Caminhos disjuntos nas arestas

Prof.ª Yoshiko Wakabayashi
Notas de aula de MAC-5727 produzida por Cristiane M. Sato em agosto/2006

20 de março de 2008

Problema MAXCDA (G, T): *Dados um grafo conexo $G = (V, E)$ e um conjunto $T = \{(s_1, t_1), \ldots, (s_k, t_k)\}$ tal que $s_i, t_i \in V$ para todo i, encontre um conjunto máximo de índices $I \subseteq [k] = \{1, \ldots, k\}$ e, para todo $i \in I$, um caminho P_i de s_i a t_i tal que $E(P_i) \cap E(P_j) = \emptyset$ para quaisquer $i, j \in I$ distintos.*

Karp [3] provou que o MAXCDA é um problema NP-difícil. Garey e Johnson [2] mostraram que o MAXCDA é NP-difícil mesmo se nos restringirmos a grafos planares.

Em 1996, Kleinberg[4] apresentou uma $O(1/\sqrt{m})$-aproximação polinomial para o MAXCDA (G, T), onde $m := |E|$. Em 2003, Chekuri e Khanna [1] apresentaram uma $O(\max(1/n^{2/3}, 1/\sqrt{m}))$-aproximação polinomial para o MAXCDA, onde $n := |V|$ e $m := |E|$.

Mostramos aqui o algoritmo de Kleinberg.

Algoritmo GULOSO-MAXCDA (G, T)

$I \leftarrow \emptyset$
$G(\emptyset) \leftarrow G$
enquanto existe um par $(s_i, t_i) \in [k] \setminus I$ não separado faça
    encontre um par (s_i, t_i) com $i \in [k] \setminus I$ tal que
        $\mathrm{dist}_{G(I)}(s_i, t_i) = \min\{\, \mathrm{dist}_{G(I)}(s_j, t_j) : j \in [k] \setminus I \,\}$
    seja P_i um caminho mínimo de s_i a t_i
    $G' \leftarrow G(I) - E(P_i)$
    $I \leftarrow I \cup \{i\}$
    $G(I) \leftarrow G'$
devolva I e os caminhos P_i para cada $i \in I$

No que segue, tome $m := |E|$. Definimos o **comprimento** de um caminho P como o número de arestas de P. Denotamos por $\|P\|$ o comprimento de P. Diremos que um caminho P de G é **curto** se $\|P\| \leq \sqrt{m}$; diremos que P é

longo, se não é curto.

Lema 1: *O número de caminhos longos em* $\text{opt}(G,T)$ *é menor do que* $\sqrt{m}$.

Demonstração: Seja J um conjunto de índices de uma solução ótima do MaxCDA. Para cada $j \in J$, a solução ótima tem um caminho P_j^* de s_j a t_j. Tome

$$\mathcal{P} := \{P_j^* : j \in J \text{ e } P_j^* \text{ é um caminho longo}\}. \tag{1}$$

Como os caminhos em $\mathcal{P}$ são longos e disjuntos nas arestas, então

$$|\mathcal{P}|\sqrt{m} < \sum_{P \in \mathcal{P}} \|P\| \leq m. \tag{2}$$

$\square$

Teorema 2: *O algoritmo* Guloso-MaxCDA (G,T) *é uma* $O(1/\sqrt{m})$*-aproximação polinomial para o problema* MaxCDA (G,T).

Demonstração: Seja $I \subseteq [k]$ o conjunto de índices devolvidos pelo algoritmo. Suponha sem perda de generalidade que $I = [|I|]$. Para cada $i \in I$, seja P_i o caminho de s_i a t_i devolvido pelo algoritmo.

Seja $J \subseteq [k]$ o conjunto de índices de uma solução ótima para o problema. Para cada $j \in J$, seja P_j^* o caminho de s_j a t_j na solução ótima.

É fácil ver que

$$\|P_i\| \leq \|P_{i+1}\|$$

para todo $1 \leq i \leq |I|$. Além disso, pelo fato de G ser conexo, é claro que $|I| \geq 1$.

Considere $j \in J \setminus I$. Dizemos que P_j^* é **bloqueado** por um caminho P_i com $i \in I$, se $E(P_j^*) \cap E(P_i) \neq \emptyset$. Seja P_j^* um caminho curto com $j \in J \setminus I$. Vamos mostrar que P_j^* é bloqueado por um caminho curto P_i com $i \in I$. Seja P_i com $i \in I$ um caminho de menor comprimento possível que bloqueia P_j^*. É claro que tal caminho existe, caso contrário, o algoritmo iria incluir na solução um caminho entre s_j e t_j, e j pertenceria a I. Vamos mostrar que P_i é curto. Se $\|P_j^*\| < \|P_i\|$, então, no momento em que P_i foi escolhido existia entre s_j e t_j um caminho de comprimento menor do que $\|P_i\|$; neste caso, tal caminho teria sido escolhido antes que P_i e, portanto, j pertenceria a I. Assim, $\|P_i\| \leq \|P_j^*\|$, o que implica que P_i é um caminho curto.

Defina

$$\begin{aligned} I_c &:= \{i \in I : P_i \text{ é curto}\} \quad \text{e} \quad I_\ell := \{i \in I : P_i \text{ é longo}\}; \\ J_c &:= \{j \in J : P_j^* \text{ é curto}\} \quad \text{e} \quad J_\ell := \{j \in J : P_j^* \text{ é longo}\}. \end{aligned}$$

Note que

$$\Big|\bigcup_{i\in I_c} E(P_i)\Big| \leq |I_c|\sqrt{m},$$

pois $\|P_i\| \leq \sqrt{m}$ para todo $i \in I_c$. Além disso, claramente, cada aresta de um caminho curto P_i pode bloquear no máximo um caminho P_j^* com $j \in J_c \setminus I$. Portanto, como cada caminho P_j^* com $j \in J_c \setminus I$ é bloqueado por pelo menos um caminho P_i com $i \in I_c$, temos que

$$|J_c \setminus I| \leq |I_c|\sqrt{m}.$$

Assim,

$$|J_c| = |(J_c \setminus I) \cup I| \leq |J_c \setminus I| + |I| \leq (\sqrt{m}+1)|I|. \tag{3}$$

Usando o lema 1, o fato de que $|I| \geq 1$ e o fato (3), temos que

$$\begin{aligned} |J| = |J_c| + |J_\ell| < |J_c| + \sqrt{m} &\leq |J_c| + \sqrt{m}|I| \\ &\leq (\sqrt{m}+1)|I| + \sqrt{m}|I| \leq (2\sqrt{m}+1)|I|. \end{aligned}$$

Portanto, o tamanho do conjunto de índices devolvido pelo algoritmo é pelo menos $(2\sqrt{m}+1)^{-1}$ o tamanho do conjunto de índices de uma solução ótima. Além disso, o algoritmo Guloso-MaxCDA é polinomial. Basta observar que temos no máximo $\binom{n}{2}$ iterações e em cada iteração temos que encontrar caminhos mínimos entre no máximo $\binom{n}{2}$ pares de vértices, o que pode ser feito facilmente em tempo $O(n^3)$. Concluímos que o algoritmo Guloso-MaxCDA é uma $O(1/\sqrt{m})$-aproximação polinomial para o MaxCDA. □

Referências

[1] C. Chekuri and S. Khanna. Edge disjoint paths revisited. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms (Baltimore, MD, 2003)*, pages 628–637, New York, 2003. ACM.

[2] M.R. Garey and D.S. Johnson. *Computers and intractability: A guide to the theory of NP-completeness, A Series of Books in the Mathematical Sciences*. W. H. Freeman and Co., San Francisco, Calif., 1979.

[3] R.M. Karp. Reducibility among combinatorial problems. In *Complexity of computer computations (Proc. Sympos., IBM Thomas J. Watson Res. Center, Yorktown Heights, N.Y., 1972)*, pages 85–103. Plenum, New York, 1972.

[4] J. Kleinberg. *Approximation Algorithms for Disjoint Path Problems*. PhD thesis, Massachusetts Institute of Technology, Boston, Massachusetts, USA, 1996. `http://www.cs.cornell.edu/home/kleinberg/pract.ps`.