A $2k$-Vertex Kernel for Maximum Internal Spanning Tree

Wenjun Li, Jianxin Wang, Jianer Chen, Yixin Cao. WADS 2015

MO829B - Tópicos em Teoria de Computação

Ulysses Rocha

6 de dezembro de 2016

k-Internal Spanning Tree

O Problema

Existe no grafo uma árvore geradora com pelo menos k vértices internos?

NP-Hard

Podemos decidir a existência de um caminho hamiltoniano em G.

Maximum Internal Spanning Tree

2003	$O(k^3)$ - Prieto e Sloper [17].
2005	$O(k^2)$ - Prieto e Sloper [18].
2013	$O(3k)$ - **Fomim et al.** [4].

A Linear Vertex Kernel For Maximum Internal Spanning Tree

Theorem

Seja L' um conjunto independente de G, tal que $|\mathbf{L}'| \geq \mathbf{2}|\mathbf{N_G}(\mathbf{L}')|$. Podemos encontrar em tempo polinomial conjuntos não vazios $S \subseteq N_G(L')$ e $L \subseteq L'$, tal que :

1. $N_G(L) = S$, *e*
2. *O grafo $(S \cup L, E(G) \cap (S \times L))$ possui uma árvore geradora onde todos os vértices de S e $|S| - 1$ vértices de L são internos.*

Seja G' obtido de G adicionando v_s adjacente a cada vértice em $N_G(S) \setminus L$, adicionando v_l adjacente a v_s, removendo os vértices de $S \cup L$, temos $opt(G') = opt(G) - 2|S| + 2$.

Tamanho do Kernel

Se $|V(G)| \geq 3k - 3$ e $I(T) < k$, então $|L(T)| \geq 2k - 2 \geq 2|I(T)| \geq 2|N_G(T)|$.

A Linear Vertex Kernel For Maximum Internal Spanning Tree

Theorem

Seja L' um conjunto independente de G, tal que $|\mathbf{L'}| \geq \mathbf{2}|\mathbf{N_G}(\mathbf{L'})|$. Podemos encontrar em tempo polinomial conjuntos não vazios $S \subseteq N_G(L')$ e $L \subseteq L'$, tal que :

1. $N_G(L) = S$, *e*
2. *O grafo $(S \cup L, E(G) \cap (S \times L))$ possui uma árvore geradora onde todos os vértices de S e $|S| - 1$ vértices de L são internos.*

Seja G' obtido de G adicionando v_s adjacente a cada vértice em $N_G(S) \setminus L$, adicionando v_l adjacente a v_s, removendo os vértices de $S \cup L$, temos $opt(G') = opt(G) - 2|S| + 2$.

Tamanho do Kernel

Se $|V(G)| \geq 3k - 3$ e $I(T) < k$, então $|L(T)| \geq 2k - 2 \geq 2|I(T)| \geq 2|N_G(T)|$.

É possível melhorar essa kernelização?

Pergunta

É possível encontrar L' onde $|L'| \geq 2|N_G(L')|$ quando $|V(G)| < 3k - 3$?

Foco do artigo

Encontrar uma estrutura redutível quando $2k < |V(G)| < 3k - 3$

Processo

1. Pré-processar a árvore T para de aumentar o número de vértices internos
2. Se essa árvore tiver mais folhas que vértices internos, então um subconjunto de suas folhas fazem parte de uma estrutura redutível.
3. Aplicaremos a regra de redução do artigo anterior exaustivamente até obter uma solução ou uma árvore com $2k$-vértices.

É possível melhorar essa kernelização?

Pergunta

É possível encontrar L' onde $|L'| \geq 2|N_G(L')|$ quando $|V(G)| < 3k - 3$?

Foco do artigo

Encontrar uma estrutura redutível quando $2k < |V(G)| < 3k - 3$

Processo

1. Pré-processar a árvore T para de aumentar o número de vértices internos
2. Se essa árvore tiver mais folhas que vértices internos, então um subconjunto de suas folhas fazem parte de uma estrutura redutível.
3. Aplicaremos a regra de redução do artigo anterior exaustivamente até obter uma solução ou uma árvore com $2k$-vértices.

Procedimento Guloso de Busca Local

Definições

1. $L(T)$ é o conjunto de Folhas em T, um vértice u é folha se $d_T(u) = 1$.
2. Um vértice v é *branchpoint* se $d_T(v) \geq 3$
3. $I(T)$ é o conjunto de nós internos, $I_3(T)$ é o conjunto de *branchpoints* em T.

Número de folhas em uma árvore T

$$|L(T)| - 2 = \sum_{v \in I(T)} (d_T(v) - 2) = \sum_{v \in I_3(T)} (d_T(v) - 2)$$

Ideia!

Minimizar o número e o grau de branchpoints em T.

Procedimento Guloso de Busca Local

Definições

1. $L(T)$ é o conjunto de Folhas em T, um vértice u é folha se $d_T(u) = 1$.
2. Um vértice v é *branchpoint* se $d_T(v) \geq 3$
3. $I(T)$ é o conjunto de nós internos, $I_3(T)$ é o conjunto de *branchpoints* em T.

Número de folhas em uma árvore T

$$|L(T)| - 2 = \sum_{v \in I(T)} (d_T(v) - 2) = \sum_{v \in I_3(T)} (d_T(v) - 2)$$

Ideia!

Minimizar o número e o grau de branchpoints em T.

Procedimento Guloso de Busca Local

Definições

1. $L(T)$ é o conjunto de Folhas em T, um vértice u é folha se $d_T(u) = 1$.
2. Um vértice v é *branchpoint* se $d_T(v) \geq 3$
3. $I(T)$ é o conjunto de nós internos, $I_3(T)$ é o conjunto de *branchpoints* em T.

Número de folhas em uma árvore T

$$|L(T)| - 2 = \sum_{v \in I(T)} (d_T(v) - 2) = \sum_{v \in I_3(T)} (d_T(v) - 2)$$

Ideia!

Minimizar o número e o grau de branchpoints em T.

Primeira Regra de Troca (RT1)

Definição

Arestas de G que não estão em T, como $E(G) \setminus E(T)$ são arestas da co-árvore de T.

Regra

1. Obtenha uma árvore geradora arbitraria de G.
2. Se existe uma aresta da co-árvore conectando duas folhas l_1 e l_2 de T, então busque uma aresta uv de $P_T(l_1, l_2)$ tal que o vértice u seja um branchpoint, e substitua $l_1 l_2$ por uv em T.

Importante!

Após aplicar $RT1$ em T, as folhas de T não são adjacentes em G.

Primeira Regra de Troca (RT1)

Definição

Arestas de G que não estão em T, como $E(G) \setminus E(T)$ são arestas da co-árvore de T.

Regra

1. Obtenha uma árvore geradora arbitraria de G.
2. Se existe uma aresta da co-árvore conectando duas folhas l_1 e l_2 de T, então busque uma aresta uv de $P_T(l_1, l_2)$ tal que o vértice u seja um branchpoint, e substitua $l_1 l_2$ por uv em T.

Importante!

Após aplicar $RT1$ em T, as folhas de T não são adjacentes em G.

Segunda Regra de Troca (RT2)

Definições

1. Uma **aresta da co-árvore de T é boa** se conecta uma folha l a um vértice interno w de T.
2. Nós dizemos que lw atravessa todas as arestas no caminho $P_T(l, w)$

Segunda Regra de Troca (RT2)

Definições

3. Todos os vértices em $I_3(T)$ são separáveis.
4. Um vértice $w \in I_2(T)$ é separável se existe na co-árvore uma aresta boa lw de T que satisfaça um dos seguintes:
    4.1 *$P_T(l, w)$ passa por um branchpoint*
    4.2 *Algum vértice v em $P_T(l, w)$ é incidente na co-árvore a uma aresta boa $l'v$ de T onde $l' \neq l$*
5. $D(T)$ é o conjunto de todos os vértices separáveis em T, sendo $D_2(T)$ os vértices separáveis em $I_2(T)$. $D(T) = D_2(T) \cup I_3(T)$

Segunda Regra de Troca (RT2)

Regra

Seja uv uma aresta atravessada por uma aresta boa lw da co-árvore de T. Substitua lw por uv se satisfizer alguma das seguintes restrições:

- $u \in I_3(T)$, e
  - $v = w$ ou $v \in I_3(T)$; ou
  - $v \in D_2(T)$, e existe uma aresta boa $l_v v$ na co-árvore, onde $l_v \neq l$.
- $u \in D_2(T)$, existe uma aresta boa $l_u u$ na co-árvore onde $l_u \neq l$.
  - $v = w$ ou $v \in I_3(T)$; ou
  - $v \in D_2(T)$ e existe uma aresta boa $l_v v$ na co-árvore onde $l_v \notin \{l, l_u\}$

Árvore Geradora Maximal

Definição

Uma árvore geradora T é maximal quando não for mais possível aplicar $RT1$ ou $RT2$

Lema 2.1

Aplicando RT1 ou RT2 a uma árvore geradora T de G resulta em uma nova árvore geradora T' onde $|I(T')| \geq |I(T)| + 1$

Teorema 2.2

A árvore geradora maximal pode ser construida em tempo polinomial

Algoritmo de Kernelização

Lema 3.1

Seja L' um conjunto independente de G, tal que $|\mathbf{L}'| \geq \mathbf{2}|\mathbf{N_G}(\mathbf{L}')|$. Podemos encontrar em tempo polinomial conjuntos não vazios $S \subseteq N_G(L')$ e $L \subseteq L'$, tal que :

1. $N_G(L) = S$, *e*
2. *O grafo $(S \cup L, E(G) \cap (S \times L)))$ possui uma árvore geradora onde todos os vértices de S e $|S| - 1$ vértices de L são internos.*

Seja G' obtido de G adicionando v_s adjacente a cada vértice em $N_G(S) \setminus L$, adicionando v_l adjacente a v_s, removendo os vértices de $S \cup L$, temos $opt(G') = opt(G) - 2|S| + 2$.

Regra de Redução

Encontre um subconjunto não vazio S e L de vértices como no lema 3.1. Retorne (G', k') onde G' é definido no lema 3.1, e $k' = k - 2|S| + 2$

Lema 3.5

Seja T uma árvore geradora maximal onde $|V(G)| \geq 4$. Se $|L(T)| > |I(T)|$, podemos achar em tempo polinomial um conjunto independente L' de G onde $|L'| \geq 2|N_G(L')|$.

Teorema 1.2

Dado um grafo G com n-vértices, podemos encontrar em tempo polinomial ou uma árvore geradora de G com pelo menos $n/2$-vértices internos ou uma estrutura redutível.

Teorema 1.1

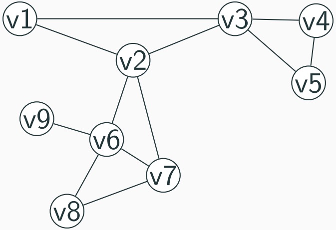
O problema da k-internal spanning tree tem um kernel de 2k-vértices

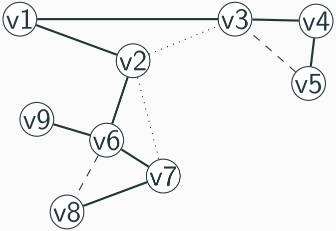
Teorema 1.3

O problema k-internal spanning tree pode ser resolvido[1] *em tempo* $O(4^k n^{O(1)})$

[1] Algoritmo de Nederlof para MIST tem complexidade $O(2^n n^{O(1)})$

Exercício





Exercício

Questão

Mostre um grafo onde seja possível aplicar as Regras de Troca 1 e 2.
Faça as trocas necessárias para obter uma árvore geradora maximal justificando as operações realizadas.