

ÁRVORE GERADORA MÍNIMA

Um problema encontrado ao se projetar redes é como conectar todos os nós eficientemente, onde os nós podem ser computadores, telefones, depósitos, etc. Uma árvore geradora mínima pode fornecer uma solução econômica, uma que necessite da menor quantidade de cabos, tubos, ou qualquer que seja o material usado para a conexão. Por questões de segurança, no entanto, a árvore geradora mínima seria aumentada, em geral, com arcos adicionais de modo que, se alguma conexão for quebrada por alguma razão, uma rota alternativa poderia ser encontrada.

Definição: Árvore Geradora Uma árvore geradora para um grafo conexo é uma árvore sem raiz cujo conjunto de nós coincide com o conjunto de nós do grafo e cujos arcos são (alguns dos) arcos do grafo.

Uma árvore geradora, portanto, conecta todos os nós de um grafo sem arcos em excesso (e sem ciclos). Existem algoritmos para a construção de uma árvore geradora mínima, uma árvore geradora de peso mínimo, para um grafo dado simples e conexo com peso.

Um desses algoritmos, chamado de algoritmo de Prim, funciona de maneira muito parecida com o algoritmo de caminho mínimo. Existe um conjunto IN que contém, inicialmente, um nó arbitrário. Para cada nó z não pertencente a IN, guardamos a distância mínima $d[z]$ entre z e qualquer nó em IN. Incluímos, sucessivamente, nós em IN, onde o próximo nó a ser incluído é um que não pertencia a IN e cuja distância $d[z]$ é mínima. O arco tendo essa distância mínima torna-se, então, parte da árvore geradora. Como pode haver distâncias mínimas iguais, a árvore geradora mínima pode não ser única. O algoritmo termina quando todos os nós pertencerem a IN.

A diferença fundamental entre as implementações dos dois algoritmos está no cálculo das novas distâncias para os nós que ainda não pertencem a IN. No algoritmo de Dijkstra, se p é o nó que acabou de ser incluído em IN, as distâncias para os nós que não estão em IN são recalculadas por

$$D[z] = \min(d[z], d[p] + A[p, z])$$

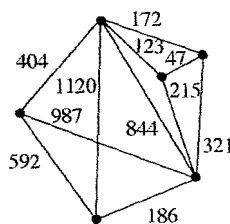
Isto é, comparando a distância atual de x a z com a distância de x a p mais a distância de p a z . No algoritmo Prim, se p é o nó que acabou de ser incluído em IN, as distâncias para os nós que não estão em IN são recalculadas por

$$D[z] = \min(d[z], A[p, z])$$

Isto é, comparando a distância atual de z a IN com a distância de p a z .

Exercício

Instrumentos sísmicos devem ser distribuídos em uma zona de fendas vulcânicas, como ilustrado abaixo, onde estão indicadas as distâncias em metros entre os locais. (As distâncias entre alguns dos locais não são dadas devido a acidentes naturais que impediriam uma conexão direta.) Qual a maneira mais econômica de colocar os dispositivos de modo que todos estejam conectados? Qual a quantidade total de cabos envolvidos?



ALGORITMO DE KRUSKAL

O algoritmo de Kruskal é um outro algoritmo para encontrar uma árvore geradora mínima em um grafo conexo. Enquanto o algoritmo de Prim faz a árvore “crescer” a partir de um ponto arbitrário inicial incluindo arcos adjacentes associados a distâncias pequenas, o algoritmo de Kruskal inclui arcos em ordem crescente de distância onde quer que estejam no grafo. Empates são resolvidos arbitrariamente. A única restrição é que um arco não é incluído se a sua inclusão criar um ciclo. O algoritmo termina quando todos os nós estiverem incorporados em uma estrutura conexa. Uma descrição em pseudocódigo é dada a seguir:

```
OutraAGM ( matriz n X n A; coleção de arcos T)
//Algoritmo de Kruskal para encontrar uma árvore geradora mínima;
//inicialmente, T é vazio; ao final, T = árvore geradora mínima.
    ordene os arcos em G por distância em ordem crescente
    repita
        se próximo arco na ordem não completa um ciclo então
            inclua esse arco em T
        fim do se
    até T ser conexo e conter todos os nós em G
fim de OutraAGM
```