

Trabalho Prático - Grafos

UFMG - ICex - DCC

Projeto e Análise de Algoritmos

Setembro 2017

1 Introdução

Nos países que seguem o presidencialismo como forma de governo, é comum o presidente buscar na câmara dos deputados fortes aliados com capacidade de intermediar acordos entre diferentes grupos. Isso traz para o governo uma maior possibilidade de aprovação das pautas de seu interesse ao serem votadas por todos os deputados. Além disso, aumenta a governabilidade do presidente e fortalece as políticas públicas a serem implementadas. Assim, para mobilizar um maior número de políticos basta focar em deputados específicos, que tenham fortes capacidades de intermediação em um segundo deputado, onde esse segundo tem de um terceiro que o primeiro diretamente não tenha, e assim por diante.

Por exemplo, o líder de um dado partido político teoricamente tende a ter uma alta similaridade nas votações com os demais membros do partido, uma vez que é ele quem fornece as orientações de voto durante as votações. Assim, ao avaliar a similaridade do líder do partido com os demais membros, calculada pelo percentual de concordância nas votações que eles votaram juntos, é de se esperar que grande parte dos deputados possuam uma maior similaridade com o líder do partido. No entanto, entre si os demais deputados podem discordar em alguns momentos, tendo um menor percentual de similaridade que dificilmente os levariam a um acordo direto. Neste caso, o líder do partido pode ser um bom intermediador entre dois possíveis deputados que diretamente não possuem uma boa concordância a ponto de chegarem a um acordo.

Nesse contexto, esse trabalho tem como objetivo encontrar deputados com maior capacidade de intermediação com base na similaridade das suas votações.

2 Definição e Modelagem do Problema

Neste trabalho, nós estamos interessados em encontrar os deputados com um potencial de influência através de uma rede de similaridade de votações. Os deputados mais influentes são aqueles que fazem parte do maior número de interações possíveis entre quaisquer outros dois deputados. Além disso, eles são aqueles que uma vez desconsiderados, diminuirão as chances dos demais deputados estabelecerem um acordo entre si já que eles fazem parte de grande número de interações entre os demais deputados.

A rede de similaridade entre os deputados pode ser modelada utilizando a teoria dos grafos. Seja a tupla $G = (V, A)$ um grafo não direcionado com arestas ponderadas, onde $V = \{v_1, v_2, v_3, \dots, v_n\}$ é um conjunto de vértices que representa os deputados e $A = \{a_1, a_2, a_3, \dots, a_n\}$ um conjunto de arestas entre dois deputados v_i e v_j onde $i \neq j$, ponderada por w , tal que $0 \leq w \leq 100$ indica o percentual de similaridade entre os deputados v_i e v_j .

Exemplo: Considere o cenário onde é preciso escolher estrategicamente deputados membros de uma comissão para negociar um apoio. Tal comissão é composta por 5 deputados ativos que possuem ligações de acordo com a similaridade entre eles calculada ao longo de vários anos de votações, como mostra a Figura 1.

Observe que nem todas as arestas estão disponíveis, pois na vida real existem casos em que um deputado não tem a mínima chance de realizar um acordo diretamente com o outro. Por outro lado, alguns deputados podem intermediar esse acordo, por exemplo: Para os deputados 2, 3 e 4 poderem chegar a um acordo com o deputado 0, obrigatoriamente, eles precisarão da intermediação do deputado 1, que é o único deputado a ter um acesso direto com o deputado 0. Em um outro cenário considere um possível acordo entre o 2 e o 4, afim maximizar as chances dele ocorrer, é interessante que o 2 mobilize primeiro o 3, para que o 3 mobilize o 4, ao invés do 2 tentar mobilizar diretamente o 4 uma vez que eles possuem uma baixa similaridade. Observe também a importância do deputado 2 em intermediar possíveis acordos entre os pares

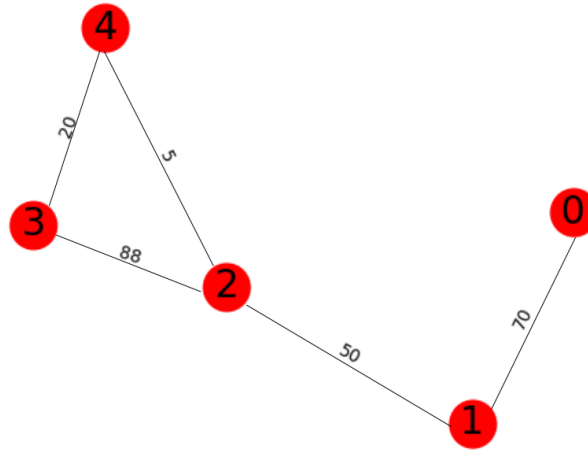


Figura 1: Exemplo 1

de deputados $(0;4)$, $(1;4)$, $(0;3)$, e $(1;3)$, e assim por diante.

Portanto, ao avaliar a capacidade de intermediação de cada deputado em todos os possíveis acordos, os deputados mais influentes, em ordem decrescente, são: 2, 1 e 3 (Empatados), 0 e 4 (Empatados). Em caso de empate, a ordem crescente do Id do deputado é preservada.

3 Implementação

As implementações devem ser testadas em uma máquina Linux do Departamento de Ciência da Computação de livre acesso aos alunos da pós-graduação via acesso remoto. Essa é a garantia de que a implementação será compilada e executada em um ambiente conhecido pelo aluno. As linguagens de programação aceitas incluem C++, Python e Java, **não sendo permitido o uso de bibliotecas de terceiros.**

O código implementado deverá ser compilado (se for o caso) da seguinte maneira:

```
$ ./compilar.sh
```

O *shell script* 'compilar.sh' deverá conter:

```
#!/bin/bash
```

```
<coloque aqui o código para compilar seu programa>
```

O código implementado deverá ser executado com a seguinte linha de comando:

```
$ ./executar.sh entrada saida
```

O shell script "executar.sh" deverá conter a seguinte estrutura:

```
#!/bin/bash

in=$1

out=$2

<coloque aqui o código para executar o seu programa
de modo que a entrada seja lida de $in e a saída seja
escrita em $out>
```

4 Entrada e Saída

O arquivo de entrada deverá ser lido de forma que uma linha contenha a seguinte estrutura $\langle Id, Id, Similaridade \rangle$, onde os dois primeiros campos indicam dois diferentes *Ids* de dois deputados e o terceiro o percentual de similaridade entre eles. Para indicar o fim da entrada de dados deverá ser lido uma linha no seguinte padrão $\langle 0, 0, 0 \rangle$.

A saída esperada contém os *Ids* dos deputados ordenados pela capacidade de intermediação, separados por vírgula em uma única linha. Em caso de empate, a ordem crescente do Id deverá ser preservada, como no exemplo apresentado.

A entrada e saída da Figura 1, são respectivamente:

Entrada:

0,1,70

1,2,50

2,3,88

3,4,20

4,2,5

0,0,0

Saída:

2,1,3,0,4

Estão disponíveis 5 casos de testes nos arquivos em anexo.

5 Documentação e Entrega

A documentação não deverá exceder o limite de **10 páginas**, sendo submetida no formato PDF juntamente com os *shell scripts* e o código fonte via *minha.ufmg* até o dia **16 de Outubro de 2017**. Além disso, a documentação deverá contemplar os seguintes requisitos:

- Introdução com uma explicação clara e objetiva de como o problema que foi resolvido, justificando os algoritmos e as estruturas de dados utilizadas. Para auxiliar nessa atividade utilize pseudocódigos, diagramas e demais figuras que achar conveniente. Não é necessário incluir trechos de código da sua implementação e nem mostrar maiores detalhes da sua implementação, exceto quando esses influenciam no seu algoritmo principal;
- Análise de complexidade de tempo e espaço usando o formalismo da notação assintótica;
- Análise experimental que avalie o tempo de execução do seu código em função das características da entrada, tais como, o número de vértices e de arestas. Cabe a você gerar as entradas para esses experimentos. A apresentação dos resultados pode ser feita em gráficos e tabelas que achar conveniente juntamente como as interpretações obtidas neles.

Bom Trabalho!