Trabalho de Teoria dos Grafos

Bruno Ramos Madson Araújo Nemuel Leal
Nilton Vasques

April 2, 2013

1 Árvore Geradora Mínima

Definição: Seja G=(V,E) um grafo não direcionado e conexo, G'=(V,E') é chamado de subgrafo gerador se possuí os mesmos vértices de G. Portanto se tivermos em G' uma árvore, então o subgrafo é uma árvore geradora. Quando G é um grafo conexo, em que cada aresta possui um valor ou peso p(e), o peso total da árvore geradora é

$$\sum_{e \in E'} p(e)$$

onde p(e) é uma função que retorna o peso da aresta *e*. Á árvore geradora mínima é a árvore G' que possui o menor peso total dentre todas as árvores possíveis do grafo G[2]. Podemos enunciar a função para encontrar a árvore geradora mínima como

$$min \sum_{e \in E'} p(e)$$

. A partir dessa noção podemos visualizar que encontrar a árvore geradora mínima não é tão trivial assim. Se propormos uma solução pela força bruta, ou seja, encontrar todas as árvores geradoras e assim então verificar qual a que possui o menor peso total. No pior caso quando temos um grafo completo(em que todos os vértices se ligam uns aos outros) teríamos n^{n-2} árvores geradoras onde n é o número de nós, sendo assim teríamos uma solução em tempo exponencial $O(n^n)$ e inviável . Diante deste cenário alguns matemáticos elaboram soluções para o problema das Árvores Geradoras Mínimas, se utilizando de heurísticas gulosas para encontrar a solução ótima. No presente artigo abordaremos o Algoritmo de Kruskal e o de Prim, como estudo de caso .

1.1 Algoritmo de Kruskal

O algoritmo de Kruskal é um algoritmo guloso, que tem por objetivo encontra uma árvore geradora mínima para um grafo conexo e valorado (com pesos nas arestas). Vale ressaltar que para árvores não conexas, o algoritmo encontra floresta geradora mínima, ou seja uma árvore geradora mínima para cada componente conexo do grafo.

1.1.1 Pseudo-código

1. Criar uma floresta F, onde cada vértice do grafo é uma árvore separada.
2. Criar um conjunto S contendo todos as arestas do grafo.
3. Enquanto S é não vazio:
   (a) Remova um aresta e com peso mínimo de S.
   (b) Se e conecta duas diferentes árvores, então adicione e para floresta F.
   (c) Caso contrário, discarte e, ou seja se a escolha de e gera um circuito em F, discarte-a.

1.1.2 Implementação

A implementação se utilizou da estrutura de dados UnionFind, para realizar as operações dos passos listados no pseudo código. No item *3.* foi necessário ordernar as arestas pelo seu peso mínimo. Como algoritmo de ordenação utilizamos o QuickSort que ordena uma lista de números em tempo $\Theta(n \log n)$.

1.1.3 Análise de Complexidade

A estrutura de dados UnionFind mantém um conjunto de elementos particionados em vários subconjuntos não sobrepostos. O algoritmo que controla essa estrutura possui duas operações principais:

- Find: Determina de qual subconjunto um elemento pertence.
- Union: Faz a união de dois subconjuntos em um só subconjunto.

É interessante notar que com a operação Find, podemos verificar se dois elementos estão no mesmo conjunto, no caso particular do algoritmo de Kruskal, usando Find podemos decidir se a inclusão de uma aresta e forma um circuito ou não. Segundo [1] a complexidade do algoritmo UnionFind na sua forma simples é $\Theta(n)$ por operação. Sendo assim a complexidade da nossa implementação é $\Theta(E \log E)$.

1.2 Algoritmo de Prim

O algoritmo de Prim...

References

[1] Thomas H. Cormem, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*, volume ISBN 0-262-03293-7, chapter 21: Data structures for Disjoint Sets, pages 498–524. MIT Press, second edition, 2001.

[2] Fernando Nogueira. Problema da Árvore Gerador Mínima. *UFJF*.

[3] F. Prado, T. Almeida, and V. N. Souza. Introdução ao Estudo sobre Árvore Geradora Mínima em Grafos com Parâmetros Fuzzy. *UNICAMP - Faculdade de Engenharia Elétrica*.