"On structural parameterizations for the 2-club problem", Hartung et al.

Jorge Augusto Hongo

01/12/16

Roteiro

Definição

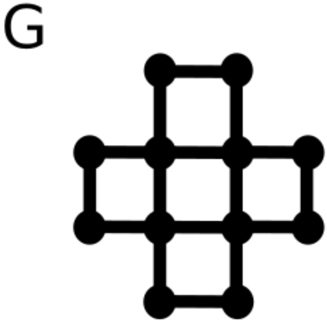
s-CLUB

Entrada: um grafo não direcionado $G = (V, E)$ e $l \in \mathbb{N}$.
Saída: Existe um conjunto de vértices $S \subseteq V$ de tamanho ao menos l tal que o subgrafo induzido $G[S]$ tem diâmetro no máximo s?

Definição

2-CLUB

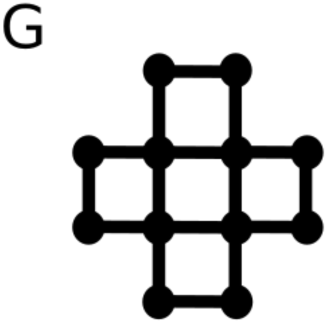
Entrada: um grafo não direcionado $G = (V, E)$ e $l \in \mathbb{N}$.
Saída: Existe um conjunto de vértices $S \subseteq V$ de tamanho ao menos l tal que o subgrafo induzido $G[S]$ tem diâmetro no máximo 2?

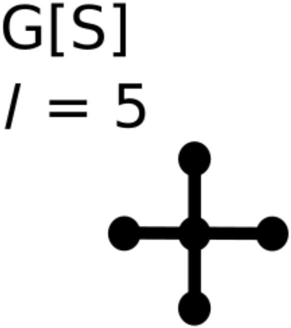


Definição

2-CLUB

Entrada: um grafo não direcionado $G = (V, E)$ e $l \in \mathbb{N}$.
Saída: Existe um conjunto de vértices $S \subseteq V$ de tamanho ao menos l tal que o subgrafo induzido $G[S]$ tem diâmetro no máximo 2?

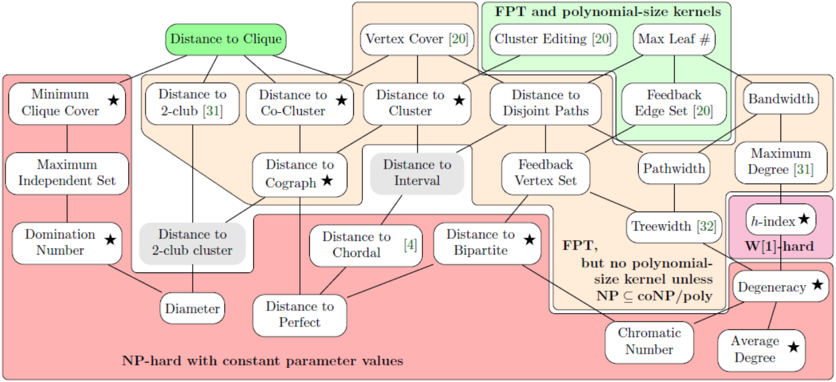




Motivação

- Modelo de relaxação de cliques.
  - CLIQUE equivale a 1-CLUB.
- Aplicação no estudo de redes sociais e biológicas.
  - CLIQUE frequentemente é uma formulação excessivamente restritiva para a propriedade que estas aplicações desejam identificar.
- NP-difícil para grafos arbitrários de diâmetro $s+1$.
  - Esforços atuais focados em entender a natureza do s-CLUB para uma variedade de parâmetros.

Histórico



- Focaremos especificamente a prova para "Distância a grafos *cluster*".

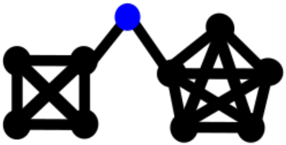
Conceitos

Grafo *cluster*

Um grafo G é um grafo *cluster* se toda componente conexa de G for uma clique.

Distância a grafo *cluster*

Número mínimo de vértices a remover de um grafo G para torná-lo um grafo *cluster*. Remoção de arestas não é considerada neste artigo.



Conceitos

Vértices gêmeos

Dois vértices v e w são *gêmeos em relação a um conjunto de vértices* X se $N(v) \cap X = N(w) \cap X$.

Classe de gêmeos

Um conjunto de vértices $T \subseteq V(G)$ é chamado de *classe de gêmeos* se, para todo par de vértices distintos $t_i, t_j \in T$, t_i e t_j forem gêmeos.

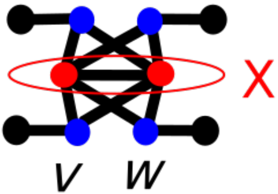
Conceitos

Vértices gêmeos em relação a X

Dois vértices v e w são *gêmeos em relação a um conjunto de vértices* X se $N(v) \cap X = N(w) \cap X$.

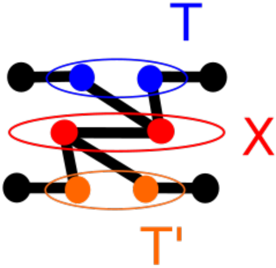
Classe de gêmeos em relação a X

Um conjunto de vértices $T \subseteq V(G)$ é chamado de *classe de gêmeos em relação a um conjunto de vértices* X se, para todo par de vértices distintos $t_i, t_j \in T$, t_i e t_j forem gêmeos em relação a X.



Classe de gêmeos em conflito em relação a X

Neste artigo, duas classes de gêmeos T e T' em relação a um conjunto X são consideradas *em conflito* se $N(T) \cap N(T') \cap X = \emptyset$.



Algoritmo

Teorema

2-CLUB é resolvido em tempo $\mathcal{O}(2^k * 3^{2^k} * nm)$, no qual k denota distância a grafo *cluster*.

Algoritmo

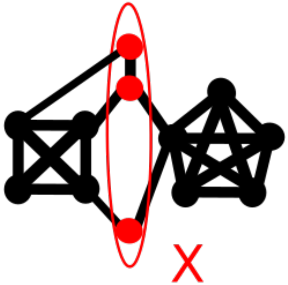
Teorema

2-CLUB é resolvido em tempo $\mathcal{O}(2^k * 3^{2^k} * nm)$, no qual k denota distância a grafo *cluster*.

- Ramificação (*branching*) de 2^k possibilidades baseadas na distância a grafo *cluster*.
- Cada ramificação segue com uma programação dinâmica em tempo $\mathcal{O}(3^{2^k} * nm)$.

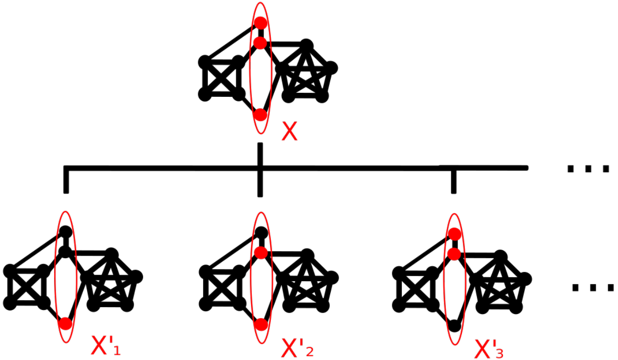
Algoritmo

- **Entrada:** instância (G, X, l)
  - X é um conjunto de vértices $X \subseteq G$ cuja remoção $G - X$ resulta em um grafo *cluster*.
  - $k = |X|$, ou seja, distância de G a um grafo *cluster*.
  - $l =$ número de vértices mínimo da solução do problema 2-CLUB.



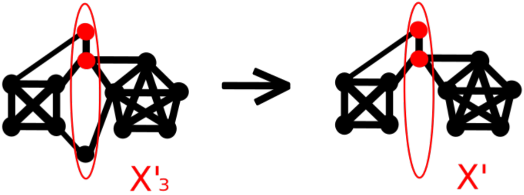
Algoritmo

- **Passo 1:** Ramifique em todos os possíveis subconjuntos $X' \subseteq X$ que possam estar na solução final de 2-CLUB.
  - Supondo que X' seja parte da solução, queremos encontrar os demais vértices $Y \subseteq V' \setminus X'$ de uma solução que contenha X'.



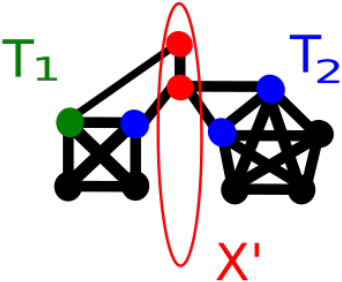
Algoritmo

- **Passo 2:** Em cada ramificação X', construa um grafo $G' = (V', E')$ por meio da remoção de $X \setminus X'$ (vértices de X que a ramificação supõe não estar na solução final) e dos vértices que não estejam dentro de uma distância 2 de algum vértice de X'.
  - Remoção de vértices que trivialmente não podem ser parte da solução dentro da ramificação proposta em X'. Note que $G' - X'$ resulta em um grafo *cluster*, visto que $G' - X' \subseteq G - X$.



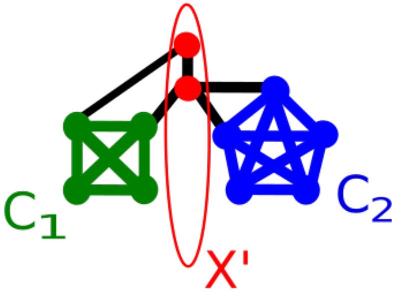
Algoritmo

- **Passo 3:** Obtenha o conjunto $\mathcal{T} = T_1, ..., T_p$ de classes de gêmeos de $V' \setminus X'$ com respeito a X'.
  - Um passo futuro percorrerá todas os possíveis subconjuntos $\mathcal{T}' \subseteq \mathcal{T}$ de classes de gêmeos de $V' \setminus X'$ (ou $G' - X'$).



Algoritmo

- **Passo 4:** Mantenha registro dos clusters $C_1, ..., C_q$ de $G' - X'$.
  - Um passo futuro verificará a possível solução final adicionando um *cluster* por vez à análise. Esta adição incremental de *clusters* será representada por $\bigcup_{1 \leq j \leq i} C_j$, sendo i o número do último cluster adicionado à análise.

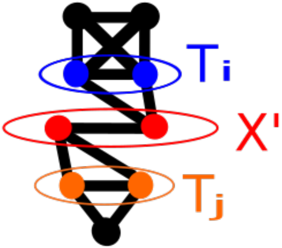


Algoritmo

- **Passo 5:** Crie uma tabela $\mathcal{A}$, no qual $\mathcal{A}[i, \mathcal{T}']$ armazena o tamanho do maior conjunto $Y \subseteq \bigcup_{1 \leq j \leq i} C_j$ tal que as classes de gêmeos em Y seja exatamente $\mathcal{T}' \subseteq \mathcal{T}$ e todo vértice de Y tenha distância até 2 dos vértices de $Y \cup X'$ no grafo induzido $G[Y \cup X']$.
  - $\mathcal{A}[i, \mathcal{T}']$ simplesmente diz quantos vértices dentro da união de clusters $\bigcup C_1, .., C_i$ são parte de uma solução contendo X' se permirtirmos apenas certas classes de gêmeos $(\mathcal{T}')$ serem parte da solução (Y). O raciocínio para realizar a análise desta forma vem de três observações:

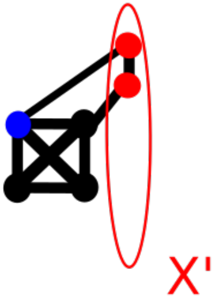
Algoritmo

- **Observação 1:** Se duas classes de gêmeos T_i e T_j em relação a X' estão em conflito, então todos os vértices de T_i e T_j que estejam em um 2-club devem estar no mesmo *cluster* de $G' - X'$.



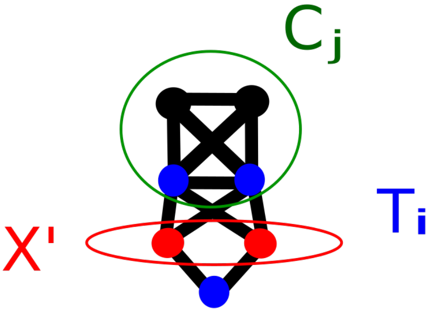
Algoritmo

- **Observação 2:** Todo vértice de $G' - X'$ só alcança todos os vértices de X' ou por meio de X', ou por meio dos vértices do seu *cluster*.



Algoritmo

- **Observação 3:** Se um vértice v de um 2-club S estiver em uma classe de gêmeos $v \in T_i$ e, simultaneamente, em um cluster $v \in C_j$, então todos os vértices que também estão simultaneamente em T_i e C_j podem ser adicionados ao 2-club solução S, o qual continuará sendo um 2-club.



Algoritmo

- **Passo 6:** Use programação dinâmica para preencher a tabela $\mathcal{A}$ por meio da recorrência:

$$\mathcal{A}[i,\mathcal{T}'] = max_{\mathcal{T}''\subseteq\mathcal{T}',\tilde{\mathcal{T}}\subseteq\mathcal{T}'} \begin{cases} \mathcal{A}[i-1,\tilde{\mathcal{T}}]+s(i,\mathcal{T}'') \text{ se } \tilde{\mathcal{T}}\cup\mathcal{T}''=\mathcal{T}'\wedge\neg conf(\tilde{\mathcal{T}},\mathcal{T}'') \\ -\infty \text{ caso contrário.} \end{cases}$$

, no qual:

  - $\tilde{\mathcal{T}} \cup \mathcal{T}'' = \mathcal{T}'$
  - $s(i,\mathcal{T}'')$ é o maior número de vértices que podemos adicionar de C_i que provêm das classes de gêmeos em $\mathcal{T}''$ conforme as condições impostas pela tabela $\mathcal{A}$.
  - $conf(\mathcal{T}'',\tilde{\mathcal{T}})$ diz se há conflito entre classes de gêmeos (o que implica distância maior que 2 entre algum par de vértices).

Algoritmo

$$\mathcal{A}[i, \mathcal{T}'] = max_{\mathcal{T}'' \subseteq \mathcal{T}', \tilde{\mathcal{T}} \subseteq \mathcal{T}'} \begin{cases} \mathcal{A}[i-1, \tilde{\mathcal{T}}] + s(i, \mathcal{T}'') \text{ se } \tilde{\mathcal{T}} \cup \mathcal{T}'' = \mathcal{T}' \wedge \neg conf(\tilde{\mathcal{T}}, \mathcal{T}'') \\ -\infty \text{ caso contrário.} \end{cases}$$

- **Passo 6a:** Em cada passo da programação dinâmica, devemos computar $s(i, \mathcal{T}'')$. Para isso, buscamos o subconjunto $C_i^{\mathcal{T}''}$ maximal de vértices de C_i cujas classes de gêmeos seja exatamaente $\mathcal{T}''$.
  - Se o subconjunto $C_i^{\mathcal{T}''}$ existir, então $s(i, \mathcal{T}'') = |C_i^{\mathcal{T}''}|$.
  - Senão, $s(i, \mathcal{T}'') = -\infty$.
  - Caso especial, $s(i, \emptyset) = 0$.
- Desta forma, $s(i, \mathcal{T}'')$ informa quantos vértices podemos adicionar à solução com a inclusão das classes de gêmeos não consideradas em $\mathcal{A}[i-1, \tilde{\mathcal{T}}]$ que $\mathcal{A}[i, \mathcal{T}']$ pretende cobrir.

Algoritmo

$$\mathcal{A}[i,\mathcal{T}'] = max_{\mathcal{T}''\subseteq\mathcal{T}',\tilde{\mathcal{T}}\subseteq\mathcal{T}'} \begin{cases} \mathcal{A}[i-1,\tilde{\mathcal{T}}]+s(i,\mathcal{T}'') \text{ se } \tilde{\mathcal{T}}\cup\mathcal{T}''=\mathcal{T}'\wedge\neg conf(\tilde{\mathcal{T}},\mathcal{T}'') \\ -\infty \text{ caso contrário.} \end{cases}$$

- **Passo 6b:** Em cada passo da programação dinâmica, devemos também computar $conf(\mathcal{T}'',\tilde{\mathcal{T}})$, ou "conflito".
    - Verdadeiro se houver duas classes de gêmeos $T_i \in \mathcal{T}''$ e $T_j \in \tilde{\mathcal{T}}$ tal que T_i e T_j estão em conflito;
    - Falso caso contrário.
- Assim, a condição $\tilde{\mathcal{T}} \cup \mathcal{T}'' = \mathcal{T}' \wedge \neg conf(\tilde{\mathcal{T}},\mathcal{T}'')$ da recorrência avalia se é possível que haja distância maior que 2 entre os vértices de duas classes de gêmeos quaisquer de $\tilde{\mathcal{T}} \cup \mathcal{T}'' = \mathcal{T}'$.

Algoritmo

- **Passo 7:** Finalizada a programação dinâmica, verifique se os vértices X' estão a uma distância até 2 entre si para o grafo induzido $G'[Y \cup X']$ da solução Y das entradas da forma $\mathcal{A}[q, \mathcal{T}']$. O maior tamanho de um 2-club em G' é então o maior valor das entradas da tabela $\mathcal{A}$ que satisfaçam tal condição.
  - A solução Y de uma entrada $\mathcal{A}[q, \mathcal{T}']$ já garante que todo vértice de Y está a uma distância até 2 de todos os demais vértices em $G'[Y \cup X']$. Falta garantir que os vértices da ramificação X' também está a uma distância até 2 entre si, seja por um vizinho em comum em X' ou por uma classe de gêmeos em Y.

Algoritmo

- **Tempo de execução** Limitado a $\mathcal{O}(2^k * 3^{2^k} * nm)$:
  - 2^k partições de X cada uma preenchendo uma tabela com 2^{2^k} entradas.
  - Uma vez que há 3^{2^k} possibilidades de particionar $\mathcal{T}$ em $\mathcal{T}''$, $\tilde{\mathcal{T}}$ e $\mathcal{T} \setminus \mathcal{T}'$, o tempo total de computar uma tabela é $\mathcal{O}(3^{2^k} * n)$.
  - O cálculo de $s(i, \mathcal{T}'')$ é feito em tempo $\mathcal{O}(nm)$ por cada *cluster* C_i ser uma clique.

Exercício

No quinto passo do algoritmo descrito, há três observações que guiam a adição de vértices à solução tal que ela permanece uma 2-club. Escolha qualquer uma das observações e prove que ela é verdadeira.

Fim