Computational Complexity of the Krausz Dimension of Graphs

Petr Hliněný and Jan Kratochvíl *

Department of Applied Mathematics,
Faculty of Mathematics and Physics, Charles University
{hlineny,honza}@kam.ms.mff.cuni.cz

March 3, 1997

Abstract

A Krausz partition of a graph G is a partition of the edges of G into complete subgraphs. The Krausz dimension of a graph G is the least number k such that G admits a Krausz partition in which each vertex belongs to at most k classes. The graphs with Krausz dimension 2 are exactly the line graphs.

This paper studies the computational complexity of the Krausz dimension problem. We show that deciding if Krausz dimension of a graph is at most 3 is NP-complete in general, but solvable in polynomial time for graphs of maximum degree 4. We pay closer attention to chordal graphs, showing that deciding if Krausz dimension is at most 6 is NP-complete for chordal graphs in general, while the Krausz dimension of a chordal graph with bounded clique size can be determined in polynomial time. We also show that for any fixed k, it can be decided in polynomial time if an interval graph has Krausz dimension at most k.

*Research supported by Czech Research grants Nos. GAUK 194/1996 and GAČR 0194/1996.

1 Introduction

A *Krausz partition* of a graph $\boldsymbol{G}$ is a partition of the edge set $E(\boldsymbol{G})$ into complete subgraphs (that are also called the *clusters* of the partition). The number of clusters containing a vertex v is called the *order* of v (in the Krausz partition). The order of the partition is the maximum order over all vertices of $\boldsymbol{G}$. The *Krausz dimension* of $\boldsymbol{G}$ is defined as the minimum partition order over all Krausz partitions of $\boldsymbol{G}$, and denoted by $dim(\boldsymbol{G})$. Note that if $\boldsymbol{G}$ is not connected, its dimension is the maximum dimension over all of its components; so we consider only connected graphs in our paper.

Every graph can be partitioned just by taking each edge alone as a cluster — thus for every graph, $dim(\boldsymbol{G}) \leq \Delta(\boldsymbol{G})$. This bound is optimal for triangle-free graphs. On the other hand, complete graphs have dimension 1. Another important class of graphs, for which the Krausz dimension is known is described by Krausz's characterization of line graphs [6], which in fact inspired the notion of the Krausz dimension:

Theorem 1 *(Krausz) The Krausz dimension of a graph is at most* 2 *if and only if this graph is a line graph.*

Our paper studies the computational complexity of the Krausz dimension for various classes of graphs. The general problem of determining the dimension of a given graph is denoted by *KrauszDim*, the question whether the dimension is at most k is denoted by $KrauszDim(k)$, and the same question restricted to graphs with maximum degrees at most d is denoted by $KrauszDim(k, d)$. These decision forms of the Krausz dimension problem clearly belong to *NP*.

2 Computational complexity of the Krausz dimension

We summarize the main results of our paper in this section. We first consider general graphs and the relation between the maximum degree of a graph and its Krausz dimension.

Theorem 2 **(a)** *The problem* $KrauszDim(3, 4)$ *is solvable in polynomial time.*
(b) *The problem* $KrauszDim(3, 5)$ *is NP–complete, even when restricted to planar graph.*

Next we pay closer attention to various classes of chordal graphs. It may seem somewhat surprising that Krausz dimension remains difficult even for chordal graphs, though here we are only able to prove the hardness result for larger dimension:

Theorem 3 *The problem* $KrauszDim(6)$ *is NP–complete for chordal graphs.*

However, we have several results showing that special classes of chordal are easier:

Theorem 4 (a) *The problem* $KrauszDim$ *is polynomial for chordal graphs with bounded maximal clique size, and in particular for chordal graphs of bounded maximum degree.*
(b) *For any fixed* D*, the problem* $KrauszDim(D)$ *is polynomial for interval graphs.*

3 Graphs with maximum degree 4

In this section, we discuss the case of Krausz dimension being just by 1 smaller than the maximum degree. This question was suggested by K. Cechlárová [private communication]. Since $dim(\boldsymbol{G}) \leq \Delta(\boldsymbol{G})$ for any graph $\boldsymbol{G}$, this is the largest dimension (with respect to the maximum degree) for which nontrivial results may be expected.

Observation 3.1 The problem $KrauszDim(k, k+1)$ for a graph $\boldsymbol{G}$ is equivalent to the question whether there exists a collection of edge disjoint complete subgraphs of $\boldsymbol{G}$ of size at least 3 that cover each vertex of the maximum degree $k+1$.

In this sense we say that a graph is *CL*-coverable, if it contains a collection of edge disjoint complete subgraphs of size at least 3 covering all of its vertices. Then we can further reduce the considered problem:

Observation 3.2 Suppose that $\boldsymbol{G}$ contains a (not necessarily induced) subgraph $\boldsymbol{H}$ that is *CL*-coverable, and there is no triangle of $\boldsymbol{G}$ having just one or two edges in $\boldsymbol{H}$. Then, setting $\bar{\boldsymbol{G}} = (V(\boldsymbol{G}), E(\boldsymbol{G}) - E(\boldsymbol{H}))$, any Krausz order-$k$-partition of $\boldsymbol{G}$ is projected to an order-k-partition of $\bar{\boldsymbol{G}}$, and conversely, any order-k-partition of $\bar{\boldsymbol{G}}$ can be extended to whole $\boldsymbol{G}$ using the *CL*-cover of $\boldsymbol{H}$. Therefore $dim(\boldsymbol{G}) \leq k$ if and only if $dim(\bar{\boldsymbol{G}}) \leq k$.

Further we focus on the problem $KrauszDim(3,4)$.

Lemma 3.3 *To find a polynomial algorithm for* $KrauszDim(3,4)$, *it suffices to consider* K_4*-free graphs in which every triangle shares exactly one of its edges with other triangles.*

Proof: Let $\boldsymbol{G}$ be a graph satisfying the assumptions. If there is a 5-clique in $\boldsymbol{G}$, then $\boldsymbol{G} \cong \boldsymbol{K}_5$ (since the degrees are 4) and $dim(\boldsymbol{K}_5) = 1$.

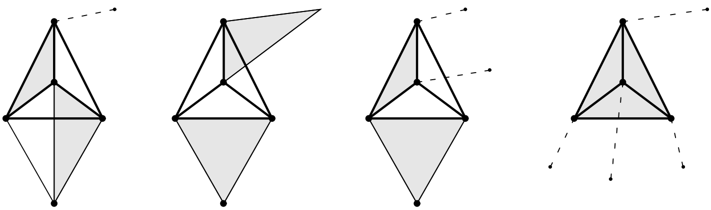

Figure 1: Possible types of neighbourhood of a 4-clique

A possible 4-clique $\boldsymbol{F}$ in $\boldsymbol{G}$ may be reduced as follows: Note that $\boldsymbol{F}$ has degrees 3, so each vertex of $\boldsymbol{F}$ is incident to at most one other edge. Discussing the positions of end vertices of these edges, there are only four possible configurations (except the 5-clique), depicted in Figure 1 (some of the dashed edges may be missing). Any of the cases leads to a subgraph in $\boldsymbol{G}$ that can be reduced using Observation 3.2—see the above picture where *CL*-covers are shaded.

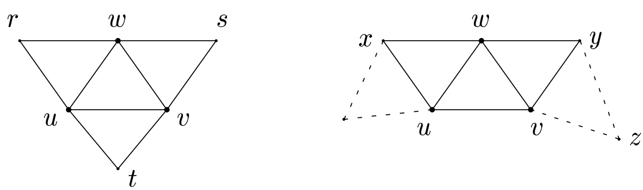

Figure 2: Neighbourhood of a triangle uvw

So from now on, our graph $\boldsymbol{G}$ has maximum clique size 3. If some of its triangles is disjoint with all other triangles in $\boldsymbol{G}$, it can be simply reduced. Otherwise, let $T = uvw$ be a triangle in $\boldsymbol{G}$ that shares more than one edge with other triangles, as in Figure 2. The situation when T shares all its edges with other triangles is shown in the left ($r \neq s \neq t$ since there is no 4-clique). Respecting that the degrees of u, v, w are 4, there can be no other triangle sharing any of the edges in the picture. Thus we reduce by Observation 3.2.

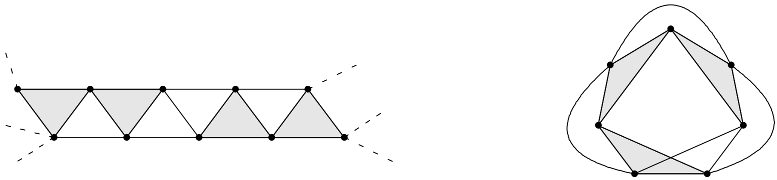

Figure 3: A chain and a closed chain of triangles

A bit more complicated situation arises when precisely two edges of T are shared with other triangles, see Figure 2 right. In this situation, there may be other triangles using the edges xu or vy, such as the triangle vyz. Clearly $z \neq u, z \neq x$ hold and zu is not an edge, since that would lead to previous cases. Then we get the same situation for the triangle vyw, and we continue. Finally, we obtain either a "chain" or a "closed chain" of triangles, as shown in Figure 3. In both cases, for any length of the chain, we can choose a suitable collection of edge-disjoint triangles covering all vertices of the chain, and reduce via Observation 3.2 again.

□

Lemma 3.4 *The problem KrauszDim$(3,4)$, for a graph satisfying the conditions of Lemma 3.3, can be reduced to finding maximum matching in a bipartite graph.*

Proof: Let $F \subseteq E(\boldsymbol{G})$ be the set of all edges of $\boldsymbol{G}$ that are contained in more than one triangle, and let $U \subseteq V(\boldsymbol{G})$ be the set of all vertices of $\boldsymbol{G}$ that have degree 4 and are not incident with any of the edges from F. The bipartite graph $\boldsymbol{B}$ is defined on the vertex set $F \cup U$, with edges of the form fu where $f = \{s,t\} \in F$, $u \in U$ and $\{u,s,t\}$ forming a triangle in $\boldsymbol{G}$.

We claim that $dim(\boldsymbol{G}) \leq 3$ if and only if $\boldsymbol{B}$ has a matching of size $|U|$: In one direction, having a Krausz order-3-partition of $\boldsymbol{G}$, each vertex $u \in U$ must be in some triangle T of the partition. Since every triangle of $\boldsymbol{G}$ shares one edge with other triangles, T contains one edge from F.

Thus for each $u \in U$ there is an edge $\{f, u\} \in E(\boldsymbol{B})$, $f = \{s, t\} \in F$, corresponding to a triangle-cluster of the partition. These edges (one for each u) form a matching since the triangles in the partition are edge-disjoint.

For the opposite direction, suppose there exists a matching M in $\boldsymbol{B}$ covering each vertex of U, and let $f_u \in F$ be the edge matched to u, $u \in U$. Denote $\bar{F} = \{f_u | u \in U\}$ and $F' = F \setminus \bar{F}$. For every $f \in F'$, pick a triangle T_f containing the edge f. Note that since every triangle shares only one edge with other traingles, every triangle contains exactly one edge of F. Therefore the triangles $T_f, f \in F'$ and $\{u\} \cup f_u, u \in U$ are pairwise edge-disjoint and cover all vertices of degree 4. The desired Krausz partition of $\boldsymbol{G}$ consists of these triangles plus the remaining edges of $\boldsymbol{G}$ as single clusters.

□

Proof of Theorem 2 a): A combination of the previous results gives a simple polynomial algorithm for testing $\mathit{KrauszDim}(3,4)$, that uses a subroutine computing maximum matching in a bipartite graph (which is well known to be polynomial). To be accurate, we present a scheme of the whole algorithm here:

Algorithm 3.5 problem $\mathit{KrauszDim}(3,4)$
begin
  input a connected graph $\boldsymbol{G} = (V, E)$ with $\Delta(\boldsymbol{G}) \leq 4$
  if $\boldsymbol{G} \cong \boldsymbol{K}_5$ **then** **output**("$\mathit{dim}(\boldsymbol{G}) = 1$") **fi**
  for every 4-tuple $X \subset V(\boldsymbol{G})$ **do**
    if $\boldsymbol{G}|_X \cong \boldsymbol{K}_4$ **then**
      examine the neighbours of X,
      find out which of the cases from Fig. 1 occurs,
      delete the edges found above from $\boldsymbol{G}$
    fi
  done
  for every triple $X \subset V(\boldsymbol{G})$ **do**
    if $\boldsymbol{G}|_X \cong \boldsymbol{K}_3$ **then**
      if the situation from Fig. 2 left occurs **then**
        delete the edges of the subgraph from $\boldsymbol{G}$
      else if the situation from Fig. 2 right occurs **then**
        go through the chain of edge-neighbouring triangles
        in both directions,
        delete the edges of the whole chain of triangles from $\boldsymbol{G}$
      fi
    fi

done
construct the graph $\boldsymbol{B}$ on $U \cup F$ as defined in the proof of Lemma 3.4
call *MatchingInBipartiteGraph*($\boldsymbol{B}$)
if matching of size $|U|$ exists **then** **output**("$dim(\boldsymbol{G}) \leq 3$")
else **output**("$dim(\boldsymbol{G}) > 3$")
end.

The running time of this algorithm is $O(n^4)$. This might be improved by more careful analysis of the reduction steps.

□

4 Graphs of bounded treewidth

We start with some essential definitions and considerations. A *tree decomposition* of a graph $\boldsymbol{G}$ is a pair $(\mathcal{S}, T)$ where $\mathcal{S} = \{X_i : i \in I\}$ is a collection of subsets of vertices of $\boldsymbol{G}$, and T is a tree on the vertex set $V(T) = I$ (one node for each element of $\mathcal{S}$), satisfying the following conditions:

1. $\bigcup_{i \in I} X_i = V(\boldsymbol{G})$,
2. for every edge $\{u, v\} \in E(\boldsymbol{G})$ there is an $i \in I$ such that $u, v \in X_i$,
3. for each vertex $v \in V(\boldsymbol{G})$, the set of nodes $\{i : v \in X_i\}$ forms a subtree of T.

The *width* of the tree decomposition is defined as $\max_{i \in I}(|X_i| - 1)$, and the *treewidth of a graph* is the minimum width over all of its tree decompositions.

We may suppose that the tree T of the decomposition is a rooted binary tree—we just choose some root, duplicate nodes of high degrees (making a binary subtrees on them), and possibly add missing leaves.

In the next text the following notation is used: A tree decomposition is allways taken in the form presented above, with the same notation of the tree and the subsets. The set of all descendants of a node i is denoted by $\downarrow i$, and $\downarrow X_i$ stands for the union $\bigcup_{j \in \downarrow i} X_j$. A restriction of a Krausz partition $\mathcal{K}$ of a graph $\boldsymbol{G}$ onto a subset $X \subseteq V(\boldsymbol{G})$, i.e. a collection of clusters of $\mathcal{K}$ restricted to the vertex set X, is written as $\mathcal{K}|_X$.

The idea of our solution to *KrauszDim* problem for bounded treewidth graphs is rather simple (see also Figure 4)—we generate all possible Krausz partitions on all subgraphs of $\boldsymbol{G}$ corresponding to leaves of T in the tree decomposition, and then we process all their consistent combinations dynamically from leaves to the root of T, finally finding the lowest dimension used in the root node. Of course, it is impossible to maintain the whole partitioning during the dynamic process, so only a

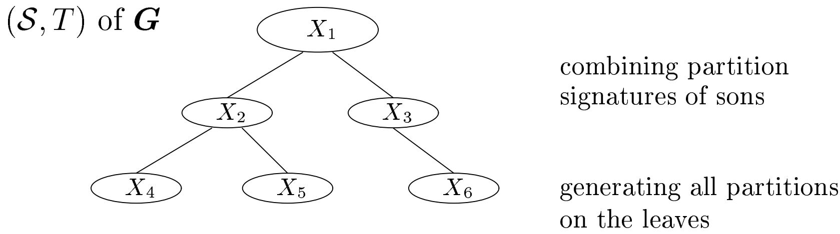

Figure 4: A tree decomposition $(\mathcal{S}, T)$ of $\boldsymbol{G}$ and a scheme of the algorithm

"partition signature" of a node (that describes everything relevant to the ancestors of this node) is carried. If the treewidth of $\boldsymbol{G}$ is bounded by some constant w, then it costs polynomial (though rather large in w) time to process every node with all possible signatures.

Theorem 5 *For any w, the problem KrauszDim is solvable in polynomial time for graphs of bounded treewidth.*

Proof: Let $\boldsymbol{G}$ be a graph with treewidth bounded by w and let $(\mathcal{S}, T)$ be a corresponding tree decomposition (that can be found in linear time for a constant w, see [5]). Based on that, we present an algorithm that determines the Krausz dimension of $\boldsymbol{G}$ in time polynomial in the size of $\boldsymbol{G}$ for a constant w.

A *partition signature* in a node i of the tree decomposition $(\mathcal{S}, T)$ of $\boldsymbol{G}$ is defined as a 4-tuple $\mathit{Sig}(i) = (\mathcal{P}, q, o, m)$, where $\mathcal{P} = \{C_1, \ldots, C_p\}$ is a partition of the edges of $\boldsymbol{G}|_{X_i}$ into complete subgraphs, $q : \mathcal{P} \to \{0, 1\}$ are labels of these subgraphs, $o : X_i \to \{0, \ldots, \Delta\}$ are values of the vertices (here $\Delta = \Delta(\boldsymbol{G})$), and m is a number. For every node i, we construct the set $\mathit{AllSig}(i)$ of admissible partition signatures in i in the following sense: $(\mathcal{P}, q, o, m) \in \mathit{AllSig}(i)$ if and only if there exists a Krausz partition $\mathcal{K}$ of $\boldsymbol{G}|_{\downarrow X_i}$ of order m, such that $\mathcal{K}|_{X_i} = \mathcal{P}$, the order of each $v \in X_i$ is $o(v)$, and for every $C \in \mathcal{P}$, $q(C) = 1$ iff $C \in \mathcal{K}$. The aim of the attribute $q(C)$ is to indicate that the complete subgraph C is alone a cluster, and not a part of a larger cluster in $\mathcal{K}$, so it can be extended to a larger cluster in ancestor nodes.

To better understand this, realize that the properties of the tree decomposition imply $\downarrow X_r \cap \downarrow X_s \subseteq X_n$, $X_n \cap \downarrow X_r \subseteq X_r$, $X_n \cap \downarrow X_s \subseteq X_s$ for a node n and its two sons r, s of the decomposition, and $\boldsymbol{G}|_{\downarrow X_n} = \bigcup_{j \in \downarrow n} \boldsymbol{G}|_{X_j}$.

A simple computation shows that there are at most $2^{2^{w+1}} \cdot 2^{(w+1)^2} \cdot (\Delta+1)^{w+1} \cdot (\Delta+1)$ distinct signatures in one node, which is polynomial in Δ. So all of them can be efficiently generated in a leaf of T (in fact, in constant time for one leaf). One can also combine in polynomial time all pairs of signatures of the two sons of an inner node, thus generating all admissible signatures of that node (testing one triple of signatures is also in constant time), even if the computation is just "by brute force". Then dynamic processing from leaves to the root of the tree T gives all admissible signatures in the root node r. Among them, a signature with lowest m_r determines the Krausz dimension of $\boldsymbol{G}$.

Now it is enough to show how the signatures of sons are combined in a node of the decomposition. Let n be an inner node of T with two sons r, s, and $Sig(r) = \big(\mathcal{P}_r, q_r, o_r, m_r\big)$, $Sig(s) = \big(\mathcal{P}_s, q_s, o_s, m_s\big)$ be admissible (in the sense experessed above) signatures in the nodes r, s. If $Sig(n) = \big(\mathcal{P}_n, q_n, o_n, m_n\big)$ is *any* signature in the node n, it is an admissible combination of $Sig(r), Sig(s)$ if and only if the following conditions are satisfied:

- *(mutual consistency of $Sig(r), Sig(s)$)*
  If $C \in \mathcal{P}_r, C' \in \mathcal{P}_s$ and $|C \cap C'| > 1$, then either $C = C'$ or $C = C' \cap X_r$, $q_r(C) = 1$ or $C' = C \cap X_s$, $q_s(C') = 1$.
- *(correct clusters in node n)*
  For each $C \in \mathcal{P}_n$, one of $|C \cap X_r| \leq 1$ or $C \cap X_r = C_0 \in \mathcal{P}_r$, $q_r(C_0) = 1$ or $C = C_0 \cap X_n$, $C_0 \in \mathcal{P}_r$ should hold.
- *(correct cluster labels in n)*
  For each $C \in \mathcal{P}_n$, the label is $q_n(C) = 0$ if and only if $C \subseteq X_r \cup X_s$, $\max\{|C \cap X_r|, |C \cap X_s|\} > 1$, and the following is satisfied: For $|C \cap X_r| > 1$, either $C \cap X_r \notin \mathcal{P}_r$ or $q_r(C \cap X_r) = 0$; and similarly for s.
- *(orders of vertices of X_n)*
  For each $v \in X_n$, the order of v is $o_n(v) = \big|\{C \in \mathcal{P}_n;\, v \in C\}\big|$ for $v \notin X_r \cup X_s$, $o_n(v) = o_r(v) + \big|\{C \in \mathcal{P}_n;\, v \in C, |C \cap X_r| \leq 1\}\big|$ for $v \in X_r - X_s$, $o_n(v) = o_s(v) + \big|\{C \in \mathcal{P}_n;\, v \in C, |C \cap X_s| \leq 1\}\big|$ for $v \in X_s - X_r$, and $o_n(v) = o_r(v) + o_s(v) - \big|\{C \in \mathcal{P}_r;\, v \in C, |C \cap X_r \cap X_s| > 1\}\big| + \big|\{C \in \mathcal{P}_n;\, v \in C, |C \cap (X_r \cup X_s)| \leq 1\}\big|$ for $v \in X_r \cap X_s$.
  The maximal order is $m_n = \max(o_n[X_n] \cup \{m_r, m_s\})$.

Finally, we summarize the above ideas in a scheme of the algorithm:

Algorithm 4.1 problem *KrauszDim* for graphs of constant treewidth
begin

```
    input  graph G = (V, E)
    suppose G connected, treewidth of G bounded by w
    call  TreeDecomposition(G)  → (S, T)
    for  l ∈ V(T), l leaf of T  do
      AllSig(l) := a collection of all admissible signatures Sig(l)
            (derived from all Krausz partitions of X_l)
    done
    while  exist n ∈ V(T) not processed yet  do
      n := lowest node in V(T) not processed
      r, s := two sons of n (already processed)
      AllSig(n) := ∅
      for  Sig(n) in all signatures possible on X_n  do
        for  [Sig(r), Sig(s)] ∈ AllSig(r) × AllSig(s)  do
          check Sig(n) against Sig(r), Sig(s), as described above
          if  Sig(n) consistent with Sig(r), Sig(s)  then
            AllSig(n) := AllSig(n) ∪ Sig(n)
          fi
        done
      done
    done
    r = root of T
    D := min{m_r | Sig(r) ∈ AllSig(r)}
    output("dim(G) = D")
end.
```

□

Note that our result also applies to chordal graphs with bounded maximum clique size, since the treewidth of a chordal graph $\boldsymbol{G}$ is exactly $\omega(\boldsymbol{G}) - 1$:

Corollary 4.2 *The problem KrauszDim is polynomially solvable for chordal graphs of bounded clique size.*

Since $\Delta(\boldsymbol{G}) \geq \omega(\boldsymbol{G}) - 1$, we have straightforwardly (compare to Theorem 2.(b)):

Corollary 4.3 *The problem KrauszDim is polynomially solvable for chordal graphs of bounded maximum degree.*

The following lemma is useful for several classes of chordal graphs.

Lemma 4.4 *A clique of size at least $D^2 - D + 2$ must be a cluster in any Krausz order-D-partition of a graph.*

Proof: Let $\mathcal{C}$ be a clique of size ω in $\boldsymbol{G}$ that is not a cluster in a Krausz order-D-partition. If s is the size of the largest cluster used to cover $\mathcal{C}$, then $\frac{\omega-1}{s-1} \leq D$ since there are at least $\frac{\omega-1}{s-1}$ clusters covering a fixed vertex $v \in \mathcal{C}$ and its $\omega - 1$ neighbours. On the other hand, $s \leq D$ because for a cluster $A \subset \mathcal{C}$, $|A| = s$ and a vertex $v \in \mathcal{C} - A$, all the s edges $\{v, a\}$, $a \in A$ must be in different clusters.

Combining the previous inequalities, we get $\omega - 1 \leq Ds - D \leq D^2 - D$, thus the largest clique that need not be a cluster has size at most $D^2 - D + 1$.

$\square$

Split graphs are a special kind of chordal graphs, whose vertices can be split into a clique and an independent set.

Corollary 4.5 *For any fixed D, KrauszDim(D) is polynomial in the class of split graphs.*

Proof: Let $\boldsymbol{G}$ be a split graph and $\omega(\boldsymbol{G}) = \omega$. If $\omega \geq D^2 - D + 2$, then this maximal clique must be a cluster of the partition, and the rest is a bipartite graph which has the only partition into single edges. Otherwise, the problem is reduced to the case of treewidth bounded by a constant $\omega - 1 \leq D^2 - D$.

$\square$

A similar result may be derived for complements of bipartite graphs. A more involved corollary of the previous results is a polynomial algorithm solving *KrauszDim*(D) for interval graphs and constant D (interval graphs are those that admit intersection representations by closed intervals on a line).

Proof of Theorem 4 (b): Suppose we are given an interval graph $\boldsymbol{G}$ decomposed into a sequence of cliques $Q_1, \ldots, Q_t$. Such a decomposition can be easily derived from an interval representation, or it can be viewed as a path decomposition of $\boldsymbol{G}$ (a tree decomposition where the tree is a path).

If there is an i such that $|Q_i \cap Q_{i+1}| \geq D^2 - D + 1$, then $dim(\boldsymbol{G}) > D$ by Lemma 4.4 since $Q_i - Q_{i+1} \neq \emptyset$, $Q_{i+1} - Q_i \neq \emptyset$. Otherwise set $Q_i^0 = Q_i \cap \bigcup_{j \neq i} Q_j = Q_i \cap (Q_{i-1} \cup Q_{i+1})$ for every i. It follows that $|Q_i^0| \leq 2(D^2 - D)$. Thus if $|Q_k| > 2(D^2 - D)$ for some k, we can choose a subset $\bar{Q}_k$ of $2(D^2 - D)$ vertices such that $Q_k^0 \subseteq \bar{Q}_k \subset Q_k$.

In that case, the clique $\bar{Q}_k$ is a cluster in any order-D-partition of the graph $\boldsymbol{G}' = \bar{Q}_k \cup \bigcup_{j \neq k} Q_j$; and since the vertices of $(Q_k - \bar{Q}_k)$ are disjoint with all other cliques in the graph, this partition can be extended using

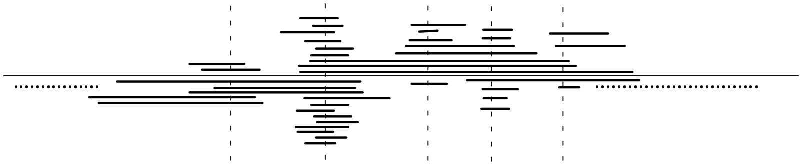

Figure 5: A decomposition of an interval graph into a sequence of cliques

the cluster Q_k to an order-D-partition of $\boldsymbol{G}$. That means $dim(\boldsymbol{G}) \leq D$ iff $dim(\boldsymbol{G}') \leq D$.

The above preprocessing either gives a negative answer, or produces an interval graph $\bar{\boldsymbol{G}}$ of maximal clique size $2(D^2 - D)$, hence of treewidth at most $2(D^2 - D) - 1$. This graph is then passed to Algorithm 4.1.

Algorithm 4.6 problem *KrauszDim*(D) for interval graphs and fixed D

begin
  input a connected interval graph $\boldsymbol{G} = (V, E)$
  call *IntervalRepresentation*$(\boldsymbol{G})$
  got representation $\{L_i = \langle l_i, r_i \rangle;\ i \in V\}$,
      where all endpoints l_i, r_i are distinct
  define $q_1 = \min\{r_i\}$, $R_{k+1} = \{r_i \mid \exists j : r_i > l_j > q_k\}$
    (R_t last nonempty), $q_{k+1} = \min R_{k+1}$ for $k + 1 \leq t$
  $Q_k := \{i \mid q_k \in L_i\}$, $k = 1, \ldots, t$
  for $i :=$ 1 to $t - 1$ **do**
    if $|Q_i \cap Q_{i+1}| \geq D^2 - D + 1$ **then** **output**("$dim(\boldsymbol{G}) > D$")
  done
  $U := V$
  for $i :=$ 1 to t **do**
    if $|Q_i| > 2(D^2 - D)$ **then**
      choose $X \subseteq Q_i - (Q_{i-1} \cup Q_{i+1})$, $|X| = |Q_i| - 2(D^2 - D)$
      $U := U - X$
    fi
  done
  $\bar{\boldsymbol{G}} := \boldsymbol{G}|_U$
  output($dim(\bar{\boldsymbol{G}})$ by Algorithm 4.1)
end.

□

5 NP reductions

For the *NP* reductions to the Krausz dimension problem, we use a special version of the well known satisfiability problem [3]. We consider a boolean formula Φ in the conjunctive normal form, with a set of clauses C over a set of variables V. By the *formula graph* we mean the bipartite graph $\boldsymbol{F}_\Phi$ on the vertex set $C \cup V$, and edges connecting each variable x to all clauses cntaining x or $\neg x$; formally $V(\boldsymbol{F}_\Phi) = C \cup V$, $E(\boldsymbol{F}_\Phi) = \big\{\{x, c\} \mid x \in V, c \in C,\ x \in c \vee \neg x \in c\big\}$. The *PLANAR* 3-*SAT* is defined as the satisfiability problem restricted to formulas with planar graphs of maximal degree 3.

The following lemma can be found in [2]:

Lemma 5.1 *The PLANAR* 3*-SAT problem is NP–complete.*

Now we show that the *PLANAR* 3-*SAT* can be reduced to the question, whether a given planar graph with degrees at most 5 has a Krausz dimension at most 3. We closely follow ideas used in [4] in the proof.

Proof of Theorem 2 (b): Given a formula Φ satisfying the conditions stated above, we construct a graph $\boldsymbol{R}_\Phi$ that has Krausz dimension at most 3 iff Φ is satisfiable. In the construction, every variable and every clause vertex of the graph $\boldsymbol{F}_\Phi$ is replaced by a special graph, see Figure 6. The variable graph $\mathcal{V}$ has two terminal vertices for its positive occurrences and two terminal vertices for negated occurrences, the clause graph $\mathcal{C}$ has three terminal vertices for its three literals. We may suppose that each variable has at most 2 positive and at most 2 negated occurrences; otherwise, if some variable has only positive (negated) occurrences, we may set it true (false) and reduce the formula.

Clauses are connected with their variables by identifying the corresponding two terminal vertices. For clauses that contain less than 3 variables, a special false terminator is used on the remaining terminals. Clearly, if the formula graph is planar, so is the constructed graph $\boldsymbol{R}_\Phi$; and also $\Delta(\boldsymbol{R}_\Phi) = 5$ is fulfilled.

To prove that the reduction is correct, see the all possible Krausz order-3-partitions depicted in Figure 7 (where clusters are the shaded triangles and the remaining edges). Note that the graphs do not contain 4-cliques, so a vertex of degree 5 must be in two triangles and a vertex of degree 4 in at least one triangle of the partition. Then focus on the central vertices in both graphs, their partitionings already determin the rest.

The two possible partitions of the variable graph $\mathcal{V}(x)$ encode the logical values *true* or *false* of the variable x, and the three partitions of

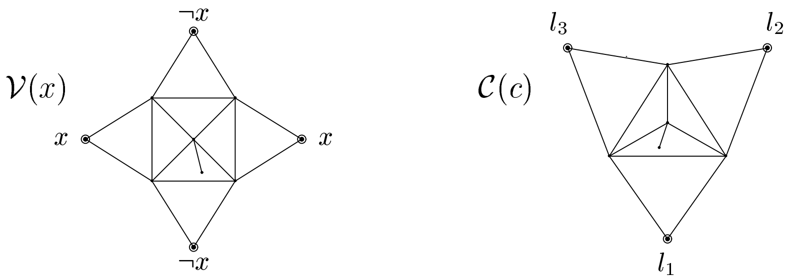

Figure 6: The variable graph $\mathcal{V}(x)$ and the clause graph $\mathcal{C}(c)$

the clause graph $\mathcal{C}(c)$ determine which of the three literals of c is chosen to satisfy the clause c. The false terminator is formed simply by adding two new vertices and two new edges connecting each of them to the terminal (i.e., adding two leaves).

Consider now a vertex t of the constructed graph $\boldsymbol{R}_\Phi$, that is a unified terminal vertex of $\mathcal{V}(x)$ and $\mathcal{C}(c)$ (Figure 8). In any Krausz order-3-partition, at least one of the triangles T_1, T_2 must be as a cluster. This implies, in the above presented interpretation, that either the literal of c represented by the terminal t is not the one that must be true in the clause c, or the occurence of the variable x in c is true (i.e., x or $\neg x$ depending on the terminal of $\mathcal{V}(x)$ used). So if $\boldsymbol{R}_\Phi$ has Krausz dimension 3, there exists an evaluation of variables in Φ such that every clause contains at least one true literal, and Φ is satisfiable. On the other hand, it is easy to construct, from a given satisfying assignment for Φ, a Krausz 3-partition of $\boldsymbol{R}_\Phi$.

□

Corollary 5.2 *For every $D \geq 3$, the problem KrauszDim$(D, D+2)$ is NP–complete, even when restricted to planar graphs.*

Proof: Pend $D-3$ vertices of degree one on each vertex of the input graph for *KrauszDim*$(3,5)$.

□

If we consider chordal graphs, the Krausz dimension is much easier problem, as was shown in the previous section. Neverthless, the problem even then remains hard if the maximal clique size (treewidth) is not

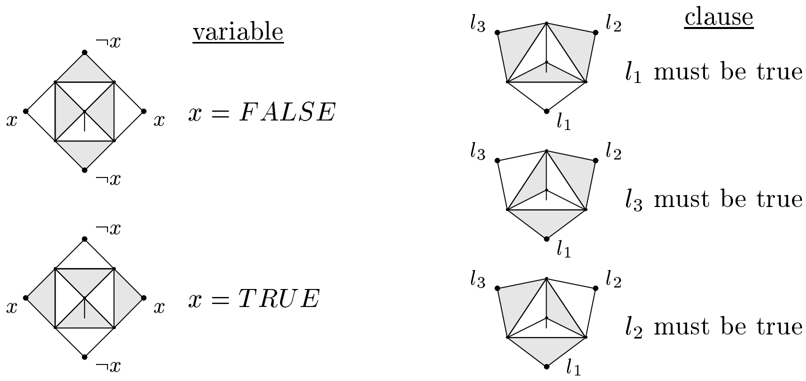

Figure 7: Possible partitions of variable and clause graphs

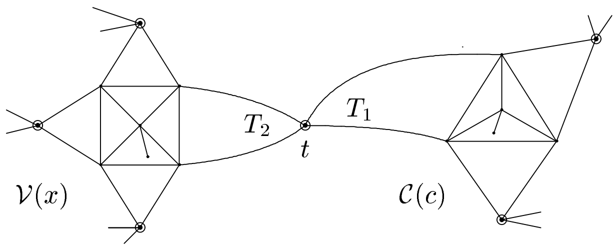

Figure 8: A terminal connection between variable and clause graphs

bounded. We show that the question whether a chordal graph $\boldsymbol{H}$ has Krausz dimension at most 6 is NP–complete.

Proof of Theorem 3: The proof is similar to the one of Theorem 2 (b).

The graph $\boldsymbol{R}^c_\Phi$ is constructed from the formula graph $\boldsymbol{F}_\Phi$ by replacing its vertices with variable and clause graphs $\mathcal{V}^c, \mathcal{C}^c$ from Figure 9, and by identifying corresponding terminals to represent its edges, similarly as above. Only variables with one negated and one or two positive occurences are considered (otherwise the variable is substituted with its negation). The false terminator is now formed by three leaves. Finally, a large clique Q containing all terminal vertices is added (to produce a chordal graph).

The key fact in the proof is that the clique Q must be a cluster in any order-6-partition of $\boldsymbol{R}^c_\Phi$ by Lemma 4.4, so we need not bother with it

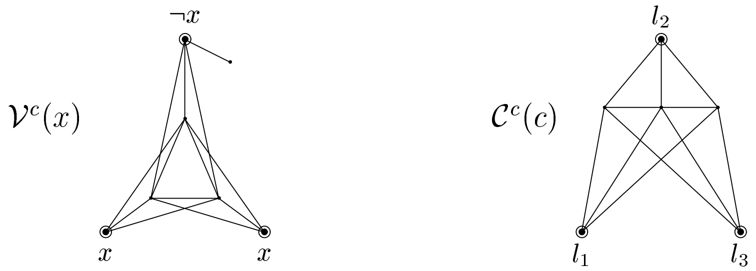

Figure 9: The variable and clause graphs for a chordal reduction

furthermore and the rest of the proof is similar to the previous one.

The reader can easily check the following facts about Krausz partitions of $\mathcal{V}^c$, $\mathcal{C}^c$:

- If a terminal vertex x of $\mathcal{V}^c(x)$ has order 2, then the terminal vertex $\neg x$ has order at least 3, and vice versa.
- At least one of the terminal vertices l_1, l_2, l_3 of $\mathcal{C}^c(c)$ has order more than 2.
- There exist partitions of $\mathcal{V}^c(x)$ such that the orders on terminals $x, \neg x, x$ are $2,3,2$ or $3,2,3$ respectively, and partitions of $\mathcal{C}^c(c)$ such that the orders on terminals l_1, l_2, l_3 are $2,2,3$ or $2,3,2$ or $3,2,2$ respectively.

These facts together imply: If Φ is satisfiable, then $dim(\boldsymbol{R}^c_\Phi) \leq 6$. Conversly, having a Krausz order-6-partition of $\boldsymbol{R}^c_\Phi$, each of its clause subgraphs has at least one terminal of order 3 (on the clause side), so the adjacent variable subgraph has order 2 on this terminal. That expresses the value true of this literal, and such evaluation is consistent over the whole formula.

$\square$

6 Open problems

We have shown that for every fixed $D \geq 3$, deciding whether $dim(\boldsymbol{G}) \leq D$ is NP-complete for graphs of maximum degree $D + 2$, while deciding $dim(\boldsymbol{G}) \geq 3$ is polynomial for graphs of maximum degree 4. Thus we have partially answered a question first raised by Cechlárová [private communication]. The general question remains open:

Problem 1. Decide the complexity of $\mathit{KrauszDim}(D, D+1)$ for $D > 3$.

We have shown that for any fixed D, the problem $\mathit{KrauszDim}(D)$ is solvable in polynomial time for split graphs and for complements of bipartite graphs. The following is, however, left open:

Problem 2. Decide the complexity of *KrauszDim* for split graphs and for complements of bipartite graphs.

Quite intriguing seems the question of Krausz dimension of general chordal graphs. We have proved that $\mathit{KrauszDim}(D)$ restricted to chordal graphs is NP-complete for $D \geq 6$, and of course this problem is polynomial for $D = 2$. The gap between 2 and 6 is open:

Problem 3. Decide the complexity of $\mathit{KrauszDim}(D)$ restricted to chordal graphs for $D = 3, 4, 5$.

References

[1] L.W. Beineke, I. Broere, *The Krausz dimension of a graph*, preprint 1994.

[2] M. Fellows, J. Kratochvíl, M. Middendorf, F. Pfeiffer, *The complexity of induced minors and related problems*, Algorithmica 13 (1995), 266–282.

[3] M.R. Garey, D.S. Johnson, Computers and Intractability, W.H. Freeman and Company, New York 1979.

[4] P. Hliněný, *Classes and recognition of curve contact graphs*, submitted to J. of Combinatorial Theory, 1996.

[5] T. Kloks, Treewidth, computations and approximations, Lecture Notes in Computer Science, Springer-Verlag 1994.

[6] J. Krausz, *Démonstration nouvelle d'un théorème de Whitney sur les résaux* (Hungarian with French summary), Mat. Fiz. Lapok 50 (1943), 75–85.