J.; RIFQI, M.; BENHADDA, H. Similarity measures for binary and numerical data: a survey. *Int. J. Knowl. Eng. Soft Data Paradigm.*, Inderscience Publishers, Inderscience Publishers, Geneva, SWITZERLAND, v. 1, p. 63–84, dez. 2009. ISSN 1755-3210. Disponível em: <http://portal.acm.org/citation.cfm?id=1479242-.1479248>.

LIPSCHUTZ, S. *Matemática Discreta*. BOOKMAN COMPANHIA ED, 2004. ISBN 9788536303611. Disponível em: <http://books.google.com.br-/books?id=2S9bwDmD1P0C>.

MARSDEN, P. V.; LAUMANN, E. O. Collective action in a community elite: Exchange, influence resources and issue resolution. *Power, Paradigms, and Community Research*, ISA/Sage, London, p. 199–255, 1977.

MATULA, D. W. Graph theoretic techniques for cluster analysis algorithms. In: RYZIN, J. van (Ed.). *Classification and Clustering*. New York: Academic Press, 1977. p. 95–129.

MIHAIL, M. et al. *On the semantics of Internet topologies*. [S.l.], 2002.

MITZENMACHER, M.; UPFAL, E. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. [S.l.]: Cambridge University Press, 2005.

NEWMAN, M. Fast algorithm for detecting community structure in networks. *Physical Review E*, v. 69, set. 2003. Disponível em: <http://arxiv.org/abs/cond-mat/0309508>.

NEWMAN, M. E. J. *A measure of betweenness centrality based on random walks*. set. 2003. Disponível em: <http://arxiv.org/abs/cond-mat/0309045>.

NEWMAN, M. E. J. Detecting community structure in networks. *The European Physical Journal B - Condensed Matter and Complex Systems*, Springer Berlin / Heidelberg, v. 38, n. 2, p. 321–330–330, mar. 2004. ISSN 1434-6028. Disponível em: <http://dx.doi.org/10.1140/epjb/e2004-00124-y>.

NEWMAN, M. E. J.; GIRVAN, M. Finding and evaluating community structure in networks. *Physical Review E*, American Physical Society, v. 69, n. 2, p. 026113+, fev. 2004. Disponível em: <http://dx.doi.org/10.1103/PhysRevE.69% -.026113>.

NIEMINEN, J. On centrality in a graph. *Scandinavian Journal of Psychology*, v. 15, p. 322–336, 1974.

NILSSON, J. W.; RIEDEL, S. A. *Circuitos Elétricos*. 6. ed. [S.l.]: LTC, 2003.

PORTER, M. A.; ONNELA, J.-P.; MUCHA, P. J. Communities in Networks. *Notices of the American Mathematical Society*, v. 56, n. 9, p. 1082–1097, set. 2009. Disponível em: <http://arxiv.org/abs/0902.3788>.

SABIDUSSI, G. The centrality index of a graph. *Psychometrika*, v. 31, p. 581–603, 1966.

SCHAEFFER, S. E. *Algorithms for Nonuniform Networks*. Espoo, Finland, abr. 2006. xxii+183 p. Doctoral dissertation.

SCHAEFFER, S. E. Graph clustering. *Computer Science Review*, v. 1, n. 1, p. 27 – 64, 2007. ISSN 1574-0137. Disponível em: <http://www.sciencedirect.com/science-/article/B8JDG-4PBG1S7-1/2/6537f3d1ffbf391086c60dbeba874b13>.

SCHWAHN, A. M. Minimum cut tree games. In: *Proceedings of the First ICST international conference on Game Theory for Networks*. Piscataway, NJ, USA: IEEE Press, 2009. (GameNets'09), p. 17–25. ISBN 978-1-4244-4176-1. Disponível em: <http://portal.acm.org/citation.cfm?id=1689499.1689502>.

SHAW, M. E. Group structure and the behaviour of individuals in small groups. *Journal of Psychology*, v. 38, p. 139–149, 1954.

SHIMBEL, A. Structural parameters of communication networks. *Bulletin of Mathematical Biology*, v. 15, n. 1, p. 501–507, 1953. Disponível em: <http://dx.doi-.org/10.1007/BF02476438>.

STEPHENSON, K.; ZELEN, M. Rethinking centrality: Methods and examples. *Social Networks*, v. 11, n. 1, p. 1–37, 1989.

TAN, P.-N.; STEINBACH, M.; KUMAR, V. *Introduction to Data Mining*. Us ed. Addison Wesley, 2005. 487—568 p. Hardcover. ISBN 0321321367. Disponível em: <http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20\&path=ASIN-/0321321367>.

THEODORIDIS, Y. *Spatial Datasets: An "unofficial" collection*. 1999. Disponível em: <http://dias.cti.gr/~ytheod/research/datasets/spatial.html>.

WEST, D. B. *Introduction to Graph Theory*. 2. ed. [S.l.]: Prentice Hall, 2000. ISBN 0130144002.

WU, F.; HUBERMAN, B. A. Finding communities in linear time: A physics approach. *European Physical Journal B*, v. 38, p. 331–338, 2004.

ZAïANE, O. R. et al. On data clustering analysis: Scalability, constraints and validation. In: *In Proceedings of the 6th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD*. [S.l.: s.n.], 2002. p. 28–39.