

# Reconhecimento de Grafos Fino de Precedência

Flavia Bonomo<sup>3</sup>, Fabiano S. Oliveira<sup>2\*</sup>, Moysés S. Sampaio Jr.<sup>1†</sup>, Jayme L. Szwarcfiter<sup>1,2‡</sup>

<sup>1</sup>COPPE/PESC – Universidade Federal do Rio de Janeiro (UFRJ), Brasil

<sup>2</sup>IME – Universidade do Estado do Rio de Janeiro (UERJ), Brasil

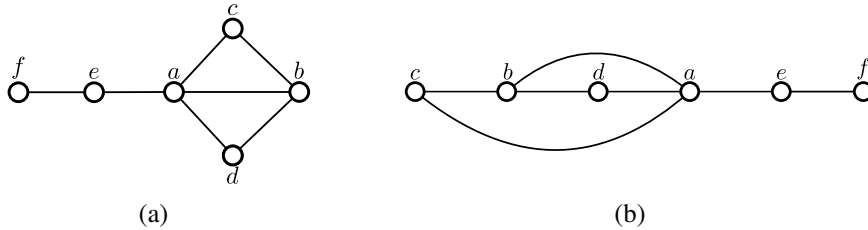
<sup>3</sup>DC-FCEN/ICC-CONICET, Universidad de Buenos Aires, Argentina

[moysessj, jayme]@cos.ufrj.br, fabiano.oliveira@ime.uerj.br, fbonomo@dc.uba.ar

**Abstract.** *The class of  $k$ -thin graphs have recently been introduced generalizing interval graphs. The complexity of the recognition of  $k$ -thin is open, even for fixed  $k \geq 2$ . We introduce a subclass of the  $k$ -thin graphs, called  $k$ -thin of precedence, presenting an efficient recognition algorithm based on PQ trees.*

## 1. Introdução

Um *grafo de intervalo*  $G$  é um grafo tal que  $V(G)$  é uma família de intervalos fechados da reta real, chamado de *modelo*, tal que  $(I, J) \in E(G)$  se, e somente se,  $I \cap J \neq \emptyset$ . Em [Olariu 1991], mostra-se que um grafo  $G$  é de intervalo se, e somente se, existir uma ordenação  $s$  de  $V(G)$  tal que, para qualquer tripla ordenada  $(p, q, r)$  de  $s$  (isto é,  $p < q < r$  em  $s$ ), se  $(p, r) \in E(G)$ , então  $(q, r) \in E(G)$ . Tal ordenação  $s$  é chamada *canônica*. A Figura 1 ilustra um grafo de intervalo e uma de suas ordens canônicas.



**Figura 1. Grafo de intervalo (a) e ordem canônica  $c < b < d < a < e < f$  de  $G$  (b).**

Um grafo é dito  *$k$ -fino* [Mannino et al. 2007] se existir um  $k$ -particionamento e uma ordenação  $s$  de  $V(G)$  tais que, para qualquer tripla ordenada  $(p, q, r)$  de  $s$ , se  $p$  e  $q$  pertencerem a uma mesma parte e  $(p, r) \in E(G)$  então,  $(q, r) \in E(G)$ . Tal ordenação é denominada *consistente*. A *finura* de um grafo  $G$  corresponde ao menor  $k$  para o qual  $G$  é  $k$ -fino (sempre existe um). Há grafos com valores arbitrários de finura e os grafos 1-finos constituem a classe dos grafos de intervalo, sendo a finura um indicativo do quão distante está um grafo de ser de intervalo. Dado um grafo  $G$ , determinar um particionamento mínimo de  $V(G)$  para o qual uma dada ordenação seja consistente pode ser resolvido em tempo polinomial [Bonomo et al. 2011], enquanto que determinar a existência de uma ordem consistente para um dado particionamento de  $V(G)$  é NP-completo [Bonomo, de Estrada 2018]. Considerando o problema geral, reconhecer se um grafo é  $k$ -fino, mesmo para  $k > 1$  fixo, continua em aberto.

\*parcialmente financiado por FAPERJ.

†financiado por CAPES.

‡parcialmente financiado por CNPq.

Neste trabalho, restringimos o problema em questão através da condição de que os vértices que pertençam a uma mesma parte tenham que ser consecutivos na ordem consistente. Definimos um grafo que possui essa característica, para um dado  $k$ -particionamento de  $V(G)$ , como um grafo  $k$ -fino de precedência ( $k$ -FP). De forma análoga, definimos uma ordem consistente desse tipo como uma ordem *consistente de precedência*. A Figura 2(a) ilustra um grafo que é 2-FP e, na Figura 2(b), é ilustrado um grafo que é 2-fino mas não é 2-FP com respeito a bipartição escolhida (os vértices de cada parte da bipartição são dispostos em uma mesma horizontal e com uma mesma cor e a ordem consistente implícita pela figura é obtida percorrendo-se os vértices da esquerda para direita, de baixo para cima).

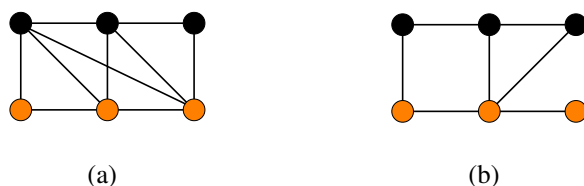


Figura 2. Exemplos de um grafo (a) 2-FP e (b) não 2-FP para dada bipartição.

## 2. Reconhecimento dos Grafos $k$ -FP com dada partição

O problema abordado nesse artigo é definido formalmente a seguir:

---

<b>PROBLEMA:</b>	Reconhecimento dos grafos $k$ -FP com partição dada
<b>ENTRADA:</b>	Um grafo $G$ e uma partição $(V_1, \dots, V_k)$ de $V(G)$
<b>QUESTÃO:</b>	Existe uma ordenação $s$ de $V(G)$ consistente tal que os vértices de $V_i$ sejam consecutivos em $s$ , para todo $1 \leq i \leq k$ ?

---

Nesta seção, apresentaremos um algoritmo eficiente para tal problema.

Uma *árvore PQ* [Booth, Lueker 1976] é uma árvore ordenada que descreve uma família de permutações para um dado conjunto  $\mathcal{C}$ . Nela, o conjunto das folhas é  $\mathcal{C}$  e os nós internos são de dois tipos:  $P$  ou  $Q$ . A família de permutações sendo representada é toda ordenação das folhas, lidas da esquerda para direita, que se pode obter através de permutações de filhos de nós  $P$  e reversões de filhos de nós  $Q$ . Árvores  $PQ$  podem ser usadas para reconhecer grafos de intervalo [Booth, Lueker 1976]. As ordens canônicas de um grafo de intervalo podem ser relacionadas a uma ordenação de suas cliques maximais. Além disso, pode-se mostrar que essas ordenações coincidem com aquelas representadas por uma árvore  $PQ$ , na qual as folhas representam as cliques maximais desse grafo [Booth, Lueker 1976]. Consequentemente, é possível representar todas as ordens canônicas de um grafo de intervalo por uma árvore  $PQ$ . Quanto a representação gráfica de uma árvore  $PQ$ , as folhas e os nós do tipo  $P$  são representados por círculos, enquanto os nós do tipo  $Q$  são representados por retângulos. A Figura 3(a) mostra uma árvore  $PQ$  associada ao grafo de intervalo da Figura 1(a). Uma árvore  $PQ$  de um grafo de intervalo pode ser construída em tempo linear.

Seja  $G$  um grafo  $k$ -FP com partição  $(V_1, \dots, V_k)$  e seja  $s$  uma ordem consistente de precedência de  $V(G)$ . Considere o par  $V_i, V_j \in \{V_1, \dots, V_k\}$  tal que todos os vértices de  $V_i$  precedam todos os vértices  $V_j$  em  $s$ . Pela definição dos grafos  $k$ -finos, os vértices de  $V_j$  impõem uma relação de precedência aos vértices de  $V_i$  dada pela propriedade a seguir. **Propriedade 1.** *Sejam  $(V_1, \dots, V_k)$  uma partição de  $V(G)$ ,  $s$  uma ordem consistente de precedência correspondente e  $1 \leq i < j \leq k$ . Se  $V_i$  precede a  $V_j$  em  $s$ , então para todo  $u, v \in V_i$  e  $w \in V_j$ , se  $(u, w) \notin E(G)$  e  $(v, w) \in E(G)$ , então  $u$  precede  $v$  em  $s$ .*

Seja  $s_T = C_1, C_2, \dots, C_q$  a ordenação de cliques maximais de um grafo de intervalo  $G$  obtidas de  $T$ , uma árvore  $PQ$ . É possível obter uma ordem canônica  $s$  de  $G$  a partir de  $s_T$  através do seguinte método. Inicialmente considere  $s$  uma sequência vazia. Para cada clique  $C_i$ ,  $1 \leq i \leq q$ , adicione ao final de  $s$  os vértices de  $C_i$  simpliciais em  $G$ , e remova-os de  $G$ . No fim desse processo,  $s$  conterá todos os vértices de  $G$ . Claramente  $s$  satisfaz à ordenação de cliques definida por  $T$ , além disso, é possível mostrar que qualquer ordem canônica que respeite essa ordenação pode ser obtida por esse mesmo método. Sejam  $G$  um grafo de intervalo,  $T$  uma árvore  $PQ$  de  $G$  e  $u, v \in V(G)$ . Dizemos que  $T$  é *compatível* com a relação de precedência  $u < v$  se for possível obter uma ordem canônica  $s$  a partir de  $T$  tal que  $u < v$  em  $s$ . Adicionalmente, diremos que  $u \in V(G)$  pertence a um nó de  $T$  se ele pertencer a alguma clique que descenda desse nó.

**Teorema 1.** *Sejam  $G$  um grafo de intervalo e  $T$  uma árvore  $PQ$  de  $G$ . Sejam  $X$  um nó de  $T$  com filhos  $X_1, \dots, X_k$ ,  $T_X$  a subárvore enraizada em  $X$  e  $u < v$  uma relação de precedência com  $u, v \in X$ . As seguintes afirmações são verdadeiras.*

- (i) *Se  $v$  pertence a todas as folhas de  $T_X$ , então  $T_X$  é compatível com  $u < v$ .*
- (ii) *Sejam  $X_i, X_j \in \{X_1, \dots, X_k\}$ . Se  $u \in X_i$ ,  $v \notin X_i$ ,  $v \in X_j$  e  $T_X$  é compatível com  $u < v$ , então  $X_i$  precede  $X_j$  em  $T_X$ .*

A seguir, delineamos a ideia geral do algoritmo de reconhecimento. Naturalmente, os grafos induzidos por  $V_1, \dots, V_k$  devem ser grafos de intervalo. Caso isso não seja verdade, o algoritmo responde NÃO. Em caso positivo, o algoritmo utiliza uma estratégia gulosa para confirmar se existe uma ordem consistente de precedência válida. Essa estratégia consiste em tentar cada parte como primeira na ordenação consistente de precedência. Ao supor que uma parte  $V_i$  é a primeira, as outras partes impõem relações de precedência aos vértices de  $V_i$  (Propriedade 1). Em seguida, o algoritmo verifica a existência de uma árvore  $PQ$  de  $G[V_i]$  que seja compatível com cada relação de precedência imposta. Se existir, então de fato  $V_i$  pode ser a primeira parte de uma ordenação consistente de precedência e o processo se repete para a escolha das demais partes. Caso contrário,  $V_i$  não pode ser a primeira e tenta-se a próxima parte. Essa estratégia é suficiente pois, pela Propriedade 1, as relações de ordem impostas sobre os vértices de uma classe são oriundas das classes que sucedem a mesma na ordem consistente.

Seja  $P = \hat{V}_1, \hat{V}_2, \dots, \hat{V}_k$  uma permutação das partes de  $G$ , é possível determinar se  $\hat{V}_i$  pode vir antes de  $\hat{V}_j$ , para  $i < j$  em  $P$ , como descrito a seguir. Construa uma árvore  $PQ$   $T_i$  para  $G[\hat{V}_i]$ . Sejam  $\{u, v\} \in \hat{V}_i$  tais que  $u$  deva preceder  $v$  pela regra imposta pela Propriedade 1. Para cada ancestral comum de  $u, v$  em  $T_i$ , utilize arestas direcionadas para definir uma ordem entre os vértices de  $T_i$  de acordo com o Teorema 1. No final do processo, se não houver ciclos nas orientações definidas significa que os vértices de  $\hat{V}_i$  podem preceder os vértices de  $\hat{V}_j$  em uma ordem consistente de precedência. Caso contrário, não existe nenhuma ordem de precedência tal que os vértices de  $\hat{V}_i$  possam preceder os vértices de  $\hat{V}_j$ . Considerando que seja encontrado uma permutação  $P$  onde todas as partes passem no teste anterior, defina  $T'_i$  como uma árvore  $PQ$  obtida a partir de  $T_i$  considerando as arestas direcionadas adicionadas durante o processo. Seja  $s_i$  uma ordem canônica obtida a partir de  $T'_i$ . Então,  $s = s_1 \dots s_k$  é uma ordem consistente de precedência de  $V(G)$ .

Como um exemplo do processo anterior, tome o grafo  $G$  definido na Figura 4 e o 3-particionamento  $(V_1, V_2, V_3)$  de  $V(G)$  onde  $V_1 = \{a, b, c, d, e, f\}$ ,  $V_2 = \{a', b', c', d', e', f'\}$

e  $V_3 = \{a'', b'', c'', d'', e'', f''\}$ . A Figura 4(a) representa a permutação  $P = V_1, V_2, V_3$  enquanto a Figura 4(b) representa a permutação  $P' = V_3, V_2, V_1$ . Neste exemplo, validaremos se  $V_1$  pode ser a primeira parte em  $P$ . Temos que  $G[V_1]$  é isomorfo ao grafo da Figura 1(a) com cliques maximais  $C_1 = \{a, b, c\}$ ,  $C_2 = \{a, b, d\}$ ,  $C_3 = \{a, e\}$  e  $C_4 = \{e, f\}$ . Além disso, compatível com a árvore  $PQ$  da Figura 3(a). Pela Propriedade 1, temos que  $a$  deve suceder todos os outros vértices de  $V_1$  em uma ordem que seja consistente de precedência. Considerando essas relações de precedência, a Figura 3(b) mostra as arestas direcionadas adicionadas à árvore  $PQ$  utilizando-se o Teorema 1. Nesta figura, as arestas azuis são as adicionadas pelo teorema e as laranjas são as inferidas pela natureza do nó  $Q$ . Isto é, como a raiz é um nó  $Q$ , a aresta direcionada azul implica na aresta direcionada laranja. Note que o vértice  $a$  pertence a todas as cliques com exceção da  $C_4$ , por isto não existe nenhuma aresta direcionada azul entre as mesmas (item (i) do Teorema 1). Como não há ciclos na árvore  $PQ$  resultante, isto significa que  $V_1$  pode preceder  $V_2$  e  $V_3$ . A Figura 3(c) mostra a árvore  $PQ$  resultante de  $V_3$  considerando a permutação  $P'$ , que indica que  $V_3$  não pode preceder  $V_1$  e  $V_2$ . Este algoritmo é  $O(k^2n^4)$ .

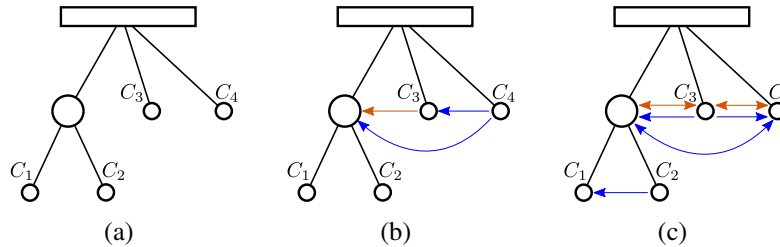


Figura 3. Árvores  $PQ$ : (a) ordinária, (b) compatível e (c) incompatível.

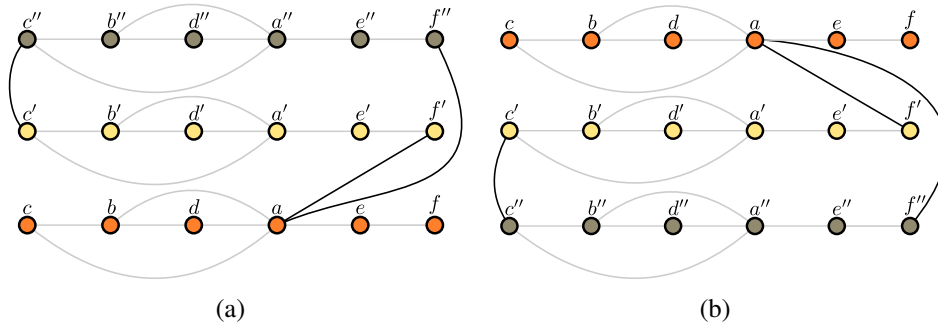


Figura 4. Permutações (a)  $P$  e (b)  $P'$

## Referências

- Bonomo, F., de Estrada, D. (2018). On the thinness and proper thinness of a graph. *Discrete Applied Mathematics*. In Press.
- Bonomo, F., Mattia, S., Oriolo, G. (2011). Bounded coloring of co-comparability graphs and the pickup and delivery tour combination problem. *Theoretical Computer Science*, 412(45):6261–6268.
- Booth, K. S., Lueker, G. S. (1976). Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms. *Journal of Computer and System Sciences*, 13(3):335–379.
- Mannino, C., Oriolo, G., Ricci, F., Chandran, S. (2007). The stable set problem and the thinness of a graph. *Operations Research Letters*, 35:1–9.
- Olariu, S. (1991). An optimal greedy heuristic to color interval graphs. *Information Processing Letters*, 37:21–25.