Introdução à Teoria dos Grafos

Bacharelado em Ciência da Computação–UFMS, 2005

ÁRVORE GERADORA DE CUSTO MÍNIMO

Resumo

No Capítulo 3 – Árvores, estudamos muitas propriedades importantes sobre esses grafos especiais. Um problema que aparece com freqüência na prática é o de encontrar uma árvore geradora de custo mínimo. Em geral, temos um grafo conexo com custos associados às suas arestas e queremos encontrar um subgrafo gerador deste grafo que não contenha circuitos e tenha custo mínimo. Este problema foi inicialmente proposto por Borůvka [1,2] em 1926, quando estava trabalhando na construção de uma rede de eletrificação rural da cidade de Morávia do Sul, na antiga Tchecoslováquia.

1 Motivação

Suponha que um grupo de voluntários tenha se prontificado a realizar trabalhos de assistência social aos habitantes de diversas cidades miseráveis do nordeste. Idealmente, linhas telefônicas devem ser construídas ao longo de algumas estradas existentes que conectam essas vilas umas com as outras. Desejamos, na verdade, construir linhas telefônicas de tal forma que todo par de cidades possa se comunicar por telefone. Além disso, o total de quilômetros de linhas telefônicas deve ser minimizado. Ou seja, o problema é determinar ao longo de quais estradas estas linhas telefônicas devem ser construídas para que se obtenha o sistema de telecomunicações desejado.

2 Modelagem do problema em teoria dos grafos

A construção de tal rede de telecomunicações tem uma representação gráfica óbvia. Podemos associar um grafo a esta descrição, onde cada vértice corresponde a uma cidade e uma aresta entre dois vértices representa uma estrada entre as cidades correspondentes. O comprimento de tal estrada é indicado no grafo pela atribuição de um custo à aresta correspondente. Assim, estaremos trabalhando com um **grafo com custos nas arestas** G.

Seja G um grafo com custos nas arestas dado pela função $c\colon E_G \to \mathbb{R}_{\geqslant 0}$. Seja $X \subseteq E_G$ e defina $c(X) = \sum_{e \in X} c(e)$ o custo do conjunto de arestas X em G. O **custo** $c(H)$ **de um subgrafo** H de G é dado então por $c(H) = c(E_H)$. A solução de nosso problema consiste em encontrar um subgrafo gerador conexo T do grafo com custos G, com o menor custo possível. O subgrafo gerador T é necessariamente uma árvore, já que sabemos que T é conexo. Além disso, se T contém um circuito, então a remoção de qualquer aresta deste circuito resulta em um subgrafo conexo de menor custo. Assim, T não pode conter circuito e portanto é uma árvore. Dessa forma, uma rede de telecomunicações desejada — contendo a quantidade mínima de quilômetros de linhas telefônicas — corresponde a uma árvore T de G com custo mínimo. Tal árvore é chamada uma **árvore geradora de custo mínimo**. Nos deparamos então com o seguinte problema:

> **Problema** MST(G, c): *dado um grafo G com custos nas arestas atribuídos por uma função custo $c\colon E_G \to \mathbb{R}_{\geqslant 0}$, encontrar uma árvore geradora de custo mínimo T de G.*

O problema da árvore geradora de custo mínimo, um dos mais bem conhecidos problemas em otimização combinatória, parece ter sido originalmente proposto por Borůvka [1,2] em 1926, quando considerava a eletrificação rural da cidade de Moravia do Sul, antiga Tchecoslováquia. Graham e Hell [4] discutem este problema e sua história. Um número razoável de soluções algorítmicas para este problema tem sido proposto, com talvez o mais famoso deles devido a Kruskal [5]. Borůvka [1,2] e Prim [6] também propuseram bons algoritmos ligeiramente diferentes entre si e entre o algoritmo de Kruskal. Uma descrição mais acessível destes algoritmos pode ser encontrada em [3].

3 Algoritmo de Kruskal

O objetivo do algoritmo de Kruskal é selecionar sucessivamente arestas de menor custo de um grafo conexo com custos, sem gerar circuitos e até que uma árvore tenha sido produzida. O algoritmo de Kruskal é um **algoritmo guloso**, já que ele seleciona repetidamente uma aresta de menor custo das arestas de entrada restantes, sem que um circuito seja produzido.

KRUSKAL(G, c): recebe um grafo G e uma função custo $c\colon E_G \to \mathbb{R}_{\geqslant 0}$ e devolve uma árvore geradora de custo mínimo T de G.

1: $\{$*Inicialize o conjunto S, que consistirá das arestas de uma árvore geradora de custo mínimo.*$\}$
$S \leftarrow \emptyset$.
2: $\{$*O conjunto S é incrementado.*$\}$
Seja e uma aresta de menor custo tal que $e \notin S$ e suponha que $G[S \cup \{e\}]$ não contém circuitos. Faça $S \leftarrow S \cup \{e\}$.
3: $\{$*Este passo determina se uma árvore geradora mínima foi construída.*$\}$
Se $|S| = p - 1$, então devolva $T = G[S]$; caso contrário, vá para o Passo 2.

Podemos agora, como exemplo, aplicar o algoritmo de Kruskal ao grafo com custos G da figura 1. Observe que as arestas de G devem estar ordenadas em ordem não descrescente de seus custos para que o algoritmo funcione satisfatoriamente.

É fácil realizar a análise deste algoritmo. Em um primeiro passo de pré-processamento, as arestas de um grafo de entrada G de ordem p e tamanho q devem ser ordenadas de tal forma

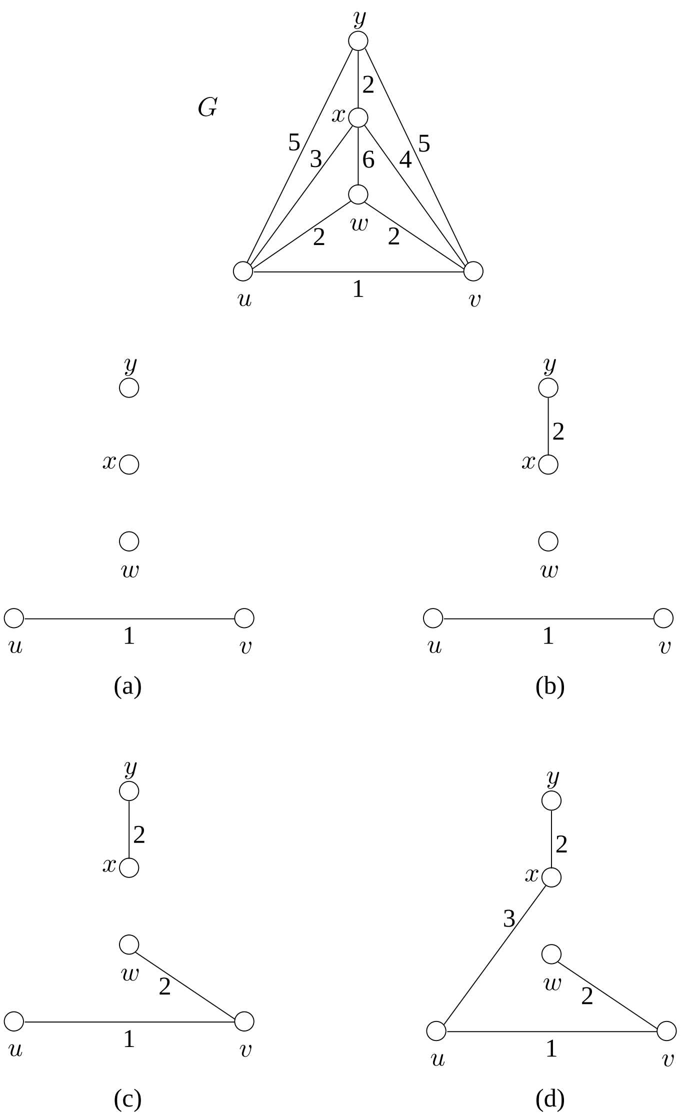

Figura 1: Construindo uma árvore geradora de custo mínimo com o algoritmo de Kruskal.

que
$$c(e_1) \leqslant c(e_2) \leqslant \ldots \leqslant c(e_q).$$
O pré-processamento pode então ser executado em tempo $O(q \log q)$. Os Passos 1, 2 e 3 do algoritmo podem ser executados em tempo constante. No entanto, os Passos 2 e 3 são executados sobre cada aresta de G, e portanto, o tempo de execução nestes dois passos é $O(q)$. Logo, o tempo de execução total do algoritmo de Kruskal é, na verdade, equivalente ao tempo gasto em seu pré-processamento, ou seja, $O(q \log q)$.

Podemos demonstrar que o grafo $G[S]$ obtido pelo algoritmo de Kruskal, a partir de um grafo não trivial, conexo e com custos G, é de fato uma árvore geradora de custo mínimo. Este resultado é apresentado a seguir e sua demonstração pode ser encontrada em [3].

TEOREMA 3.1 *O algoritmo de Kruskal produz uma árvore geradora de custo mínimo de um grafo não trivial, conexo e com custos nas arestas.*

PROVA.

Seja G um grafo conexo não trivial com custos nas arestas dados pela função custo c, de ordem p, e seja T um subgrafo produzido pelo algoritmo de Kruskal. Certamente, T é uma árvore geradora de G com tamanho $p-1$ e $E_T = \{e_1, e_2, \ldots, e_{p-1}\}$, onde $c(e_1) \leqslant c(e_2) \leqslant \ldots \leqslant c(e_{p-1})$. Portanto, o custo de T é
$$c(T) = \sum_{i=1}^{p-1} c(e_i).$$

Suponha, pelo contrário, que T não é uma árvore geradora de custo mínimo de G. Então, escolha uma árvore T' entre as árvores geradoras de G que tem número máximo de arestas em comum com T. Como as árvores T e T' não são idênticas existe pelo menos uma aresta de T que não pertence a T'. Seja e_i, com $1 \leqslant i \leqslant p-1$, a primeira aresta de T que não está em T' e defina $G_0 = T' + e_i$. Então, G_0 tem exatamente um circuito C. Como T não tem circuitos, existe uma aresta e_0 de C que não está em T. O grafo $T_0 = G_0 - e_0$ é também uma árvore geradora de G e
$$c(T_0) = c(T') + c(e_i) - c(e_0).$$

Como $c(T') \leqslant c(T_0)$, segue que $c(e_0) \leqslant c(e_i)$. Pelo algoritmo de Kruskal, e_i é uma aresta de custo mínimo tal que o subgrafo $G[\{e_1, e_2, \ldots, e_{i-1}\} \cup \{e_i\}]$ não contém circuitos. Portanto, $G[\{e_1, e_2, \ldots, e_{i-1}, e_0\}]$ é um subgrafo de T' que não contém circuitos e $c(e_i) = c(e_0)$. Assim, $c(T_0) = c(T')$ o que implica que T_0 é também uma árvore geradora de custo mínimo de G. Mas T_0 tem mais arestas em comum com T que T' com T, o que contradiz a nossa suposição inicial. ❑

Referências

Este texto foi produzido com a consulta à referência [3]. Os artigos originais que propõem os algoritmos para encontrar uma árvore geradora de custo mínimo de Borůvka, Kruskal e Prim estão listado a seguir em [1,2], [5] e [6], respectivamente. Uma história sobre o problema da árvore geradora de custo mínimo é descrita em [4].

[1] O. Borůvka, *O jistém problému minimálním*, (Czech), Acta Societ. Scient. Natur. Moravicae, 3, pp. 37-58, 1926.

[2] O. Borůvka, *Príspevek k reseníotázky ekonomické stavby elektrovodnich sítí*, (Czech), Elecktrotechniky obzor, 15, pp. 153-154, 1926.

[3] G. Chartrand and O. R. Oellermann, *Applied and Algorithmic Graph Theory*, McGraw-Hill, Inc., 1993.

[4] R. L. Graham and P. Hell, *On the history of the minimum spanning tree problem*, Ann. History Comput., pp. 43-57, 1985.

[5] J. B. Kruskal, *On the shortest spanning tree of a graph and the traveling salesman problem*, Proc. Amer. Math. Soc., 7, pp. 48-50, 1956.

[6] R. C. Prim, *Shortest connection networks and some generalizations*, Bell Syst. Tech. J., 36, pp. 1389-1401, 1957.