Aula 18 – Comunidade, Estabilidade, Cobertura e Cliques

Notas de Aula de Teoria dos Grafos

Prof[a.]: Patrícia D. L. Machado

UFCG – Unidade Acadêmica de Sistemas e Computação

Sumário

Nesta aula, estudaremos conceitos básicos para análise de grafos, em especial, o conceito de comunidade e conceitos auxiliares utilizados para determinar ou identificar comunidades.

Comunidade

Uma comunidade é um subgrafo denso ou ***cluster*** dentro de um grafo cujos vértices estão mais conectados com aqueles dentro do *cluster* do que com os fora do *cluster*.

A categorização em comunidade é utilizada para se realizar diferentes análise em grafos:

- **Estática** (investigação de conceitos e propriedades em uma versão específica do grafo - escopo desta disciplina): Quais são as comunidades? Quem pertence a uma comunidade? Quão densa uma comunidade é?
- **Dinâmica** (estudo da evolução do grafo ao longo do tempo): Como esta comunidade se formou? Quais comunidades são estáveis? Que comunidades são temporárias?

- **Predição**: (Estimar futuras versões com base em dados históricos e técnicas especializadas): É provável que uma comunidade cresça? Que vértices deverão fazer parte da comunidade no futuro? Papéis dominantes irão emergir em uma comunidade?

Duas comunidades podem ter vértices em comum. A definição mais geral é baseada no princípio de que pares de vértices são mais prováveis de serem conectados se ambos forem membros da(s) mesma(s) comunidade(s), e menos provável de serem conectados se não compartilharem comunidades.

O estudo e a identificação de comunidades é importante pois:

- O conceito nos permite criar um mapa de grande escala de um grafo, uma vez que as comunidades individuais agem como meta-vértices o que facilita seu estudo.
- Comunidades individuais geralmente correspondem a unidades funcionais do sistema. Comunidades geralmente têm propriedades muito diferentes das propriedades médias dos grafos.
- A formação de comunidades afeta a disseminação de informações.
- Características de uma comunidade viabiliza a previsão de links perdidos e a detecção de links falsos.

Diferentes conceitos na teoria dos grafos são utilizados para auxiliar na identificação de comunidades. Alguns destes conceitos são apresentados a seguir.

Conjunto Estável

Um **conjunto estável** em um grafo *G* é um subconjunto de vértices W que não são adjacentes, isto é, para todos os pares $v, w \in W$, vw não é uma aresta de *G.*

Conjuntos estáveis são também chamados de **conjuntos independentes**.

Na Figura 1, temos 2 grafos de Petersen nos quais o conjunto de vértices com background preto formam um conjunto estável. Para um mesmo grafo, é possível identificar diferentes conjuntos estáveis.

Um conjunto estável em um grafo é **máximo** se o grafo não contém outro maior e um conjunto estável em um grafo é **maximal** se este não pode ser estendido para outro maior.

Em nosso exemplo, o conjunto a esquerda é maximal, visto que não pode mais ser estendido, todos os vértices em laranja são adjacentes a algum vértices em preto. O conjunto a direita é maximal, mas também é máximo, pois, para este grafo, não existe outro de tamanho maior.

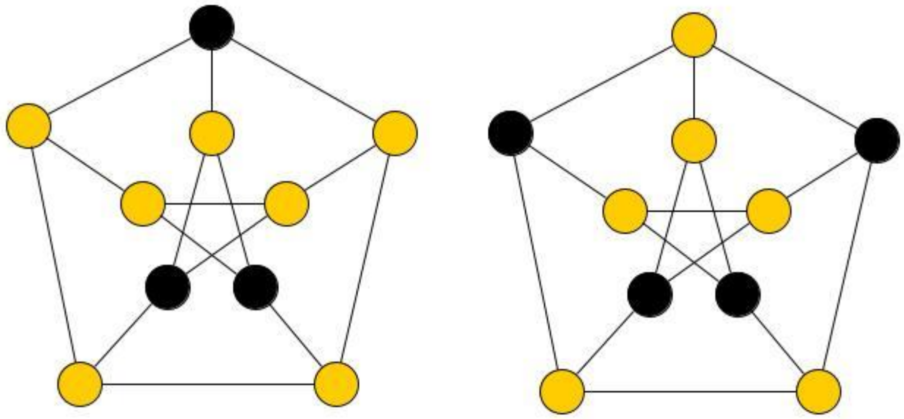


Figura 1

O **número de estabilidade** de $G\ -\alpha(\boldsymbol{G})$ ***– é definido como*** a cardinalidade do conjunto estável máximo em um grafo. Em nosso exemplo, o número de estabilidade do grafo de Petersen é 4.

Este conceito pode ser aplicado para auxiliar na detecção de comunidades em um grafo (*clustering*). Um conjunto estável maximal pode representar um grupo de vértices relacionados. Pelo menos um vértice do conjunto estável está conectado com os outros vértices omitidos do grupo. *Clustering* é uma técnica básica utilizada para resolver inúmeros problemas práticos como parte de estratégias tal com aprendizagem de máquina e mineração de dados que podem ser utilizadas para análise e reconhecimento de padrões e recuperação de informações. Objetos de um mesmo cluster são considerados mais semelhantes entre si do que com os de outros clusters.

Porém, encontrar conjuntos estáveis máximos é um problema computacionalmente difícil de resolver na prática. Na **JGraphT**, temos uma implementação restrita a grafos cordais, a classe **ChordalGraphIndependentSetFinder** no pacote org.jgrapht.alg.independentset. Um **grafo cordal (*chordal*)** é aquele em que todos os ciclos de quatro ou mais vértices possuem uma corda, que é uma aresta que não faz parte do ciclo, mas conecta dois vértices do ciclo. Na Figura 2, temos um exemplo de um grafo cordal. Note que o ciclo maior é formado pelos vértices $(1,2,3,4,5,6,7,8)$ e as aresta 16 e 26 são exemplos de cordas. Neste grafo, $\mathrm{W}\ =\ \{8,2,4,9\}$ é um exemplo de um conjunto estável (independente).

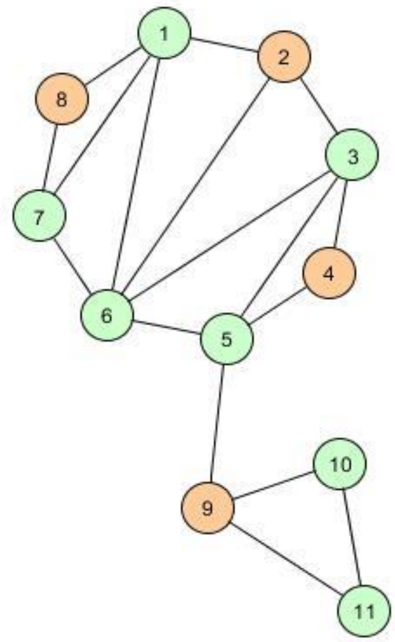
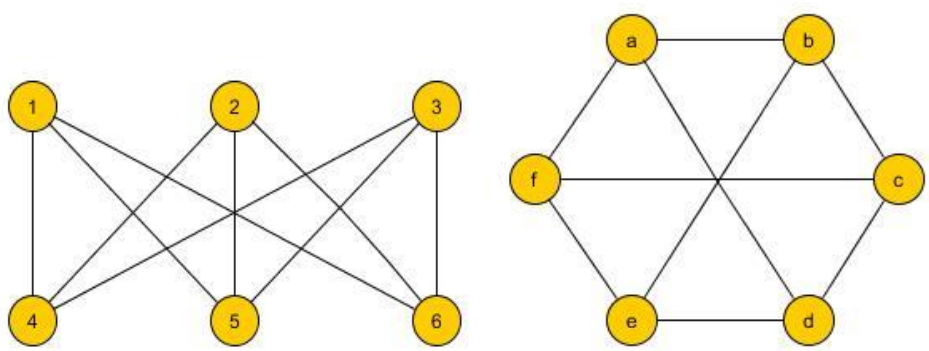


Figura 2

Exercício: Mostre que o número de estabilidade é invariante sob isomorfismo. Em outras palavras, se G e H são grafos isomorfos então $\alpha(G) = \alpha(H)$.

Sejam H e G dois grafos isomorfos. Seja W um conjunto estável máximo em G. Então, visto que G e H são isomórficos, há um vértice correspondente em H para cada vértice de G. Então, é possível encontrar em H um conjunto estável de mesmo tamanho. Assim, G e H possuem o mesmo número de estabilidade. A Figura 3 abaixo ilustra estes conceitos.



Grafo G　　　　Grafo H

θ = (1→b, 2→d, 3→f, 4→c,5→e, 6→a)

$W_G = \{1,2,3\}$ e $W_H = \{b, d, f\}$

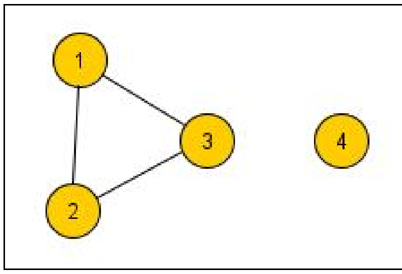
Figura 3

Cobertura

Uma **cobertura de vértices** em um grafo *G* é um subconjunto de vértices *C* tal que, para toda aresta $xy \in E(G), x \in C \ ou \ y \in C$. Ou seja, *C* possui pelo menos um dos terminais de cada aresta do grafo *G*.

O número de vértices em uma **cobertura mínima**, ou seja, não há outra de tamanho menor, é definido como o **número de cobertura** de *G, representado por* $\boldsymbol{\beta}(\boldsymbol{G})$.

Para o grafo da Figura 4, os conjuntos C_1, ..., C_7 são coberturas de vértices no grafo, mas apenas C_5, C_6 e C_7 são coberturas mínimas. Portanto, o número de cobertura deste grafo é 2.



$C_1 = \{1,2,3,4\}$
$C_2 = \{1,2,4\}$
$C_3 = \{1,3,4\}$
$C_4 = \{2,3,4\}$
$C_5 = \{1,2\}$
$C_6 = \{2,3\}$
$C_7 = \{1,3\}$

Figura 4

O conceito de cobertura pode ser aplicado para determinar a quantidade de recursos necessários a serem alocados de forma que os relacionamentos entre vértices sejam contemplados. Por exemplo, considere o seguinte problema ilustrado na Figura 5. Quantos vigilantes são necessários para monitorar todos os corredores deste plano? Considere que os terminais dos corredores e cruzamentos estão representados por vértices e as arestas denotam os corredores. Um vigilante consegue monitor todos os corredores que conseguir visualizar, ou seja, aqueles adjacentes ao vértice onde está posicionado.

Para determinar o número mínimo de vigilantes e seu posicionamento, basta encontrar uma cobertura de tamanho mínimo no grafo. Neste caso, C pode ser formada pelos vértices 2,4 e 5.

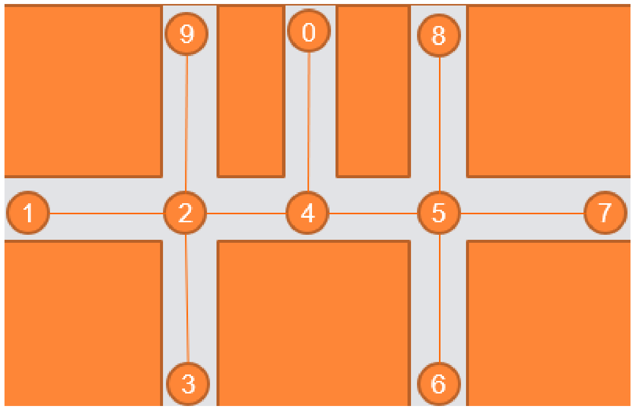


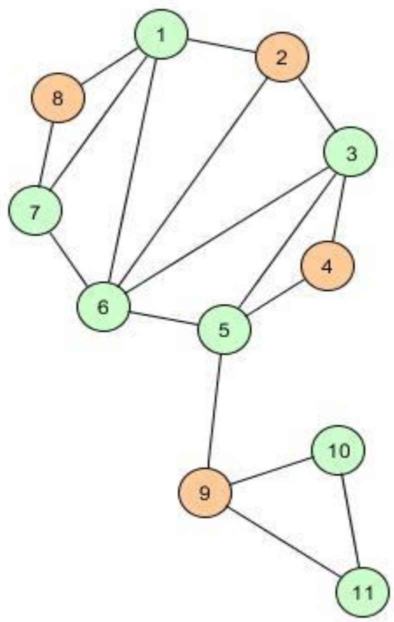
Figura 5

Identidade de Gallai:

- Um conjunto S é **estável** se e somente se $V\setminus S$ é uma cobertura de vértices G.
- $\alpha(G) + \beta(G) = v(G)$

O exemplo apresentando na Figura 1 pode ser utilizado para ilustrar a identidade de Gallai. Enquanto os vértices em preto representam conjuntos estáveis S, tanto no grafo de Petersen da esquerda quanto no da direita, os vértices em laranja, representam coberturas de vértice V/S. Note que o grafo de Petersen possui $v(G) = 10$ vértices, seu número de estabilidade, $\alpha(G)$, é 4 e seu número de cobertura, $\beta(G)$, é 6. O grafo da direita apresenta um conjunto estável máximo e uma cobertura mínima.

O problema da cobertura mínima de vértices pode ser resolvido por aproximações (implementações que não se baseiam em um algoritmo preciso, mas podem retornar resultados quasi-ótimos com um tempo computacional menor) e algoritmos. Estudamos aproximações na aula sobre complexidade computacional. A **JGraphT** dispõe de diferentes aproximações e um algoritmo preciso, **RecursiveExactVCImpl**, no pacote org.jgrapht.alg.vertexcover. A Figura 6 abaixo ilustra as coberturas encontradas por estas implementações considerando o grafo exemplo.



BarYehudaEvenTwoApproxVCImpl :
[2, 3, 6, 5, 1, 7, 9, 10]

ClarksonTwoApproxVCImpl:
[6, 3, 1, 5, 9, 7, 10]

EdgeBasedTwoApproxVCImpl:
[2, 3, 6, 5, 1, 7, 9, 10]

GreedyVCImpl (Heurística):
[6, 3, 1, 9, 5, 7, 10]

RecursiveExactVCImpl (Algoritmo):
[6, 3, 1, 5, 7, 10, 11]

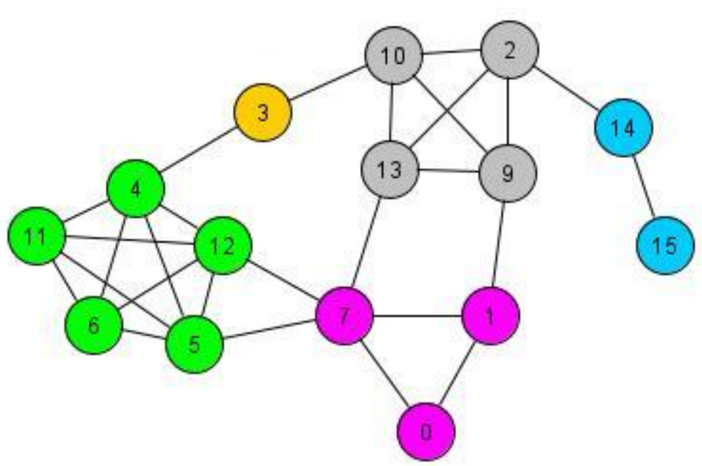
Figura 6

Clique

Um **clique** W em um grafo G é um conjunto de vértices tal que para todo $v, w \in W$, existe uma aresta $vw \in E(G)$. Ou seja, um clique é um conjunto de vértices mutuamente adjacentes.

Outra forma de definir clique é como um subgrafo completo induzido de um grafo G.

A Figura 7 abaixo apresenta um grafo com cliques de diferentes tamanhos.



Exemplos de cliques de tamanho 1 a 5 neste grafo:

$W_1 = \{3\}$

$W_2 = \{14, 15\}$

$W_3 = \{1,7,0\}$

$W_4 = \{2,10,9,13\}$

$W_5 = \{4, 6, 5, 11, 12\}$

Figura 7

Na prática, temos interesse em encontrar os cliques de tamanho máximo, particularmente, maior ou igual a 3, já que os de tamanho 2 e 1 são trivialmente definidos.

Note que:

- Todo vértice sozinho forma um clique de tamanho 1.
- Dois vértices adjacentes formam um clique de tamanho 2.
- Todo clique de tamanho n, possui cliques incorporados de tamanho 1 a $n - 1$.

Ou seja, no clique W_5 do exemplo anterior, podemos encontrar cliques incorporados de tamanho 1, 2, 3 e 4. Por exemplo, $\{11\}, \{4,12\}, \{11, 5, 6\}, \{4, 12, 5, 6\}$, respectivamente. Isto se deve ao fato de que todo $\mathrm{K}_{\text{n-i}}, 1 \leq \mathrm{i} < \mathrm{n}$, está incorporado em um K_n.

O termo clique foi inicialmente aplicado para designar grupos de pessoas que se conhecem umas as outras. Atualmente é largamente aplicado para solução de problemas de problemas de *clustering* em diversas áreas do conhecimento, particularmente em bioinformática.

No reconhecimento de faces, características de uma face são extraídas e modeladas como vértices. Arestas ligam dois vértices dependendo da orientação relativa dos vértices. Semelhança entre duas imagens pode ser determinada a partir do tamanho dos cliques maximais em comum (Figura 8).

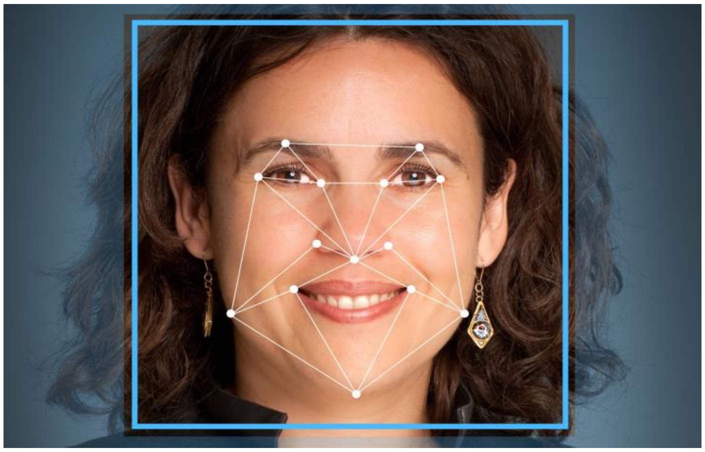


Figura 8[1]

Relações entre os Conceitos

Os conceitos de conjunto estável, cobertura e clique estão relacionados da seguinte forma. Seja G um grafo simples:

[1] Imagem: www.pcguia.pt

a) *G* possui um conjunto estável com *k* vértices
b) O complemento de *G* tem um clique com *k* vértices
c) *G* possui uma cobertura de vértices com *n*-*k* vértices

Na Figura 9 abaixo, temos que o grafo de petersen e o seu complemento (a direita). Este grafo possui um conjunto independente de tamanho 4 (a) e uma cobertura de tamanho 10-4 (c). O mesmo conjunto de vértices do conjunto estável, é um clique no complemento do grafo (b).

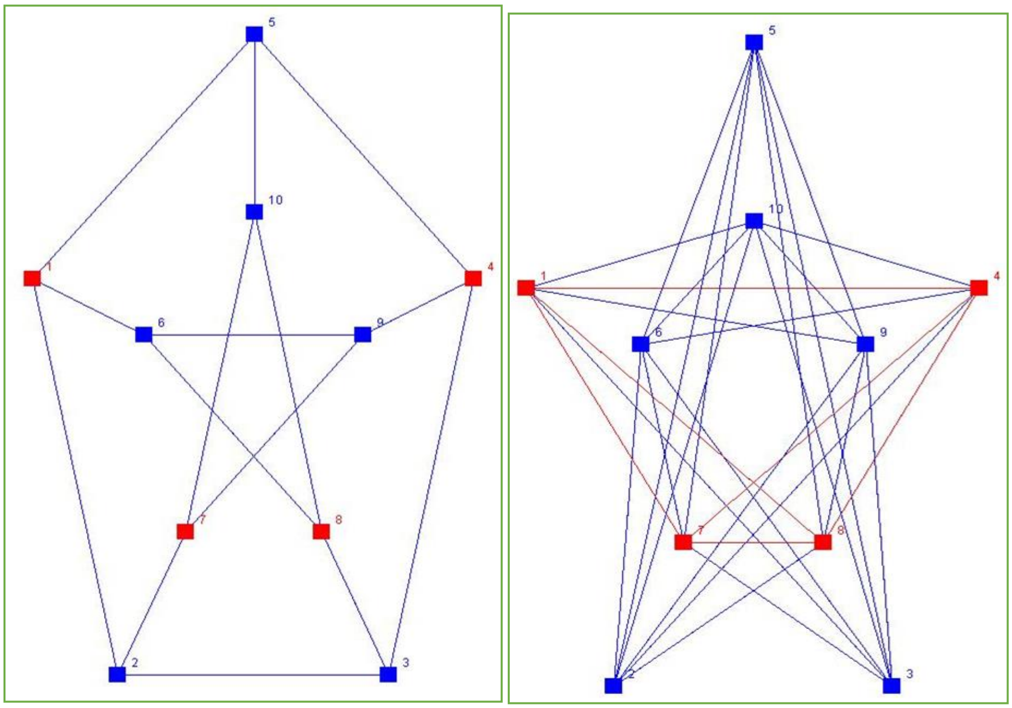


Figura 9

O Problema do Clique

Em Ciência da Computação, o problema do clique é o problema computacional de encontrar um clique máximo ou todos os cliques em um certo grafo. Este problema é NP-completo e difícil de ser aproximado (estudaremos este conceito na aula sobre complexidade computacional).

No entanto, muitos algoritmos para computar cliques tem sido desenvolvidos, executando ou em tempo exponencial (Bron–Kerbosch algorithm) ou para famílias especializadas de grafos, por exemplo: grafos planares, para os quais o problema pode ser resolvido em tempo polinomial.

O algoritmo de força bruta encontra um clique de tamanho 4 em um grafo de 7 vértices, verificando sistematicamente todas os possíveis subgrafos de 4 vértices, um total de 35 subgrafos.

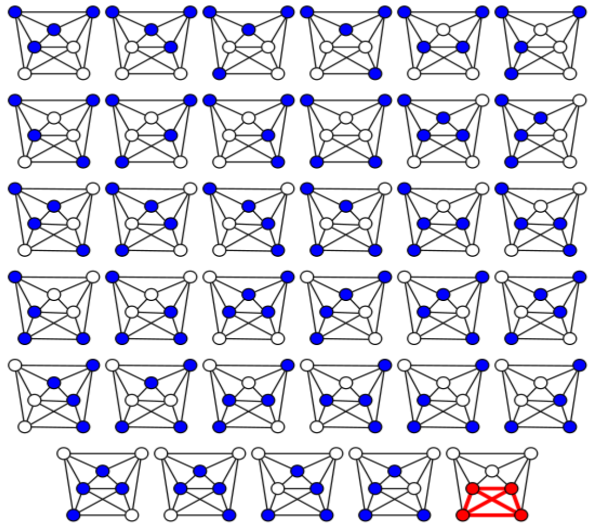
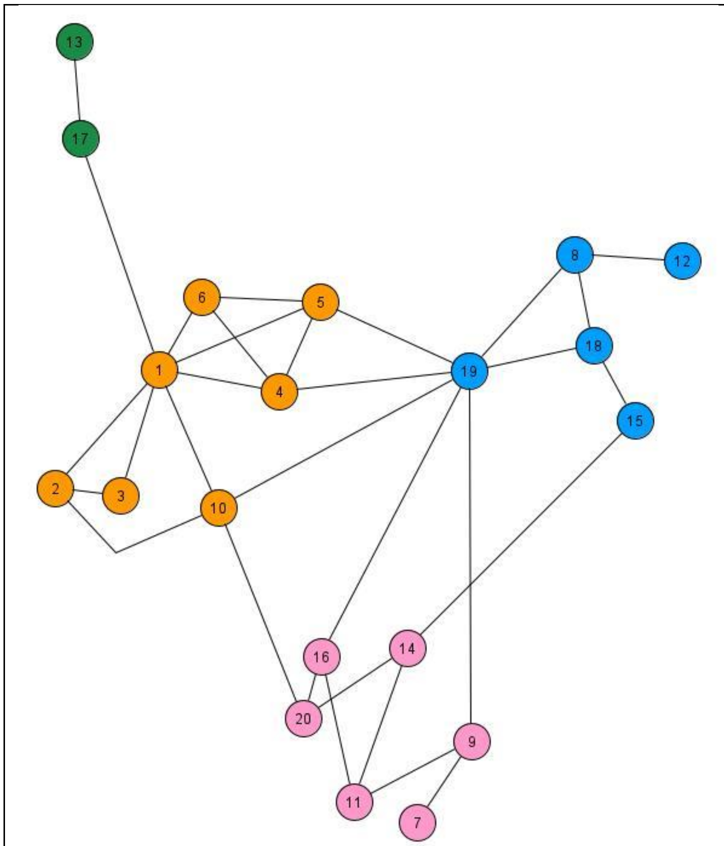


Figura 10 [2]

O pacote **[org.jgrapht.alg.clique]()** da **JGraphT** possui implementações que determinam os cliques maximais e máximos em um grafo. Para o grafo abaixo, são encontrados os seguintes cliques:



DegenearyBronKerboschCliqueFinder cliques:
[12, 8] [9, 7] [17, 13] [17, 1] [9, 11] [9, 19] [18, 8, 19] [18, 15] [14, 11] [11, 16] [14, 20] [14, 15] [20, 10] [20, 16] [2, 3, 1] [2, 10, 1] [10, 19] [19, 16] [4, 19, 5] [6, 4, 1, 5]

PivotBronKerboschCliqueFinder cliques:
[20, 14] [20, 10] [20, 16] [12, 8] [9, 7] [9, 11] [14, 11] [11, 16] [2, 10, 1] [2, 3, 1]
[6, 4, 1, 5] [17, 1] [9, 19] [4, 19, 5] [10, 19]
[18, 8, 19] [19, 16] [14, 15] [18, 15] [17, 13]

Figura 11

[2] Imagem: http://en.wikipedia.org/wiki/Clique_problem

N-Clique

Um $\boldsymbol{n}$**-clique** W em um grafo *G* (***near clique***) é um conjunto de vértices tal que para todo $v, w \in W$, existe um caminho entre v e w de tamanho máximo n.

Se n = 1, um n-clique é igual a um clique. Ou seja, existe um caminho de tamanho máximo 1 entre todos os vértices do clique.

Na Figura 12 abaixo, apresentamos um exemplo de 1-clique e 2-clique no grafo de Petersen.

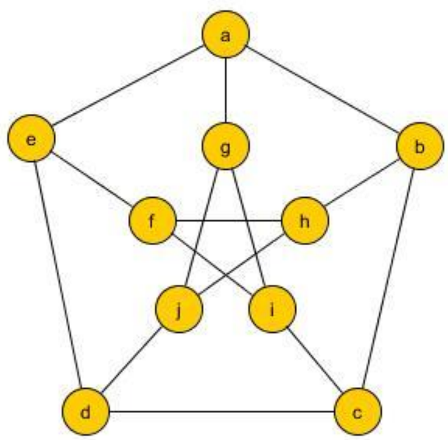
	**1-clique** $W = \{a, g\}$ **2-clique** $W = \{a, g, f, j, i, j, d, c\}$

Figura 12

Detecção de Clusters na yEd

A ferramenta yEd utiliza um algoritmo para identificar grupos ou clusters em um grafo que pode ser executado a partir da opção **Grouping** no menu principal (versão *desktop*). Dentre as opções de tipos de *clusters*, a Figura 13 abaixo ilustra os "*natural clusters*".

Um agrupamento em *natural clusters* deve cumprir as seguintes propriedades:

- cada vértice é membro de exatamente um grupo;
- cada vértice deve ter muitas adjacências com outros membros de seu grupo; e
- cada vértice deve ter poucas ou mesmo nenhuma adjacência com vértices de outros grupos.

O algoritmo é baseado no *Edge Betweenness Clustering* proposto por Girvan e Newman.

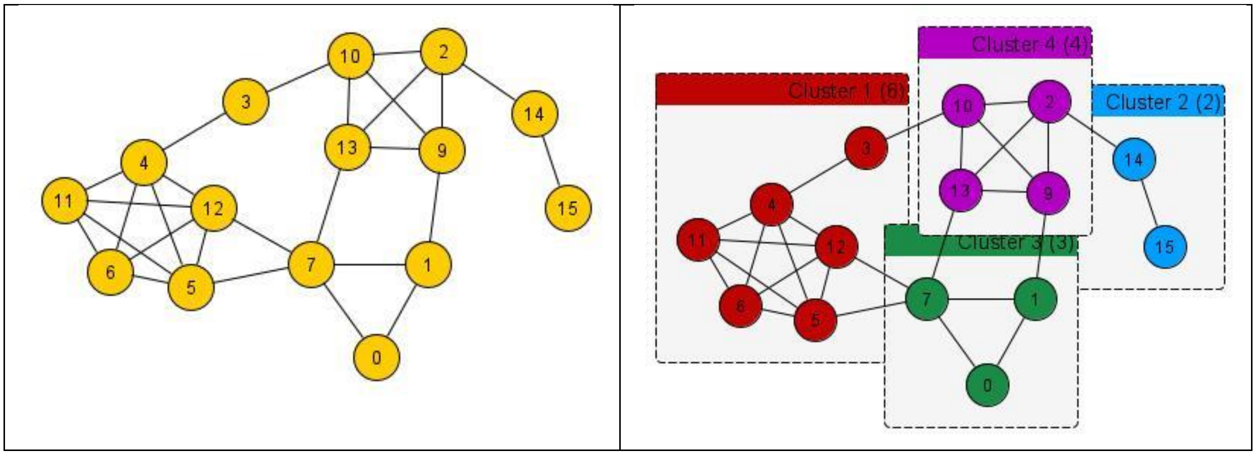

10
2
3
14
4
13
9
11
12
15
6
7
1
5
0
Cluster 4 (4)
Cluster 1 (8)
Cluster 2 (2)
Cluster 3 (3)

Figura 13

Exercícios Propostos

1. Encontre um conjunto estável máximo em um K_n. Encontre um conjunto estável máximo no complemento de um K_n.

2. Suponha que X e Y são conjuntos estáveis maximais de um grafo. É verdade que X e Y são disjuntos (ou seja, que $X \cap Y = \emptyset$)?
Questão adaptada de: http://www.ime.usp.br/~pf/grafos-exercicios/

3. Calcule um conjunto estável máximo em um caminho. Calcule um conjunto estável máximo em um circuito.

4. Encontre um clique máximo em um K_n. Encontre um clique máximo no complemento de um K_n.

5. Seja G um circuito de comprimento 6. Encontre um clique máximo no grafo G.
Questão adaptada de: http://www.ime.usp.br/~pf/grafos-exercicios/

6. Suponha que G é um grafo bipartido. Quantos vértices tem um clique máximo em G?
Questão adaptada de: http://www.ime.usp.br/~pf/grafos-exercicios/

7. Qual a relação entre o problema do clique máximo e o problema do conjunto estável máximo? Como é possível usar um algoritmo que resolve um dos problemas para resolver o outro?

8. O que é uma cobertura minimal? Construa um exemplo de uma cobertura minimal. É verdade que toda cobertura minima é minimal? É verdade que toda cobertura minimal é minima?
Questão adaptada de: http://www.ime.usp.br/~pf/grafos-exercicios/

9. Suponha que T é uma árvore. É verdade que toda cobertura minimal de T é mínima?
Questão adaptada de: http://www.ime.usp.br/~pf/grafos-exercicios/

Referências

J. A. Bondy and U. S. R. Murty.Graph Theory. Springer, 2008,2010.

- 12.1

Wikipedia

- http://en.wikipedia.org/wiki/Independent_set_(graph_theory)
- http://en.wikipedia.org/wiki/Clique_(graph_theory)
- https://en.wikipedia.org/wiki/Community_structure