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

Jorge Augusto Hongo
RA: 071294

01/12/16

1 Introdução

Este resumo trata do artigo ”On structural parameterizations for the 2-club problem”, no qual os autores demonstram se o problema 2-CLUB (definido na seção 2) é NP-difícil, W[1]-difícil, ou tratável por parâmetro fixo, para um amplo leque de possíveis parâmetros. Dentro do contexto da disciplina, focaremos a prova de um parâmetro para o qual 2-CLUB admite um algoritmo FPT: distância a grafos *cluster*.

2 Histórico

O problema 2-CLUB corresponde ao caso $s = 2$ do s-CLUB definido a seguir:

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?

Caso especial: Clique corresponde ao caso 1-CLUB.

O problema 2-CLUB é um modelo de relaxação de cliques, sua motivação sendo o desejo de retratar a coesão dentro de um grafo de forma relaxada em relação à definição de clique. As suas principais aplicações são dentro de redes sociais e biológicas, nos quais grupos fortemente coesos são candidatos potenciais de comunidades ou redes biológicas funcionais.

Pela literatura atual, s-CLUB é considerado:

- NP-completo em um grafo de diâmetro $s+1$, em grafos *split* e em grafos cordais.
- Polinomial em grafos bipartidos, em árvores e em grafos de intervalo.
- FPT quando parametrizado em l ou em largura de árvore, para $s \geq 2$.

Para $s = 2$ especificamente, incluindo o artigo em questão, tem-se que 2-CLUB é:

- NP-difícil para os seguintes parâmetros com valores constantes: Cobertura mínima de clique, Conjunto independente máximo, Número de dominação, diâmetro, Distância de um grafo perfeito, Distância de um grafo bipartido, Número cromático, Degeneração, Grau médio.
- FPT sem núcleo polinomial (exceto se $NP \subseteq coNP/poly$): Cobertura de vértices, Distância a 2-CLUB, Distância a caminhos disjuntos, Conjunto de vértices de retroalimentação, Largura de caminho, Largura de árvore, Grau máximo, Largura de banda, Distância a *cluster graph*, Distância a *co-cluster graph*, Distância a cografos.
- FPT com núcleo polinomial: Edição de cluster, Número máximo de folhas, Conjunto de arestas de retroalimentação.
- W[1]-difícil relativo ao índice h (um grafo possui índice h igual a k se k for o maior número tal que o grafo possua k vértices de grau ao menos k).
- NP-difícl para grafos que são bipartidos com a remoção de um vértice.
- Aberto se FPT quando parametrizado em: Distância a grafo de intervalo, Distância a *cluster* de 2-club.
- Aberto de admite núcleo polinomial quando parametrizado pela Distância a clique.

3 Conceitos utilizados

Um grafo G é um grafo *cluster* se toda componente conexa de G for uma clique. Equivalentemente, um grafo G é um grafo *cluster* se e somente se não houver um caminho induzido de largura 3 (P_3*-free graph*). Cada clique é comumente chamada de um *cluster* de G.

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 para este parâmetro neste artigo.

Dois vértices v e w são gêmeos (*twins*) se $N(v) \setminus \{w\} = N(w) \setminus \{v\}$.

- 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.

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

- 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.

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

4 RESULTADO E ALGORITMO

O resultado obtido por Hartung et al. sobre 2-CLUB parametrizado pela distância a grafos *cluster* é descrito pelo seguinte teorema:

Teorema 4.1. *2-CLUB é resolvido em tempo $\mathcal{O}(2^k * 3^{2^k} * nm)$, no qual k denota distância a grafo cluster.*

O algoritmo usado para provar este resultado recebe como entrada uma instância (G, X, l), no qual X é um conjunto de vértices $X \subseteq G$ cuja remoção $G - X$ resulta em um grafo *cluster*, com tamanho $|X| = k$, sendo k a distância de G a um grafo *cluster*. Como lembrete, l é o número de vértices mínimo da solução do problema 2-CLUB.

Os passos do algoritmo, e a intuição desses passos, podem ser resumidos como:

- 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'.
- 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 $G' - X' \subseteq G - X$.
- 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 $G' - X'$).
- 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 $\bigcup_{1\le j\le i} C_j$, sendo i o número do último cluster adicionado à análise.

- Crie uma tabela $\mathscr{A}$, no qual $\mathscr{A}[i,\mathscr{T}']$ armazena o tamanho do maior conjunto $Y \subseteq \bigcup_{1\le j\le i} C_j$ tal que as classes de gêmeos em Y seja exatamente $\mathscr{T}' \subseteq \mathscr{T}$ e todo vértice de Y tenha distância até 2 dos vértices de $Y \cup X'$ no grafo induzido $G[Y \cup X']$.
  - $\mathscr{A}[i,\mathscr{T}']$ 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 ($\mathscr{T}'$) serem parte da solução (Y). O raciocínio para realizar a análise desta forma vem de três observações:
    - **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 $G' - X'$.
    - **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*.
    - **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.

- Use programação dinâmica para preencher a tabela $\mathscr{A}$ por meio da recorrência:

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

  , no qual $s(i,\mathscr{T}'')$ é o maior número de vértices que podemos adicionar de C_i que provêm das classes de gêmeos em $\mathscr{T}''$ conforme as condições impostas pela tabela $\mathscr{A}$.
  - Dados quantos vértices existem numa solução de instância prévia $\mathscr{A}[i-1,\tilde{\mathscr{T}}]$, a recorrência expande a análise em mais uma classe (C_i) e tenta adicionar os vértices das classes de gêmeos em $\mathscr{T}''$ à solução. Os valores de $s(i,\mathscr{T}'')$ e $conf(\mathscr{T}'',\tilde{\mathscr{T}})$, explicados abaixo, determinarão se é possível e em quanto a solução aumenta. Caso não seja possível adicionar os vértices das classes de gêmeos em $\mathscr{T}''$, marcamos $-\infty$.

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

pois a adição de tais classes de gêmeos à solução resulta em distância maior que 2, violando a exigência do 2-CLUB.

- Em cada passo da programação dinâmica, devemos também computar $conf(\mathcal{T}'', \tilde{\mathcal{T}})$. Esse é definido como 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, e 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 $\tilde{\mathcal{T}} \cup \mathcal{T}'' = \mathcal{T}'$.
- 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.

O tempo de execução é limitado a $\mathcal{O}(2^k * 3^{2^k} * nm)$: temos 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.

5 Pergunta/Exercício

No quinto passo do algoritmo descrito na seção 4, 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.