Teoria dos Grafos

Grafos, segunda parte: Grafos Eulerianos e o Algoritmo de Dijkstra

Leon Ferreira Bellini

RA 22218002–8

e

Guilherme Ormond Sampaio

RA 22218007–7

1 Introdução

Desde a elaboração do primeiro relatório, conteúdo relevante foi acrescentado ao nosso repertório, como caminhos, trilhas e passeios por um grafo, árvores, grafos conexos e grafos isomorfos, contudo, os objetos em foco são os grafos Eulerianos e o algoritmo de Dijsktra, o qual propõe uma resolução para o problema do caminho de peso mínimo.

2 Conceitos Eulerianos

Muitos dos teoremas inicialmente denotados por **Leonard Paul Euler** se utilizam dos conceitos relacionados a passeios e suas derivações, como o **Tour de Euler**, este sendo diretamente ligado ao **grafo de Euler** e por fim, **percurso pré-Euleriano**. Serão explicados a seguir:

2.1 O Tour ou cadeia de Euler

Passeios de uma forma geral, podem ser abertos ou fechados, tours são caminhos fechados os quais incluem toda aresta do grafo em questão, já o **Tour de Euler** requere que a aresta apareça apenas uma vez no passeio fechado.

2.2 Grafo Euleriano

Um grafo **G** = (V, E) é euleriano se apresenta um ciclo (caminho fechado) que contenha todas as arestas deste **G**. A presença de um tour de euler também configura um grafo como euleriano. Outro importante fato a se notar é o que grafos eulerianos não possuem graus ímpares e a presença de uma trilha de euler não o torna um grau euleriano se este possuir grau (s) ímpar (es).

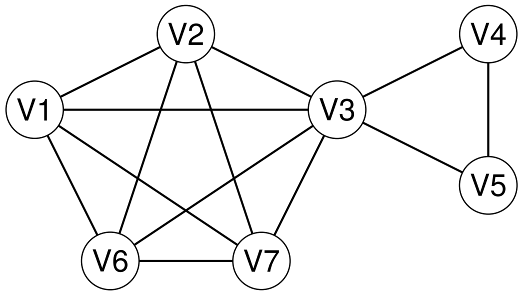

Figura 1: Exemplo de grafo Euleriano, note que todos os vértices têm índice par

2.3 O percurso pré-euleriano

Desta vez relacionado à quantidade de arestas, é o percurso que passa em todas arestas do grafo **G**.

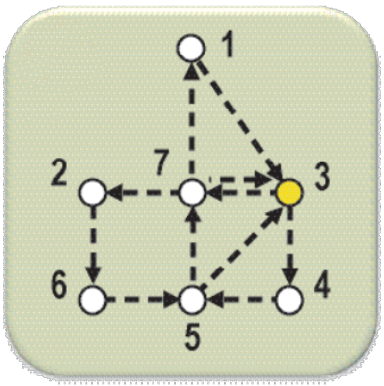

Figura 2: Grafo de sequência 3-7-1-3-4-5-3-7-2-6-5-7-3 se configura como pré euleriano – Grafos: Conceitos, algoritmos e aplicações (2014)

3 O algoritmo de Dijkstra

Um grafo **G** = (V, E) é ponderado quando pesos são dados a cada aresta. Também chamados de custos, o intuito é definir a partir de valores numéricos a disponibilidade, tempo, dificuldade, etc. de ir de uma aresta para outra.

Teorizado pelo holandês Edsger Dijkstra, teria o objetivo de resolver problemas relacionados ao caminho mais curto num grafo simples direcionado ou não. Deve ser aplicado em grafos que contenham arestas de peso não negativo, porém, é possível adicionar um valor único que deixe todas arestas positivas.

Partindo do pressuposto que os pesos p (e) são maiores ou iguais a zero, o algoritmo se baseia nos seguintes passos:

1. Considerar "i" como sendo o contador para índice do vértice atual e "S" como o subconjunto dos vértices de G, i inicia em 0 e S contendo o primeiro vértice definido no passo a seguir;

2. Definir o vértice inicial v0 (vértice de onde o algoritmo partirá) como rótulo zero, todos os outros vértices devem receber o rótulo infinito;

3. Para todos os vértices adjacentes à v0, definir seus rótulos com o mesmo peso das arestas que os ligam, isto se dá ao fato que qualquer valor de cada aresta será menor que o valor de infinito ∞;

4. O vértice adjacente cujo caminho é o mais curto será o escolhido para a próxima iteração, "i" é iterado. O vértice atual é adicionado à lista de vértices visitados;

5. É repetido o processo de rotular os vértices adjacentes ainda não visitados, porém, se o peso da aresta que liga o vértice da iteração atual e o vértice adjacente for menor que o valor rotulado deste, seu rótulo é substituído pela soma de pesos atual;

6. Repetir até não repetir mais vértices, com "i" sendo igual à quantidade de vértices de **G** — 1, ou ter chegado ao vértice desejado;

4 Conclusão

Ambos os temas abordados neste relatório são de extrema importância para a estruturação de dados e técnicas relacionadas, tendo o algoritmo de Dijkstra como destaque. Sua utilidade varia desde a aplicação em algoritmos ainda mais extensos para redes sociais até engenharia de tráfego e algoritmos que tenham a função de determinar o caminho mais curto de uma rede de estradas. Apesar disso, o procedimento ainda apresenta uma ineficiência quanto a grafos não direcionados ou grafos que possuam *dead ends*.

5 Referências bibliográficas

GOLDBARG, Marco e Elizabeth. **Grafos: Conceitos, algoritmos e aplicações.**1.ed.- São Paulo: Elsevier,2012.

Dijkstra Algorithm. Direção de Computerphile. Inglaterra: Setor de Ciência da computação da Universidade de Nottingham, 2017. Disponível em: <https://www.youtube.com/watch?v=GazC3A4OQTE>. Acesso em 18 mai. 2019.

Grafos Eulerianos. Universidade Federal de Santa Catarina. Disponível em: <http://www.inf.ufsc.br/grafos/temas/euleriano/euleriano.htm>. Acesso em 18 mai. 2019.