

Corte Multiseparador Mínimo

Notas de aula de MAC-5727

(Material Extra do Capítulo 2)

Prof.^a Yoshiko Wakabayashi

— Versão pós-aula produzida por Fábio A. C. Tisovec em agosto de 2006 —

16 de agosto de 2016

Problema MINCMS (G, c, S) : Dados um grafo conexo $G = (V, E)$ com custo $c_e \in \mathbb{Q}_{\geq}$ para cada $e \in E$, e um conjunto $S = \{s_1, s_2, \dots, s_k\} \subset V$, encontrar $C \subseteq E$, tal que $c(C)$ seja mínimo e C separa 2 a 2 todos os vértices de S .

Num grafo $G = (V, E)$, dizemos que $C \subseteq E$ separa os vértices de um conjunto $S \subset V$ se em $G - C$ cada elemento de S pertence a um componente distinto (ou seja, em $G - C$ não existe nenhum caminho entre os vértices de S). Também dizemos que C é um S -corte multiseparador (*multiway cut*).

Dahlhaus, Johnson, Papadimitriou, Seymour e Yannakakis [2] mostraram que o MINCMS é NP-difícil para $k \geq 3$ (mesmo para grafos planares com custos unitários).

Esses autores também mostraram que MINCMS (G, c, S) é polinomial para k fixo (existe algoritmo com complexidade $O((4k)^k n^{2k-1} \log n)$). Note que no caso $k = 2$ temos o bem conhecido problema do corte de custo mínimo (que separa 2 vértices), que pode ser resolvido eficientemente com algoritmos para problemas de fluxo.

Dahlhaus *et al.* [2] provaram que existe uma $(2 - 2/k)$ -aproximação polinomial para o MINCMS (caso geral). Veremos isso a seguir.

Algoritmo GULOSO-MINCMS (G, c, S)

- 1 Para $i = 1, 2, \dots, k$ faça: **bibi**
- 2 $C_i \leftarrow$ CORTEMÍNIMOSEPARADOR (G, c, S, s_i)
- 3 Seja j tal que $c(C_j) = \max\{c(C_i), i = 1, \dots, k\}$.
- 4 Devolva $C_1 \cup \dots, C_{j-1} \cup C_{j+1} \cup \dots, C_k$

No passo 2 o algoritmo CORTEMÍNIMOSEPARADOR (G, c, S, s) devolve um conjunto de arestas C , tal que em $G - C$ não existe caminho de s a $S \setminus \{s\}$ (ou seja, C é um corte que separa s dos vértices de $S \setminus \{s\}$). Dahlhaus *et al.* [2] propõem a seguinte maneira para encontrar tal corte: construir um novo grafo G' a partir de G

identificando-se todos os vértices em $S \setminus \{s\}$ num único vértice s' , e encontrar em G' um corte de custo mínimo que separa s e s' .

Uma outra maneira para se resolver o passo 2 do algoritmo é obter um novo grafo acrescentando-se um novo vértice t e arestas wt para todo $w \in S \setminus \{s\}$, definir custo $c_{wt} = \infty$, e encontrar um corte de custo mínimo que separa s e t .

Podemos concluir imediatamente do passo 2 do algoritmo o seguinte resultado.

Lema 1: *O algoritmo GULOSO-MINCMS (G, c, S) devolve um S -corte multiseparador.*

Teorema 2: *O algoritmo GULOSO-MINCMS (G, c, S) é uma $(2 - 2/k)$ -aproximação polinomial para o problema MINCMS (G, c, S) , onde $k = |S|$.*

Demonstração: Considere uma dada instância (G, c, S) do MINCMS e a solução $C := (\cup_{i=1, i \neq j}^k C_i)$ devolvida pelo algoritmo GULOSO-MINCMS (G, c, S) .

Seja C^* uma solução ótima para essa instância. Então $G - C^*$ tem k componentes, digamos $G_i = (V_i, E_i)$, $i = 1, \dots, k$, tais que $s_i \in V_i$.

Claramente, $\delta(V_i)$ é um corte que separa s_i de $S \setminus \{s_i\}$. Logo,

$$c(C_i) \leq c(\delta(V_i)), \text{ para todo } 1 \leq i \leq k,$$

e portanto,

$$\sum_{i=1}^k c(C_i) \leq \sum_{i=1}^k c(\delta(V_i)) = 2c(C^*). \quad (1)$$

Como $c(C_j) = \max\{c(C_i), i = 1, \dots, k\}$, é imediato que

$$c(C_j) \geq \sum_{i=1}^k \frac{c(C_i)}{k}. \quad (2)$$

Então, usando (2) e (1) nas desigualdades abaixo, obtemos

$$\begin{aligned} c(C) &= \sum_{i=1}^k c(C_i) - c(C_j) \\ &\leq \sum_{i=1}^k c(C_i) - \sum_{i=1}^k \frac{c(C_i)}{k} \\ &= (1 - 1/k) \sum_{i=1}^k c(C_i) \\ &\leq (1 - 1/k) 2c(C^*) \\ &= (2 - 2/k) \text{opt}(G, c, S). \end{aligned}$$

O algoritmo é claramente polinomial, pois o algoritmo usado no passo 2 pode ser executado em tempo polinomial.

□

Em 1998, Călinescu, Karloff e Rabani [1] obtiveram uma $(3/2 - 1/k)$ -aproximação probabilística pra o MINCMS (G, c, S) , e em seguida uma versão desaleatorizada com a mesma razão. A idéia consiste em formulá-lo como um programa linear e, usando arredondamento probabilístico, decidir quais vértices ficam no mesmo componente de s_i , para todo i .

Referências

- [1] G. Călinescu, H. Karloff, and Y. Rabani. An improved approximation algorithm for MULTIWAY CUT. *Journal of Computer and System Sciences*, 60(3):564–574, 2000. [30th Annual ACM Symposium on Theory of Computing (Dallas, TX, 1998)].
- [2] E. Dahlhaus, D.S. Johnson, C.H. Papadimitriou, P.D. Seymour, and M. Yannakakis. The complexity of multiterminal cuts. *SIAM Journal on Computing*, 23(4):864–894, 1994.