

Disciplina DCE529 - AEDS 3	Método de realização Presencial	Data de apresentação 07/06/2023 às 8h00
Professor Iago Augusto de Carvalho (iago.carvalho@unifal-mg.edu.br)		

Trabalho prático 3 - Algoritmo em Grafos

O objetivo deste segundo trabalho é utilizar algoritmos em grafos para resolver um problema de centralidade em redes complexas.

Uma rede complexa pode ser entendida como um grande grafo sem uma estrutura definida, cujas conexões não podem ser definidas de uma forma simples. Os vértices destas redes podem ser categorizados de acordo com sua importância relativa. Diz-se que quanto mais importante é um vértice, maior é sua centralidade.

Existem diversas métricas de centralidade de vértices em redes complexas, como o *betweenness*, o *closeness*, o *eigenvector*, o índice de centralidade de *Katz* e a centralidade harmônica, dentre outros. Neste trabalho, estamos interessados no índice *betweenness*.

O *betweenness* quantifica o número de vezes que um vértice faz parte do caminho mínimo entre outros dois vértices de um grafo. Quanto mais vezes este vértice é utilizado nos caminhos mínimos de outros vértices, maior é o seu índice *betweenness*. Matematicamente, o valor de *betweenness* B_v de um vértice $v \in V$ pode ser definido como

$$B_v = \sum_{s \neq v \neq t \in V} \frac{\delta_{st}(v)}{\delta_{st}},$$

onde δ_{st} é o número total de caminhos mínimos no grafo e $\delta_{st}(v)$ é o número destes caminhos que passam pelo vértice v . Note que δ_{st} é igual ao número de pares de vértices no grafo e $\delta_{st}(v)$ não considera o número de caminhos iniciados ou finalizados em v .

O que deve ser desenvolvido: Neste trabalho cada grupo deverá implementar o cálculo de *betweenness*. Para isto, deverão ser implementados

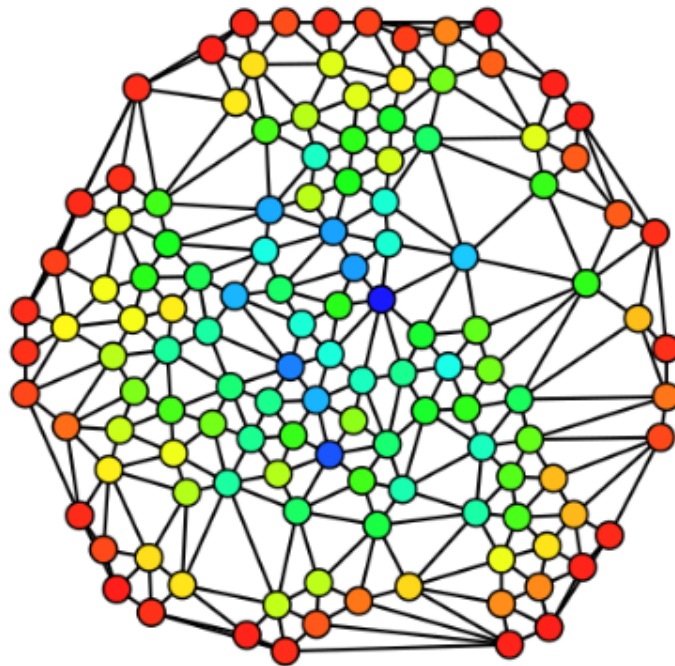
- Uma estrutura de dados para grafos
 - Matriz de adjacência ou lista de adjacência
- Um algoritmo de caminho mínimo
- O cálculo de *betweenness*
- Um algoritmo de desenho de grafos

Além disso, cada grupo deverá, também, propor cinco diferentes grafos a serem utilizados nos testes. Estes grafos deverão ser conexos, poderados e não-orientados. Cada grafo deverá ter entre 25 e 75 vértices. O peso das arestas, sua disposição e a densidade do grafo são de escolha do grupo. Recomenda-se criar os grafos o mais diferente o possível como forma de analisar bem o algoritmo proposto.

O que deve ser entregue: Cada grupo deverá desenvolver um documento *.pdf* contendo as seguintes sessões

1. Introdução (introduzir e definir o problema de centralidade em grafos)
2. Algoritmos (descrever os algoritmos utilizados e analisar sua complexidade)
3. Resultados (exibir a centralidade de cada um dos vértices do grafo, qual é o vértice mais central e uma figura do grafo mostrando os vértices, sua topologia e sua centralidade)

A figura deve ser semelhante a abaixo. Os vértices mais centrais são coloridos em azul, enquanto os menos centrais são coloridos em vermelho.



Além disso, os grupos também deverão montar uma apresentação de slides (também em formato *.pdf*) para apresentação em sala de aula no dias 07/06, sendo que a apresentação deverá durar entre 7 e 12 minutos.

Por fim, deverá ser entregue o código desenvolvido. Os algoritmo poderão ser implementados utilizando C, C++ ou python. O código deverá ser entregue em um único arquivo *.zip* contendo um cabeçalho com o nome dos integrantes do grupo

Método de entrega: Todos os três arquivos deverão ser entregues no Moodle da disciplina até as 8h00 do dia 07/06.

Método de avaliação: A apresentação corresponderá por 30% da nota total. De forma complementar, o outro documento *.pdf* corresponderá também por 30% da nota e o código corresponderá por 40% da nota.

Na apresentação, serão avaliados:

- Adequação ao tempo
- Postura dos apresentadores
- Assertividade na fala
- Corretude da prova apresentada
- Uso correto da língua portuguesa
- Qualidade dos slides

No documento *.pdf* com a descrição do problema, dos algoritmos e os resultados, serão avaliados:

- Uso correto da língua portuguesa
- Qualidade e clareza na apresentação dos algoritmos
- Análise correta das complexidades dos algoritmos
- Qualidade geral da comparação dos resultados; O uso de figuras e tabelas é necessário

No código serão avaliados:

- A qualidade e clareza do código
- Comentários explicativos
- Execução correta dos algoritmos

Ponto extra: Será acrescentado 10% no valor total do trabalho caso o documento com a descrição do problema, dos algoritmos e a análise dos resultados seja criado utilizando L^AT_EX