VOZES DIVERSAS DIFERENTES SABERES
Salão UFRGS 2018
SALÃO DE INICIACAO CIENTIFICA
XXX SIC
15 A 19 OUTUBRO CAMPUS DO VALE
UFRGS PROPESQ

Localização de Autovalores em Grafos com Pequeno Clique-Width

Rafael Jacobs Kehl
Orientador: Carlos Hoppen

Universidade Federal do Rio Grande do Sul

Instituto de Matemática e Estatística

Sumário

1 Definições
- Grafo
- Slick k-expressão
- Slick Clique-Width
- Matriz de Adjacência

2 Algoritmo
- Motivação
- Algoritmo

3 Referências

Definições

Grafo

Um grafo é formado pelos conjuntos (V, A) em que V é um conjunto arbitrário e A são pares não ordenados de V. Os elementos de V são chamados *vértices* e os de A de *arestas*.

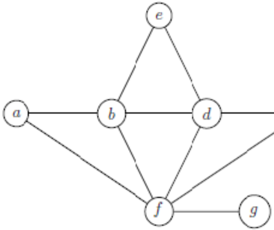


Figura: Exemplo de um grafo.

Localização de Autovalores
Rafael Kehl
Definições
Grafo
Slick k-expressão
Slick Clique-Width
Matriz de Adjacência
Algoritmo
Motivação
Algoritmo
Referências

Slick k-expressão

Uma *Slick k-expressão, ou simplesmente k-expressão, é formada por átomos* $i(v)$ *e uma operação binária* $\oplus_{S,L,R}$*, onde* L*,* R *são funções de* $[k]$ *em* $[k]$ *e* S *é uma relação binária em* $[k]$*, tal que:*

Definições

Slick k-expressão

- $i(v)$ cria o vértice v com rótulo i, com $i \in [k]$.

$$1(a) \rightarrow a_1$$

$$2(b) \rightarrow b_2$$

Definições

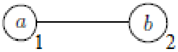
Slick k-expressão

- $i(v)$ cria o vértice v com rótulo i, com $i \in [k]$.
- Dados G e H grafos com vértices rotulados em $[k]$, obtemos $G \oplus_{S,L,R} H$ executando as seguintes operações:
  - Comece com a união disjunta de G e H

$$1(a) \oplus_{\{(1,2)\},\{1 \to 2\},id} 2(b) \quad \to$$

Slick k-expressão

- $i(v)$ cria o vértice v com rótulo i, com $i \in [k]$.
- Dados G e H grafos com vértices rotulados em $[k]$, obtemos $G \oplus_{S,L,R} H$ executando as seguintes operações:
  - Comece com a união disjunta de G e H
  - Adicione arestas entre todo vértice rotulado com $i \in G$ para todo vértice rotulado com $j \in H$ para todo $(i,j) \in S$

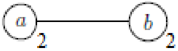
$1(a) \oplus_{\{(1,2)\},\{1\to 2\},id} 2(b) \quad \to$ 

Definições

Slick k-expressão

- $i(v)$ cria o vértice v com rótulo i, com $i \in [k]$.
- Dados G e H grafos com vértices rotulados em $[k]$, obtemos $G \oplus_{S,L,R} H$ executando as seguintes operações:
  - Comece com a união disjunta de G e H
  - Adicione arestas entre todo vértice rotulado com $i \in G$ para todo vértice rotulado com $j \in H$ para todo $(i, j) \in S$
  - Substitua todo rótulo i da componente G por $L(i)$, e todo rótulo i da componente H por $R(i)$.

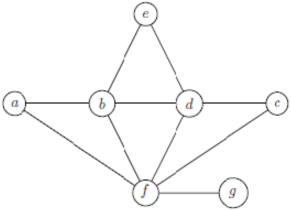
$1(a) \oplus_{\{(1,2)\},\{1 \to 2\},id} 2(b) \quad \rightarrow$



Definições

Slick Clique-Width

O slick clique-width de um grafo G, denotado por $scw(G)$, é o menor k com o qual podemos definir G



$$[1(f) \oplus_{\{(1,2)\},id,id} 2(g)] \oplus_{\{(1,2)\},\{2\to1\},\{2\to1\}}$$
$$[\{1(e)\} \oplus_{\{(1,2)\},id,\{1\to2\}}$$
$$\{((1(c) \oplus_{\{(1,2)\},id,id} 2(d))) \oplus_{\{(2,2)\},id,id} ((1(a) \oplus_{\{(1,2)\},id,id} 2(b)))\}]$$

Figura: *Grafo e sua Slick k-expressão. Note que $k = scw(G) = 2$.*

Localização de Autovalores
Rafael Kehl
Definições
Grafo
Slick k-expressão
Slick Clique-Width
Matriz de Adjacência
Algoritmo
Motivação
Algoritmo
Referências

Slick Clique-Width

Outra maneira de representar uma *Slick k-expressão é através de sua árvore sintática.*

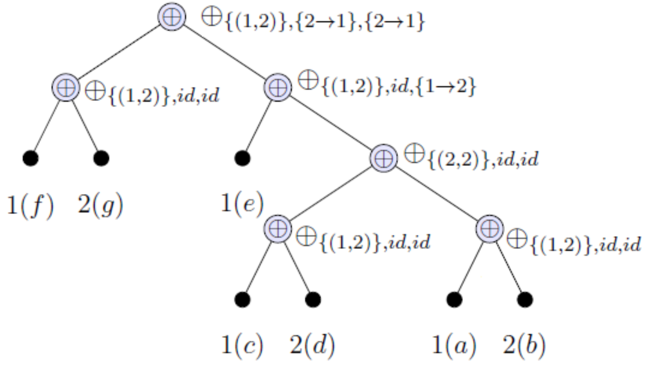


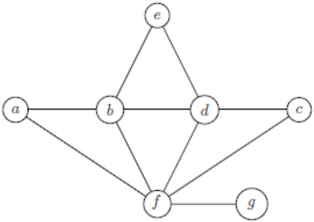
Figura: *A árvore sintática da Slick k-expressão anterior.*

Definições

Matriz de Adjacência

A matriz de adjacência é uma das formas de representar um grafo. As entradas da matriz são tais que

$$a_{ij} = \begin{cases} 1, & \text{se } i \text{ é adjacente a } j \\ 0, & \text{se } i \text{ não é adjacente a } j \end{cases}$$



$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Figura: Grafo e sua matriz de adjacência.

Motivação

A Teoria Espectral dos Grafos estuda a relação existente entre o espectro de um grafo e suas propriedades estruturais. Portanto, é de grande interesse ser capaz de dizer rapidamente quantos autovalores existem em um dado intervalo.

Neste contexto, M. Fürer, C. Hoppen, D. P. Jacobs e V. Trevisan desenvolveram um algoritmo que, dada uma *slick k-expressão* que gera um grafo G, encontra uma matriz diagonal D congruente a $B_c = A - cI$, para c um número real qualquer, onde A é a matriz de adjacências de G.

Motivação

A diagonalização é feita em um tempo da ordem de $O(poly(k)n)$ onde n é o número de vértices de G. Dessa forma, podemos rapidamente dizer quantos autovalores existem em um dado intervalo fazendo o uso da *Lei da Inércia de Sylvester*.

Lei da Inércia de Sylvester

Duas matrizes reais simétricas de ordem $n \times n$ são congruentes se e somente se elas têm o mesmo número de autovalores negativos e o mesmo número de autovalores positivos.

Algoritmo

O algoritmo atua na árvore sintática T da *slick k-expressão das folhas para raiz. Em particular, uma das maneiras de se fazer isso é com um caminhamento pós-fixado à esquerda, que foi o implementado no trabalho.*

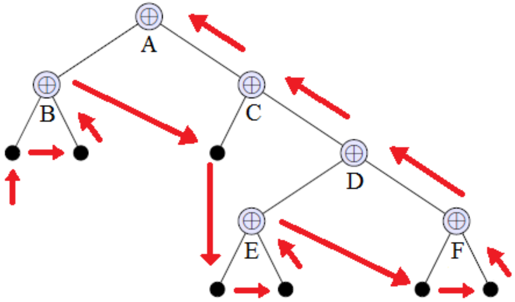


Figura: *Caminhamento pós-fixado à esquerda na arvore do exemplo.*

Algoritmo

Um nodo $Q \in T$ produz uma estrutura chamada *k-box*, que é uma 4-upla da forma $b_Q = [k', k'', M, \Lambda]$. A desigualdade $0 \leqslant k' \leqslant k'' \leqslant k$ sempre deve ser mantida e diremos que k' e k'' são, respectivamente, o número de vértices de *tipo-i (pré-diagonalizado) e tipo-ii (não processado) no subgrafo gerado pela k-expressão até o nodo Q.*

$$b_{i(v)} = [0, 1, (0), (i)]$$

$$b_F = [0, 2, \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right), \left(\begin{smallmatrix} 1 \\ 2 \end{smallmatrix}\right)]$$

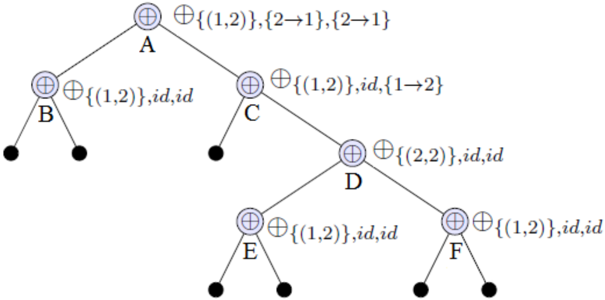
Algoritmo

No caso do nosso exemplo, tomando $c = 0$ e seguindo o caminhamento dado, teremos o seguinte ao processar as folhas:

$$b_Q = \begin{cases} [0, 1, (0), (1)], & \text{se a folha for } 1(v) \\ [0, 1, (0), (2)], & \text{se a folha for } 2(v) \end{cases}$$

Algoritmo

Note que os vértices B, E e F possuem a mesma operação, $1(v) \oplus_{\{(1,2)\},id,id} 2(u)$, portanto as *box* geradas por eles também serão idênticas.



Algoritmo

Seja $Q = Q_l \oplus_{S,L,R} Q_r$ o nodo que estamos processando. Para combinar as *box*, começamos com uma união disjunta das matrizes M e vetores Λ dos filhos.

Em seguida, para todo $(i,j) \in S$ fazemos $m_{uv} = m_{vu} = 1$, se $u \in Q_l$ é de *tipo-ii com rótulo i e se $v \in Q_r$ é de tipo-ii com rótulo j e troque os rótulos de Q_l e Q_r de acordo com as funções L e R, respectivamente.*

Para terminar, faça $k' = k'_l + k'_r$ e $k'' = k''_l + k''_r$. Dessa forma, teremos:

$$1(v) \oplus_{\{(1,2)\},id,id} 2(u) \to b_B = b_E = b_F = [0, 2, \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right), \left(\begin{smallmatrix} 1 \\ 2 \end{smallmatrix}\right)]$$

Algoritmo

Fazendo o mesmo para o nodo D obteremos a seguinte *box*:

$$b_D = [0, 4, \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 2 \\ 1 \\ 2 \end{pmatrix}]$$

Observe que a desigualdade $k' = 0 \leqslant k'' = 4 \leqslant k = 2$ não é mais verdadeira. Portanto, precisamos transformar algumas linhas de *tipo-ii em tipo-i. Para tal, enunciaremos agora o Lema 1.*

Algoritmo

Lema 1

Se M_Q possui duas linhas *tipo-ii com o mesmo rótulo, podemos produzir uma matriz congruente onde k'' diminui em um e k' aumenta em um.*

Para tal, sejam dois vértices de *tipo-ii com mesmo rótulo,* j *e* j'*. Note que, se dois vértices vértices possuem o mesmo rótulo em um nodo* Q *eles possuírão as mesmas adjacências em qualquer grafo gerado em um nodo acima de* Q*. Portanto, as operações*

$$R_{j'} \leftarrow R_{j'} - R_j$$
$$C_{j'} \leftarrow C_{j'} - C_j$$

irão zerar todas as entradas de $R_{j'}$ *e* $C_{j'}$ *fora da matriz* M_Q*. Então* j' *se tornou uma linha de tipo-i, como queríamos.*

Algoritmo

Aplicando o *Lema 1 para os pares de vértices* $(1,3)$ *e* $(2,4)$ *e agrupando as linhas de tipo-i e tipo-ii na parte superior e inferior da matriz, respectivamente, obtemos a box:*

$$b_D = [2, 2, \left(\begin{array}{cc:cc} 0 & 2 & 0 & -1 \\ 2 & -2 & -1 & 1 \\ \hdashline 0 & -1 & 0 & 1 \\ -1 & 1 & 1 & 0 \end{array}\right), \begin{pmatrix} 1 \\ 2 \\ 1 \\ 2 \end{pmatrix}]$$

Para facilitar, agora subdividiremos M *da seguinte forma:*

$$M = \left(\begin{array}{c:c} M^{(0)} & M^{(1)} \\ \hdashline M^{(1)T} & M^{(2)} \end{array}\right)$$

onde as linhas tipo-i estão em $M^{(0)}$ *e as tipo-ii estão em* $M^{(2)}$*.*

Algoritmo

Avançando para o nodo C, adicionamos um vértice ao grafo gerado por D, aplicando S, L e R, temos inicialmente a seguinte *box*:

$$b_C = [2, 3, \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 2 & 0 & -1 \\ 0 & 2 & -2 & -1 & 1 \\ 0 & 0 & -1 & 0 & 1 \\ 1 & -1 & 1 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 2 \\ 2 \\ 2 \\ 2 \end{pmatrix}]$$

Observe que, novamente, a desigualdade $k' = 2 \leqslant k'' = 3 \leqslant k = 2$ não é mais verdadeira. Portanto, precisamos transformar algumas linhas de *tipo-ii em tipo-i. Temos então que aplicar o Lema 1 e agrupar as linhas.*

Algoritmo

Com isso, ficamos com a seguinte *box*:

$$b_C = [3, 2, \left(\begin{array}{ccc:cc} -2 & 2 & 2 & 0 & 1 \\ 2 & 0 & 1 & 0 & -1 \\ -2 & 1 & 2 & -1 & 1 \\ \hdashline 0 & 0 & -1 & 0 & 1 \\ 1 & -1 & 1 & 1 & 0 \end{array}\right), \begin{pmatrix} 2 \\ 2 \\ 2 \\ 1 \\ 2 \end{pmatrix}]$$

Porém, temos $k' = 3 > 2 = k''$, violando nossa desigualdade. Portanto, precisamos fazer com que k' diminua. Para isso, enunciaremos o *Lema 2.*

Algoritmo

Lema 2

Dada uma matriz M_Q, podemos tornar $M^{(0)}$ nula ou vazia.

Sejam i e j linhas de $M^{(0)}$ com $m_{ii} \neq 0$, basta aplicar as operações

$$R_j \leftarrow R_j - \frac{m_{ij}}{m_{ii}} R_i$$
$$C_j \leftarrow C_j - \frac{m_{ij}}{m_{ii}} C_i$$

Assim, a linha e coluna será diagonalizada e poderá ser retirada de $M^{(0)}$, diminuindo k' em um.

Algoritmo

Caso $m_{ii} = 0$, porém existe um $m_{ij} \neq 0$, executamos as seguintes operações

$$R_j \leftarrow R_j - \tfrac{1}{2} R_i$$
$$C_j \leftarrow C_j - \tfrac{1}{2} C_i$$

Seguidas por

$$R_i \leftarrow R_i - R_j$$
$$C_i \leftarrow C_i - C_j$$

O que transforma a matriz da seguinte forma:

$$\begin{pmatrix} 0 & m_{ij} \\ m_{ij} & 0 \end{pmatrix} \rightarrow \begin{pmatrix} 0 & m_{ij} \\ m_{ij} & m_{ij} \end{pmatrix} \rightarrow \begin{pmatrix} -m_{ij} & 0 \\ 0 & m_{ij} \end{pmatrix}$$

Tornando possível a aplicação das operações para $m_{ii} \neq 0$.

Algoritmo

Para o caso em que $M^{(0)}$ é uma matriz nula temos o *Lema 3.*

Lema 3

Considere uma matriz M_Q após a aplicação do Lema 2, então podemos tornar $k' \leqslant k''$.

Tomando $M^{(0)} = 0$ com $k' > k''$, com operações simples entre as linhas, podemos tornar $M^{(1)}$ triangular superior. Aplicando as mesmas operações nas colunas de $M^{(1)T}$ diagonalizaremos pelo menos $k' - k''$ linhas. Teremos assim $k' \leqslant k''$.

Algoritmo

Aplicando as operações do *Lema 2 em* b_C*, conseguimos extrair os elementos* $-2, 2$ *e* $-\frac{1}{2}$*, fazendo* $M^{(0)}$ *vazia. Assim, a box fica da seguinte forma:*

$$b_C = [0, 2, \begin{pmatrix} 2 & 1 \\ 1 & \frac{1}{2} \end{pmatrix}, \begin{pmatrix} 1 \\ 2 \end{pmatrix}]$$

Finalmente, b_C *está pronta para ser transmitida e só nos resta processar o nodo A.*

Algoritmo

Combinando b_B e b_C, teremos a seguinte *box*:

$$b_A = [0, 4, \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 1 \\ 1 & 0 & 1 & \frac{1}{2} \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}]$$

Como $k'' > k$, precisamos aplicar as operações do *Lema 1, o que nos dá:*

$$b_A = [3, 1, \frac{1}{2} \cdot \left(\begin{array}{ccc|c} -3 & 1 & -3 & 1 \\ 1 & 1 & -1 & -1 \\ -3 & -1 & 1 & 1 \\ \hline 1 & -1 & 1 & 1 \end{array} \right), \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}]$$

Algoritmo

Finalmente, vamos aplicar as operações do *Lema 2 em* b_A *e extrair os elementos* $-\frac{3}{2}, \frac{2}{3}, \frac{1}{2}$ *e* 0, *fazendo* M *vazia e terminando o algoritmo, pois temos os elementos diagonais*

$$D = (-2, -\frac{3}{2}, -\frac{1}{2}, 0, \frac{1}{2}, \frac{2}{3}, 2)$$

Dessa forma, concluímos que o grafo possui 3 *autovalores positivos,* 3 *autovalores negativos e um autovalor nulo. De fato, o espectro do grafo pode ser aproximado por*

$$[-1.9098; -1.6180; -1.2726; 0; 0.6180; 0.8692; 3.3132]$$

Agradecimentos

Professores: Carlos Hoppen, Vilmar Trevisan, Luiz Emílio Allem e Daniel Monçalves.

Referências

M. Fürer, C. Hoppen, D. P. Jacobs, V. Trevisan
Eigenvalue location in graphs of small clique-width.
Linear Algebra and its Applications, Volume 560, 1 January 2019, Pages 56-85, https://doi.org/10.1016/j.laa.2018.09.015

Reinhardt Diestel
Graph Theory (2nd edition)
Springer-Verlag, 1:2–26, 12:251–277, 2000.

N. Abreu, R. Del-Vecchio, V. Trevisan, C. Vinagre.
Teoria Espectral de Grafos – Uma Introdução.
III° Colóquio de Matemática da Região Sul., 6:145–172, 2014.

Definições
Grafo
Slick k-expressão
Slick Clique-Width
Matriz de Adjacência
Algoritmo
Motivação
Algoritmo
Referências

Muito obrigado!