Combinatorial Problems on H-graphs

Steven Chaplick

Lehrstuhl für Informatik I, Universität Würzburg, Germany,
Email: `first.last@uni-wuerzburg.de`

Peter Zeman

Department of Applied Mathematics, Faculty of Mathematics and Physics,
Charles University in Prague, Czech Republic,
Email: `zeman@kam.mff.cuni.cz`

Abstract

Biró, Hujter, and Tuza introduced the concept of H-graphs (1992), intersection graphs of connected subgraphs of a subdivision of a graph H. They naturally generalize many important classes of graphs, e.g., interval graphs and circular-arc graphs. We continue the study of these graph classes by considering coloring, clique, and isomorphism problems on H-graphs.

We show that for any fixed H containing a certain 3-node, 6-edge multigraph as a minor that the clique problem is APX-hard on H-graphs and the isomorphism problem is isomorphism-complete. We also provide positive results on H-graphs. Namely, when H is a cactus the clique problem can be solved in polynomial time. Also, when a graph G has a Helly H-representation, the clique problem can be solved in polynomial time. Finally, we observe that one can use treewidth techniques to show that both the k-clique and list k-coloring problems are FPT on H-graphs. These FPT results apply more generally to *treewidth-bounded* graph classes where treewidth is bounded by a function of the clique number.

Keywords: intersection graphs, clique, isomorphism, coloring, treewidth.

1 Introduction

An intersection representation of a graph assigns a set to each vertex and uses intersections of those sets to encode its edges. More formally, an intersection representation $\mathcal{R}$ of a graph G is a collection of sets $\{R_v\}_{v \in V(G)}$ such that $R_u \cap R_v \neq \emptyset$ if and only if $uv \in E(G)$. Many important classes of graphs arise from restricting the sets R_v to geometric objects (e.g., intervals, convex sets).

We study *H-graphs*, intersection graphs of connected subsets of a fixed topological pattern given by a graph H, introduced by Biró, Hujter, and Tuza [1]. We obtain new algorithmic results on clique, coloring, and isomorphism problem. In a companion paper [7], we studied recognition and dominating set problems on H-graphs. We begin with related graph classes.

Interval graphs (INT) form one of the most studied and well-understood classes of intersection graphs. In an *interval representation*, each set R_v is a closed interval of the real line; see Fig. 1a. A primary motivation for studying interval graphs (and related classes) is the fact that many important computational problems can be solved in linear time on them; see for example [4,6,17].

A graph is *chordal* when it does not have an induced cycle of length at least four. Equivalently, as shown by Gavril [12], a graph is chordal if and only if it can be represented as an intersection graph of subtrees of some tree; see Fig. 1b. This immediately implies that INT is a subclass of the chordal graphs (CHOR). Some important problems (e.g., dominating set [3] and graph isomorphism [17]) are harder on chordal graphs than on interval graphs.

The *split graphs* (SPLIT) form an important subclass of chordal graphs. These are the graphs that can be partitioned into a clique and an independent set. Note that every split graph is an intersection graph of subtrees of a *star* S_d, where S_d is the complete bipartite graph $K_{1,d}$.

Circular-arc graphs (CARC) generalize interval graphs by having each set R_v be an arc of a circle. A graph G is a *Helly circular-arc graph* if the collection of circular arcs $\mathcal{R} = \{R_v\}_{v \in V(G)}$ satisfies *Helly property*, i.e., in each sub-collection of $\mathcal{R}$ whose sets pairwise intersect, the common intersection is non-empty. Interestingly, the coloring problem is NP-hard on Helly CARC [13].

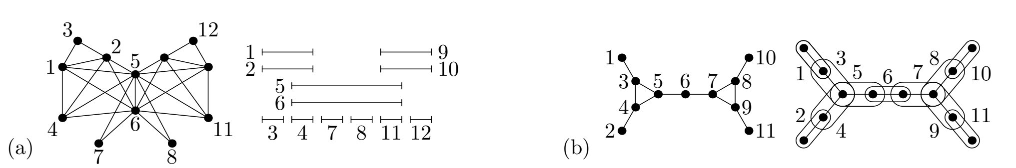

Fig. 1. (a) An interval graph and one of its interval representation. (b) A chordal graph and one of its representation as an intersection graph of subtrees of a tree.

H-graphs. Biró, Hujter, and Tuza [1] introduced *H-graphs*. Let H be a fixed graph. A graph G is an *intersection graph of* H if it is an intersection graph of connected subgraphs of H, i.e., the assigned subgraphs H_v and H_u of H share a vertex if and only if $uv \in E(G)$. A *subdivision* H' of a graph H is obtained when the edges of H are replaced by internally disjoint paths of arbitrary lengths. A graph G is a *topological intersection graph* of H if G is an intersection graph of a subdivision H' of H. We say that G is an *H-graph* and the collection $\{H'_v : v \in V(G)\}$ of connected subgraphs of H' is an *H-representation* of G. The class of all H-graphs is denoted by H-GRAPH. We have the following relations: INT $= K_2$-GRAPH, SPLIT $\subsetneq \bigcup_{d=2}^{\infty} S_d$-GRAPH, CARC $= K_3$-GRAPH, and CHOR $= \bigcup_{\{\text{Tree}\}\ \text{T}} T$-GRAPH. Moreover, for any pair of (multi-)graphs H_1 and H_2, if H_1 is a *minor* of H_2, then H_1-GRAPH $\subseteq$ H_2-GRAPH. If H_1 is a subdivision of H_2, then H_1-GRAPH $= H_2$-GRAPH.

H-graphs were introduced in the context of the (p, k) *pre-coloring extension problem*. Here, one is given a graph G together with a p-coloring of $W \subseteq V(G)$, and the goal is to find a proper k-coloring of G extending this *pre-coloring*. PrColExt(k, k) has an XP algorithm (in k and $\|H\|$) for H-GRAPH.

Coloring H-graphs. From the above discussion, we note a dichotomy regarding computing a minimum coloring on H-GRAPH. Namely, if H contains a cycle, then computing a minimum coloring in H-GRAPH is already NP-hard even for Helly H-GRAPH. Additionally, when H is acyclic, a minimum coloring can be computed in linear time since H-GRAPH is a subclass of CHOR.

Our Results. We prove that for any fixed H containing a *double triangle* (depicted in Fig. 2) as a minor, the clique problem is APX-hard on H-graphs and the isomorphism problem is isomorphism-complete (see Section 3). We also provide positive results on H-graphs in Sections 2 and 4. Namely, when a graph G has a Helly H-representation, the clique problem can be solved in polynomial time (see Theorem 2.2). Also, when H is a cactus the clique problem can be solved in polynomial time (see Theorem 2.4). Finally, we use treewidth techniques to show that both the k-clique and list k-coloring problems are FPT on H-graphs (see Propositions 4.3 and 4.2 respectively). These FPT results extend to *treewidth-bounded* graph classes.

2 Finding Cliques in H-graphs

This section concerns cases where the clique problem can be solved efficiently on H-GRAPH, for a fixed graph H. First, we consider a case where we have a "nice" representation but H is arbitrary. Second, we restrict H to be a cactus.

Helly H-graphs. A *Helly H*-graph G has an H-representation $\{H'_v : v \in V(G)\}$ such that the collection $\mathcal{S} = \{V(H'_v) : v \in V(G)\}$ satisfies the *Helly property*, i.e., for each sub-collection of $\mathcal{S}$ whose sets pairwise intersect, their common intersection is non-empty. Notice that, when H is a tree, every H-representation satisfies the Helly property. Furthermore, when a graph G has a Helly H-representation, we obtain the following relationship between the size of H and the number of maximal cliques in G.

Lemma 2.1 *Each Helly H-graph G has at most $|V(H)| + |E(H)| \cdot |V(G)|$ maximal cliques.*

Proof. Let H' be a subdivision of H such that G has a Helly H-representation $\{H'_v : v \in V(G)\}$. Note that, for each maximal clique C of G, $\bigcap_{v \in C} H'_v \neq \emptyset$, i.e., C corresponds to a node x_C of H'. For every edge $xy \in E(H)$, we consider the corresponding path $P = (x, x_1, \ldots, x_k, y)$ in H'. Let G_{xy} be the subgraph of G formed by maximal cliques of G which "occur" on P. The graph G_{xy} is a Helly cicular-arc graph. Now, since Helly circular arc graphs have at most linearly many maximal cliques [11], G has at most $|V(H)| + |E(H)| \cdot |V(G)|$ maximal cliques. □

We can now use Lemma 2.1 to find the largest clique in G in polynomial time. In fact, we can do this without needing to compute a representation of G. In particular, the maximal cliques of a graph can be enumerated with polynomial delay [18]. Thus, since G has at most linearly many maximal cliques, we can simply list them all in polynomial time and report the largest, i.e., if the enumeration process produces too many maximal cliques, we know that G has no Helly H-representation. This provides the following theorem.

Theorem 2.2 *The clique problem is polytime solvable on Helly H-graphs.*

Note that some co-bipartite circular arc graphs have have exponentially many maximal cliques and as such are not contained in Helly H-graphs for any fixed H. However, the clique problem is polytime solvable on CARC [15].

Cactus-graphs. The clique problem is efficiently solvable on chordal graphs [14] and circular arc graphs [15]. In particular, when H is either a tree or a cycle, the clique problem can be solved in polynomial time independent of the size of H. In Theorem 2.4, we observe that these results easily generalize to the case when G is in H-GRAPH for some cactus H. With this in mind, we say that such a graph G belongs to the class *cactus*-GRAPH, where *cactus*-$\mathsf{GRAPH} = \bigcup\{H\text{-}\mathsf{GRAPH} : H \text{ is a cactus.}\}$.

To prove the result we will use the *clique-cutset decomposition* – which is defined as follows. A *clique-cutset* of a graph G is a clique K in G such that $G \setminus K$ has more connected components than G. An *atom* is a graph without a clique-cutset. An *atom of a graph* G is an induced subgraph A of G which is an atom. A *clique-cutset decomposition* of G is a set $\{A_1, \ldots, A_k\}$ of atoms of G such that $G = \bigcup_{i=1}^{k} A_i$ and for every i, j, $V(A_i) \cap V(A_j)$ is either empty or induces a clique in G. Algorithmic aspects of clique-cutset decompositions were studied by Whitesides [22] and Tarjan [21]. In particular, if $k \leq n$, then for any graph G a clique-cutset decomposition $\{A_1, \ldots, A_k\}$ of G can be computed in $O(n \cdot (n+m))$ [21]. Additionally, to solve the clique problem on a graph G it suffices to solve it for each atom of G from a clique-cutset decomposition [22,21]. Theorem 2.4 now follows from the following easy lemma and the fact that the clique problem can be solved in polynomial time on circular arc graphs [15].

Lemma 2.3 *If $G \in$ cactus-*GRAPH*, then each atom A of G is in* CARC*.*

Proof. Consider an H-representation $\{H_v : v \in V(G)\}$ of G where H is a cactus. Now let $H|_A = \bigcup_{v \in V(A)} H_v$. Clearly, if $H|_A$ is a path or a cycle, then we are done. Otherwise, $H|_A$ must contain a cut-node x. Let $C_1, \ldots, C_t$ be the components of $H|_A \setminus \{x\}$, and let S be the vertices of A whose representations contain x. Note that S is a clique in A. Moreover, since A is an atom, S is not a clique-cutset. Thus, there is a component C_j such that the subgraph H' of H induced by $V(C_j) \cup \{x\}$ provides a representation of A. In particular, if H' is either a cycle or a path we are again done. Moreover, when H' is neither a path nor a cycle, repeating this argument on H' provides a smaller subgraph of H on which A can be represented, i.e., this eventually produces either a path or cycle. □

Theorem 2.4 *The clique problem can be solved in polynomial time on the class cactus-*GRAPH*.*

3 Clique and Isomorphism Hardness Results

To obtain our hardness results we show that there are graphs H such that the complement of a 2-subdivision of every graph is an H-graph. The 2-subdivision of a graph G is the result of subdividing every edge of G twice. The complement of a graph G is denoted by $\overline{G}$. We use SUBD_2 to denote the class of all 2-subdivisions of graphs and $\overline{\mathsf{SUBD}_2}$ to denote their complements.

This seemingly esoteric family of graphs is interesting for two reasons. The first is that graph isomorphism is closed under k-subdivision and complement

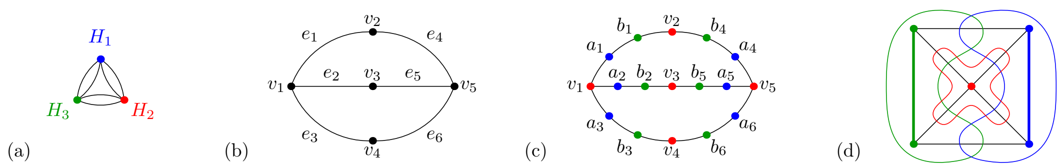

Fig. 2. (a) The *double triangle* graph. (b) A graph G. (c) The 2-subdivision G^* of G. A three-clique cover of $\overline{G^*}$ is indicated by colors. (d) The 4-wheel graph (which contains the double triangle as a minor) and a *sketch* of our H-representation of $\overline{G^*}$. For example, the edges between the green clique and the blue clique are represented where the green and blue regions intersect.

operations. Thus, isomorphism testing in $\overline{\mathsf{SUBD}_2}$ is as hard as it is for general graphs, i.e., the class $\overline{\mathsf{SUBD}_2}$ is *isomorphism-complete*. The second is that the clique problem is APX-hard on $\overline{\mathsf{SUBD}_2}$. More specifically, Chlebík and Chlebíková [8] proved that the maximum independent set problem is APX-hard on the class of $2k$-subdivisions of 3-regular graphs for any fixed integer $k \geq 0$; in particular, for 2-subdivisions. Thus, showing that $\overline{\mathsf{SUBD}_2} \subseteq H\text{-}\mathsf{GRAPH}$ for a fixed H, implies that the maximum clique problem is APX-hard on H-GRAPH and that H-GRAPH is isomorphism-complete.

Theorem 3.1 *If H contains the graph in Fig. 2a as a minor, then* $\overline{\mathsf{SUBD}_2} \subseteq H\text{-}\mathsf{GRAPH}$.

Proof. Since H contains the graph in Fig. 2 as a minor, it can be partitioned into three connected subgraphs H_1, H_2, H_3 such that there are at least two edges connecting H_i and H_j for each $i \neq j$. For every graph G, we show that the complement of its 2-subdivision has and H-representation.

The construction proceeds similarly to the constructions used by Francis et al. [10], and we borrow their convenient notation. Let G be a graph with vertex set $\{v_1, \ldots, v_n\}$ and edge set $\{e_1, \ldots, e_m\}$. If $e_k \in E(G)$ and $e_k = v_i v_j$ where $i < j$, we define $l(k) = i$ and $r(k) = j$ (as if v_i and v_j were respectively the *left* and *right* ends of e_k). In the 2-subdivision G^* of G, the edge e_k of G is replaced by the path $(v_{l(k)}, a_k, b_k, v_{r(k)})$; see Fig. 2a and Fig. 2b.

Note that $\overline{G^*}$ can be covered by three cliques, i.e., $C_v = \{v_1, \ldots, v_n\}$, $C_a = \{a_1, \ldots, a_m\}$, and $C_b = \{b_1, \ldots, b_m\}$. We now describe a subdivision H' of H which admits an H-representation $\{H'_v : v \in V(\overline{G^*})\}$ of $\overline{G^*}$. We obtain H' by subdividing the six edges connecting H_1, H_2, and H_3. Specifically:

- we n-subdivide the two edges connecting H_1 to H_2 to obtain two paths $P_{12} = (\alpha_0, \alpha_1, \ldots, \alpha_n, \alpha_{n+1})$, $Q_{12} = (\beta_0, \beta_1, \ldots, \beta_n, \beta_{n+1})$ where $\alpha_0, \beta_0 \in H_1$ and $\alpha_{n+1}, \beta_{n+1} \in H_2$, and

- we n-subdivide the two edges connecting H_1 to H_3 to obtain two paths $P_{13} = (\gamma_0, \gamma_1, \ldots, \gamma_n, \gamma_{n+1})$, $Q_{13} = (\eta_0, \eta_1, \ldots, \eta_n, \eta_{n+1})$ where $\gamma_0, \eta_0 \in H_1$ and $\gamma_{n+1}, \eta_{n+1} \in H_2$.
- m-subdivide the two edges connecting H_2 and H_3 to obtain two paths $P_{23} = (\mu_0, \mu_1, \ldots, \mu_m, \mu_{m+1})$, $Q_{23} = (\nu_0, \nu_1, \ldots, \nu_m, \nu_{m+1})$ where $\mu_0, \nu_0 \in H_2$ and $\mu_{m+1}, \eta_{m+1} \in H_2$.

We now describe each H_{v_i}, H_{a_j} and H_{b_j}. The idea is that H'_{v_i} will contain H_1 and extend from the "start" of P_{12} up to the position i, and from the "start" of Q_{12} up to position $(n-i)$. From the other side, each H'_{a_j} will contain H_2 and extend from the "end" of P_{12} down to position $(l(j)+1)$, and from the end of Q_{12} down to position $(n - l(j) + 1)$; an example is sketched in Fig. 2d. In this way, we ensure that H'_{a_j} does not intersect $H'_{v_{l(j)}}$ while H'_{a_j} does intersect every H'_{v_i} for $i \neq l(j)$. The other pairs proceed similarly, and we describe the subgraphs $H_{v_i}, H_{a_j}, H_{b_j}$ for each $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, m\}$ as follows:

- $H'_{v_i} = H_1 \cup \{\alpha_1, \ldots, \alpha_i\} \cup \{\beta_1, \ldots, \beta_{n-i}\} \cup \{\gamma_1, \ldots, \gamma_i\} \cup \{\eta_1, \ldots, \eta_{n-i}\}$.
- $H'_{a_j} = H_2 \cup \{\alpha_n, \ldots, \alpha_{l(j)+1}\} \cup \{\beta_n, \ldots, \beta_{n-l(j)+1}\} \cup \{\mu_1, \ldots, \mu_j\} \cup \{\nu_1, \ldots, \nu_{m-j}\}$.
- $H'_{b_j} = H_3 \cup \{\gamma_n, \ldots, \gamma_{r(j)+1}\} \cup \{\eta_n, \ldots, \eta_{n-r(j)+1}\} \cup \{\mu_m, \ldots, \mu_{j+1}\} \cup \{\nu_m, \ldots, \nu_{m-j+1}\}$.

□

Recall that, Theorem 2.4 states that the clique problem can be solved in polynomial time on cactus-graphs. Thus, the open cases which remain are when H is not a cactus (i.e., H contains a diamond as a minor), but H does not satisfy the conditions of Theorem 3.1. On the other hand, while the isomorphism problem can be solved in linear time on interval graphs and Helly circular-arc graphs [20], it is isomorphism-complete on split graphs [17]. Many questions remain open for the complexity status of the isomorphism problem on H-GRAPH, even for the simplest non-chordal case, circular-arc graphs [20].

4 FPT Results via Treewidth-bounded Graph Classes

In this section we discuss the concept of *treewidth-bounded* graph classes. We will use the fact that the class H-GRAPH has "well-behaved" treewidth (see Lemma 4.1) together with some observations about more general *treewidth-bounded* graph classes to study optimization problems on H-GRAPH.

Treewidth was introduced by Robertson and Seymour [19]. A *tree decomposition* of a graph $G = (V, E)$ is a pair (X, T), where T is a tree and $X = \{X_i \mid i \in V(T)\}$ is a family of subsets of V, called *bags*, such that (1) for all $v \in V$, the set of nodes $T_v = \{i \in V(T) \mid v \in X_i\}$ induces a non-empty connected subtree of T, and (2) for each edge $e = \{u, v\} \in E(G)$ there exists $i \in V(T)$ such that both u and v are in X_i. The maximum of $|X_i| - 1$, $i \in V(T)$, is called the *width* of the tree decomposition. The *treewidth*, $tw(G)$, of a graph G is the minimum width over all tree decompositions of G.

An easy lower bound on the treewidth of a graph G is the size of the largest clique in G, i.e., its clique number $\omega(G)$. This follows from the fact that each edge of G belongs to some bag of T and that a collection of pairwise intersecting subtrees of a tree must have a common intersection (i.e., they satisfy the Helly property). With this in mind, we say that a graph class $\mathcal{G}$ is *treewidth-bounded* if there is a function $f : \mathbb{N} \to \mathbb{N}$ such that for every $G \in \mathcal{G}$, $tw(G) \leq f(\omega(G))$. This concept generalizes the idea of $\mathcal{G}$ being *χ-bounded*, namely, that the *chromatic number* $\chi(G)$ of every graph $G \in \mathcal{G}$ is bounded by a function of the clique number of G. In particular, the chromatic number of a graph G is bounded by its treewidth since a tree decomposition G is a tree representation of a chordal supergraph G' of G where $\omega(G') = tw(G) + 1$, i.e., $\chi(G') = tw(G) + 1$ since chordal graphs are perfect. It was recently shown that the graphs which do not contain *even holes* (i.e., cycles of length $2k$ for any $k \geq 2$) and *pans* (i.e., cycles with a single pendent vertex attached) as induced subgraphs are treewidth bounded by $f(\omega) = \frac{3}{2}\omega - 1$ [5].

For a function $f : \mathbb{N} \to \mathbb{N}$, we use $\mathcal{G}_f$ to denote the class of graphs G where $tw(G) \leq f(\omega(G))$. Each class H-GRAPH is known to be a subclass of $\mathcal{G}_f$ for certain linear functions f, as in the following lemma of Biro et al [1].

Lemma 4.1 *[1] For every $G \in H$-GRAPH, $tw(G) \leq (tw(H) + 1) \cdot \omega(G) - 1$, i.e., H-GRAPH $\subseteq \mathcal{G}_{f_H}$ where $f_H(\omega) = (tw(H) + 1) \cdot \omega - 1$.*

We now apply the existing literature to describe the computational complexity of k-coloring problems as well as the k-clique problem on treewidth-bounded graph classes, and, in particular, the H-GRAPH classes.

For each fixed $k \geq 3$, it is also known that testing for a *(k, k)-pre-colouring extension* in the class H-GRAPH can be done in XP time [1]. They use Lemma 4.1 together with a simple argument to obtain their result. We use a similar argument together with a more recent result regarding bounded treewidth graphs to observe that an even more general problem, *list k-coloring* (where each list is a subset of $\{1, \ldots, k\}$), is FPT on any treewidth-bounded graph class, and as such also on H-GRAPH, i.e., Proposition 4.3. We first show that the k-clique problem is FPT on any treewidth-bounded graph class.

Proposition 4.2 *For any computable function $f : \mathbb{N} \to \mathbb{N}$, the k-clique problem can be solved in $\mathcal{O}((5 \cdot f(k))^{5 \cdot f(k)} \cdot n)$ time on $\mathcal{G}_f$. Thus, for H-GRAPH, the k-clique problem can be solved in $\mathcal{O}((5 \cdot tw(H) \cdot k)^{5 \cdot tw(H) \cdot k} \cdot n)$ time.*

Proof. To test if G contains a k-clique, we first try to generate a tree decomposition of G with width roughly $f(k)$ via a recent algorithm [2] which, for any given graph G and number t, provides a tree decomposition of width at most $5 \cdot t$ or states that the treewidth of G is larger than t – this algorithm runs in $2^{\mathcal{O}(t)} \cdot n$ time. If this algorithm provides tree decomposition, we use it to test whether G has a k-clique in $\mathcal{O}((5 \cdot f(k))^{5 \cdot f(k)} \cdot n)$ time via a known algorithm [9]. If not, then G must contain a k-clique, and we are done. □

Proposition 4.3 *For any function $f : \mathbb{N} \to \mathbb{N}$, the list-$k$-coloring problem can be solved in $\mathcal{O}(((5 \cdot f(k))^{5 \cdot f(k)} + k^{5 \cdot f(k)+2}) \cdot n)$ time on $\mathcal{G}_f$. Thus, for H-GRAPH, the list-k-coloring problem can be solved in $\mathcal{O}(((5 \cdot tw(H) \cdot k)^{5 \cdot tw(H) \cdot k} + k^{(5 \cdot tw(H) \cdot k)+2}) \cdot n)$ time.*

Proof. For fixed k, clearly, if G contains a clique of size $k+1$ then G has no k-coloring, i.e., no list-k-COL regardless of the lists. We use Proposition 4.2 to test for such a clique, and reject if one is found. Otherwise, we have a $5 \cdot f(k)$ width tree decomposition, and this time use it to solve the list-k-COL problem via the known $\mathcal{O}(n \cdot k^{t+2})$ time algorithm when given a width t tree decomposition [16], i.e., list-k-COL can be solved in $\mathcal{O}(n \cdot k^{5 \cdot k \cdot tw(H)+2})$-time on H-GRAPH. □

References

[1] Biró, M., M. Hujter and Z. Tuza, *Precoloring extension. i. interval graphs*, Discrete Mathematics **100** (1992), pp. 267–279.

[2] Bodlaender, H. L., P. G. Drange, M. S. Dregi, F. V. Fomin, D. Lokshtanov and M. Pilipczuk, *A c^k n 5-approximation algorithm for treewidth*, SIAM J. Comput. **45** (2016), pp. 317–378.

[3] Booth, K. S. and J. H. Johnson, *Dominating sets in chordal graphs*, SIAM J. Comput. **11** (1982), pp. 191–199.

[4] Booth, K. S. and G. S. Lueker, *Testing for the consecutive ones property, interval graphs, and planarity using PQ-tree algorithms*, J. Comput. System Sci. **13** (1976), pp. 335–379.

[5] Cameron, K., S. Chaplick and C. T. Hoáng, *On the structure of (pan, even hole)-free graphs*, (accepted) Journal of Graph Theory. (2017+), arXiv:1508.03062.

[6] Chang, M.-S., *Efficient algorithms for the domination problems on interval and circular-arc graphs*, SIAM J. Comput. **27** (1998), pp. 1671–1694.

[7] Chaplick, S., M. Töpfer, J. Voborník and P. Zeman, *On H-topological intersection graphs*, CoRR **abs/1608.02389** (2016), accepted to WG 2017.

[8] Chlebík, M. and J. Chlebíková, *The complexity of combinatorial optimization problems on d-dimensional boxes*, SIAM J. Disc. Math. **21** (2007), pp. 158–169.

[9] Courcelle, B., *The monadic second-order logic of graphs iii: Tree-decompositions, minors and complexity issues*, Informatique théorique et applications **26** (1992), pp. 257–286.

[10] Francis, M. C., D. Gonçalves and P. Ochem, *The maximum clique problem in multiple interval graphs*, Algorithmica **71** (2013), pp. 812–836.

[11] Gavril, F., *Algorithms on circular-arc graphs*, Networks **4** (1974), pp. 357–369.

[12] Gavril, F., *The intersection graphs of subtrees in trees are exactly the chordal graphs*, Journal of Combinatorial Theory, Series B **16** (1974), pp. 47–56.

[13] Gavril, F., *Intersection graphs of Helly families of subtrees.*, Discrete Appl. Math. **66** (1996), pp. 45–56.

[14] Golumbic, M. C., "Algorithmic graph theory and perfect graphs," Annals of Discrete Mathematics **57**, Elsevier, 2004.

[15] Hsu, W., *Maximum weight clique algorithms for circular-arc graphs and circle graphs*, SIAM J. Comput. **14** (1985), pp. 224–231.

[16] Jansen, K. and P. Scheffler, *Generalized coloring for tree-like graphs*, Discrete Applied Mathematics **75** (1997), pp. 135 – 155.

[17] Lueker, G. S. and K. S. Booth, *A linear time algorithm for deciding interval graph isomorphism*, Journal of the ACM (JACM) **26** (1979), pp. 183–195.

[18] Makino, K. and T. Uno, *New algorithms for enumerating all maximal cliques*, in: *Algorithm Theory-SWAT 2004*, LNCS **3111**, Springer, 2004 pp. 260–272.

[19] Robertson, N. and P. D. Seymour, *Graph minors. iii. planar tree-width*, Journal of Combinatorial Theory, Series B **36** (1984), pp. 49–64.

[20] Szwarcfiter, J. L., J. P. Spinrad, F. J. Soulignac, Y. Nussbaum, R. M. Mcconnell, M. C. Lin and A. R. Curtis, *Isomorphism of graph classes related to the circular-ones property*, Disc. Math. & Theoretical Computer Science **15** (2013).

[21] Tarjan, R. E., *Decomposition by clique separators*, Discrete Math. **55** (1985), pp. 221–232.

[22] Whitesides, S. H., *A method for solving certain graph recognition and optimization problems, with applications to perfect graphs*, in: *Topics on perfect graphs.*, 1984, pp. 281–298, annals of Discrete Mathematics 21.