



Trabalho 3 — Implementação e análise empírica do algoritmo de Karger

Neste trabalho, vocês deverão implementar, em qualquer linguagem de programação, o algoritmo de Karger para a determinação de um corte mínimo em um grafo simples. O programa de vocês fará a leitura de um grafo informado através de um arquivo de texto. O formato do arquivo é o mesmo dos trabalhos anteriores:

Formato do arquivo

```
n #numero de vertices
0 1 0 ... 1 #linha da matriz de adjacencias relativa ao vertice 1
1 0 1 ... 0 #linha da matriz de adjacencias relativa ao vertice 2
...
1 0 0 ... 0 #linha da matriz de adjacencias relativa ao vertice n
```

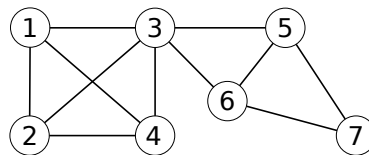


Figura 1: Um grafo simples com 7 vértices.

Para o grafo ilustrado na Figura 1, por exemplo, o arquivo de entrada seria o seguinte:

Arquivo para o grafo da Figura 1

```
7
0 1 1 1 0 0 0
1 0 1 1 0 0 0
1 1 0 1 1 1 0
1 1 1 0 0 0 0
0 0 1 0 0 1 1
0 0 1 0 1 0 1
0 0 0 0 1 1 0
```

O trabalho terá as três partes descritas a seguir.

Implementação do algoritmo de Karger. Nesta primeira parte, deve-se apenas implementar o algoritmo. Como referência, indico o seguinte material:

<https://www.dropbox.com/s/ari9bf7zlc28j/Lecture16-Stanford-Karger.pdf?dl=0>.

Implementação de uma abordagem randomizada ingênua. O algoritmo de Karger, apesar de tomar decisões randomizadas, não se trata de uma abordagem ingênua, pois, à medida em que as iterações são executadas, aumenta-se a probabilidade de se realizar contrações que retirem muitas arestas do corte final (pense sobre esta afirmação). Isso ficará evidente ao compará-lo à abordagem ingênua que simplesmente particiona aleatoriamente os vértices do grafo em dois conjuntos. Esta segunda parte consiste em implementar tal abordagem ingênua.

Análise empírica dos algoritmos. Serão compartilhados com a turma exemplos de grafos e um dos seus respectivos cortes mínimos (um grafo pode ter mais de um corte mínimo). Para cada grafo, deve-se construir um gráfico com duas curvas, uma para cada algoritmo implementado. É importante frisar que uma execução de um dos algoritmos possui N_{iter} iterações, e que será retornado o menor corte encontrado nessas iterações. Nos gráficos em questão, o eixo das abscissas conterá o número de iterações N_{iter} , enquanto que o eixo das ordenadas apresentará a probabilidade p do respectivo algoritmo encontrar o corte mínimo. Essa probabilidade



será estimada executando-se o algoritmo N_{exec} vezes. Sugere-se $N_{exec} = 10000$. A faixa de valores considerados para o N_{iter} deve ser calibrada pelo discente. A dica é parar de aumentar o valor assim que a probabilidade estiver muito próxima de 100%.

Casos de teste. https://www.dropbox.com/sh/b9vcwq0u3rj516o/AAA1vn2_ONQ7srmq5s2KaVwla?dl=0

Encontrando o corte mínimo. Para encontrar o corte mínimo das instâncias compartilhadas com a turma, foi utilizado um modelo de programação de linear inteira (assunto da disciplina Pesquisa Operacional). Considere $G = (V, E)$ o grafo de entrada. Define-se, para cada vértice $j \in V$, uma variável binária x_j que determina em qual partição o vértice está. Além disso, para cada aresta $ij \in E$, tem-se uma variável binária y_{ij} que assume o valor um se, e somente se, a aresta correspondente estiver no corte. O modelo é dado a seguir.

$$\min \sum_{ij \in E} y_{ij} \quad (1)$$

$$\text{s.a.} \quad \sum_{j \in V} x_j \leq |V| - 1 \quad (2)$$

$$\sum_{j \in V} x_j \geq 1 \quad (3)$$

$$y_{ij} \geq x_i - x_j, \quad \forall ij \in E \quad (4)$$

$$y_{ij} \geq x_j - x_i, \quad \forall ij \in E \quad (5)$$

$$x_j \in \{0, 1\}, \quad \forall j \in V \quad (6)$$

$$y_{ij} \in \{0, 1\}, \quad \forall ij \in E. \quad (7)$$

É verdade que há algoritmos disponíveis para a determinação do corte mínimo que apresentariam tempos computacionais bem mais competitivos que o do modelo acima. Escolheu-se a apresentação do modelo apenas para refrescar o tema na mente dos discentes.

Outras informações. O trabalho pode ser feito individualmente ou em dupla. O código deve ser enviado através da tarefa cadastrada no sigaa por apenas um membro da equipe.