

# Exact algorithms for Intervalizing Colored Graphs

*Hans L. Bodlaender*

*Johan M. M. van Rooij*

Technical Report UU-CS-2010-024  
October 2010

Department of Information and Computing Sciences  
Utrecht University, Utrecht, The Netherlands  
[www.cs.uu.nl](http://www.cs.uu.nl)

ISSN: 0924-3275

Department of Information and Computing Sciences  
Utrecht University  
P.O. Box 80.089  
3508 TB Utrecht  
The Netherlands

# Exact algorithms for Intervalizing Colored Graphs

Hans L. Bodlaender and Johan M. M. van Rooij

Department of Information and Computing Sciences

Utrecht University, P.O. Box 80.089, 35089TB, the Netherlands

## Abstract

In the INTERVALIZING COLORED GRAPHS problem, one must decide for a given graph  $G = (V, E)$  with a proper vertex coloring of  $G$  whether  $G$  is the subgraph of a properly colored interval graph. For the case that the number of colors  $k$  is fixed, we give an exact algorithm that uses  $\mathcal{O}^*(2^{n/\log^{1-\epsilon}(n)})$  time for all  $\epsilon > 0$ . We also give an  $\mathcal{O}^*(2^n)$  algorithm for the case that the number of colors  $k$  is not fixed.

## 1 Introduction

In this paper, we consider exact algorithms for the INTERVALIZING COLORED GRAPHS problem. This problem is defined in the following way. Given a graph  $G = (V, E)$  together with a proper vertex coloring  $c : V \rightarrow \{1, \dots, k\}$  of  $G$  (a coloring  $c$  is proper if for all edges  $\{v, w\} \in E$ :  $c(v) \neq c(w)$ ), one must decide if  $G$  is subgraph of a properly colored interval graph, i.e., can we add edges, such that each edge is between vertices of different colors and the result is an interval graph? The problem has its original motivation in DNA physical mapping [13]

This problem is NP-complete [13] (see also [16]), even when the number of colors  $k$  equals 4 [5, 6], and in addition, inputs are restricted to caterpillar trees [1]. We denote the version of the problem where the number of colors  $k$  is fixed by INTERVALIZING  $k$ -COLORED GRAPHS, and the version with a potentially unbounded number of colors by INTERVALIZING COLORED GRAPHS.

If the number of colors  $k = 2$ , the problem is trivially solvable in linear time. For three colors, the problem is solvable in quadratic time with a complicated algorithm [7]; the case for three colors and biconnected graphs is described in [6].

Our first algorithm deals with the case that the number of colors is a constant that is at least four. We give an algorithm that solves this version exactly, using slightly less than exponential time.

Most NP-hard problems that have subexponential algorithms deal with planar graphs and generalizations of planar graphs, see e.g., [12, 14, 21]. Typically, the running time of such algorithms is of the form  $\mathcal{O}^*(2^{\mathcal{O}(\sqrt{n})})$ . The result of our paper is a curious exception to the general pattern, both as inputs are general graphs (but a positive answer implies

bounded pathwidth of the input), and as the running time is ‘just subexponential’: for every  $\epsilon > 0$ , the running time is  $\mathcal{O}^*(2^{n/(\log^{1-\epsilon}(n))})$ .

Our algorithm for INTERVALIZING  $k$ -COLORED GRAPHS can be viewed as a dynamic programming algorithm in Held-Karp style [19], resembling algorithms for some graph layout problems given e.g., in [10], with one additional improvement: an isomorphism step for certain parts of the graph during the dynamic programming. Important concepts that facilitate the presentation of our results are the notions of *path decomposition* and *nice path decomposition*. Our  $\mathcal{O}^*(2^n)$  time algorithm for INTERVALIZING COLORED GRAPHS is a simple dynamic programming algorithm, also in Held-Karp style.

## 2 Preliminaries

In this section, we introduce some standard notations, and give a few preliminary results on path decompositions.

The graphs in this paper are considered to be undirected and simple. If not stated otherwise, the graphs we consider are *labeled* graphs, i.e., two isomorphic graphs with different labels are considered to be different. We also considered *unlabeled* graphs: two isomorphic unlabeled graphs are considered to be the same object. The number of vertices of graph  $G = (V, E)$  is denoted by  $n$ .

For a graph  $G = (V, E)$  and a set of vertices  $W \subseteq V$ , we denote  $G[W]$  as the subgraph induced by  $W$ :  $G[W] = (W, \{\{v, w\} \in E \mid v, w \in W\})$ .

**Definition 1** A graph  $G = (V, E)$  is an interval graph if we can associate to each vertex  $v \in V$  an interval on the real line  $I_v = [\ell_v, r_v]$  such that, for all  $v, w \in V$ ,  $v \neq w$ :  $\{v, w\} \in E$ , if and only if,  $I_v \cap I_w \neq \emptyset$ .

**Definition 2** A graph  $H = (V, F)$  is an interval completion of a graph  $G = (V, E)$  if  $G$  and  $H$  have the same vertex set,  $E \subseteq F$ , and  $H$  is an interval graph. More background can be found in [15]; see also [18].

**Definition 3** A path decomposition of a graph  $G = (V, E)$  is a sequence of subsets of  $V$  called bags,  $(X_1, X_2, \dots, X_r)$  such that:

- $\bigcup_{1 \leq i \leq r} X_i = V$
- for all  $\{v, w\} \in E$ : there is an  $i$ ,  $v, w \in X_i$
- for all  $i_0, i_1, i_2$ :  $1 \leq i_0 \leq i_1 \leq i_2 \leq r$ :  $X_{i_1} \subseteq (X_{i_0} \cap X_{i_2})$

**Definition 4** The width of a path decomposition  $(X_1, X_2, \dots, X_r)$  is  $\max_{1 \leq i \leq r} |X_i| - 1$ . The pathwidth of a graph  $G$  is the minimum width of a path decomposition of  $G$ .

**Definition 5** A path decomposition  $(X_1, X_2, \dots, X_r)$  is nice, if for all  $i$ ,  $1 \leq i < r$ , one of the following two cases holds:

- There is a vertex  $v \in V$  with  $X_{i+1} = X_i \cup \{v\}$ . We call  $X_{i+1}$  an introduce node.
- There is a vertex  $v \in V$  with  $X_{i+1} = X_i - \{v\}$ . We call  $X_{i+1}$  a forget node.

If  $|X_1| = 1$ , we also call 1 an introduce node. The following proposition is well known. We give the proof for later reference.

**Proposition 6 (Folklore)** *Each graph  $G = (V, E)$  with pathwidth  $k$  has a nice path decomposition of width  $k$  with  $2n$  bags, with  $|X_1| = 1$ , and  $X_r = \emptyset$ .*

**Proof:** Suppose we have a path decomposition  $(X_1, X_2, \dots, X_r)$ . We can turn it in a nice path decomposition as follows. First, remove all bags that are empty. If for some  $i$ ,  $1 \leq i < r$ ,  $i + 1$  is not an introduce or forget bag, then we insert some new bags between  $i$  and  $i + 1$ : first forget nodes, one for each vertex in  $X_i - X_{i+1}$ , and then we have one introduce node for each vertex in  $X_{i+1} - X_i$ . Similarly, we add introduce nodes before  $X_1$  when  $|X_1| \neq 1$ , and add forget nodes at the end of the procedure till  $X_r = \emptyset$ . We have one introduce and one forget node per vertex, so we have  $2n$  bags.  $\square$

**Proposition 7** *There are at most  $(k + 2^k + 1)^{2n-1}$  unlabeled graphs with pathwidth at most  $k$  that are pairwise non isomorphic.*

**Proof:** Consider a nice path decomposition of a graph with  $n$  vertices, with  $|X_1| = 1$ , and with  $2n$  bags. For each of the bags  $X_i$ ,  $i > 1$ , there are at most  $k + 2^k + 1$  possibilities: we can have a forget node, where we have the choice which of the at most  $k + 1$  vertices in  $X_i$  we forget, or we can have an introduce node, where we have the choice to which of the at most  $k$  vertices in  $X_i$  the introduced vertex has an edge, i.e., at most  $2^k$  choices for an introduce node. If we have two graphs with two path decompositions that we can construct while always making the same choices, then these graphs are isomorphic.  $\square$

**Proposition 8** *Let  $G = (V, E)$  be a graph with proper vertex coloring  $c : V \rightarrow \{1, 2, \dots, k\}$ . The following are equivalent.*

1.  $G$  has a properly colored interval completion.
2.  $G$  has a path decomposition  $(X_1, X_2, \dots, X_r)$  such that for all  $v, w \in V$ : if  $v \neq w$  and there is an  $i$  with  $v, w \in X_i$ , then  $c(v) \neq c(w)$
3.  $G$  has a nice path decomposition  $(X_1, X_2, \dots, X_{2|V|})$  of width at most  $k - 1$  such that for all  $v, w \in V$ : if  $v \neq w$  and there is an  $i$  with  $v, w \in X_i$ , then  $c(v) \neq c(w)$

This proposition is also well known. Given a (nice) path decomposition  $(X_1, X_2, \dots, X_r)$  from Proposition 8 (ii) or (iii), one obtains the corresponding interval graph by making each  $X_i$  a clique. The corresponding interval graph model is obtained by taking for a vertex  $v$  the interval  $[\min_{v \in X_i} i, \max_{v \in X_i} i]$ . As all colors in a bag  $X_i$  are different, the width of the path decompositions is bounded by  $k - 1$ .

Proposition 8 motivates the definition of a *properly colored path decomposition*:  $(X_1, \dots, X_r)$  is a properly colored path decomposition of  $G$ , if and only if it is a path decomposition of  $G$ , and for all  $v, w \in V$ , if  $v \neq w$  and there is an  $i$  with  $v, w \in X_i$ , then  $c(v) \neq c(w)$ .

### 3 Partial path decompositions

In this section, we introduce a number of notions that will be used for our dynamic programming algorithm in the next section.

**Definition 9** A partial path decomposition of a graph  $G = (V, E)$  is a sequence of subsets of  $V$   $(X_1, X_2, \dots, X_s)$  such that:

- $(X_1, X_2, \dots, X_s)$  is a path decomposition of  $G[\bigcup_{1 \leq i \leq s} X_i]$
- For each connected component of  $G[V - X_s]$  with vertex set  $W$ , either  $W \subseteq \bigcup_{1 \leq i \leq s-1} X_i$  or  $W \cap (\bigcup_{1 \leq i \leq s-1} X_i) = \emptyset$ .

The following proposition follows from well known facts about path and tree decompositions.

**Proposition 10** Let  $(X_1, X_2, \dots, X_r)$  be a path decomposition of  $G$ . Then, for each  $s$ ,  $1 \leq s \leq r$ ,  $(X_1, X_2, \dots, X_s)$  is a partial path decomposition of  $G$ .

Consider a partial path decomposition  $(X_1, X_2, \dots, X_r)$  and a vertex set  $X$ . Later,  $X$  will typically be the set  $X_r$  for some partial path decomposition  $(X_1, X_2, \dots, X_r)$ . A *component* of  $X$  is a vertex set that forms a connected component of the graph  $G[V - X_r]$ .

**Definition 11** Two components  $Y$  and  $Z$  of  $X$  are isomorphic components of  $X$ , if there is a graph isomorphism  $f$  of  $G[Y \cup X]$  to  $G[Z \cup X]$  that preserves colors and is the identity when restricted to  $X$ , i.e.,  $f$  is a bijective function, such that:

1. for all  $v, w \in Y \cup X$ :  $\{v, w\} \in E \Leftrightarrow \{f(v), f(w)\} \in E$
2. for all  $v \in Y \cup X$ :  $c(v) = c(f(v))$
3. for all  $v \in X$ :  $f(v) = v$ .

**Definition 12** A component  $W$  of  $X_r$  is said to be a left component of the partial path decomposition  $(X_1, \dots, X_r)$ , if  $W \subseteq \bigcup_{1 \leq i \leq r-1} X_i$ , and a right component of  $(X_1, \dots, X_r)$ , if  $W \cap (\bigcup_{1 \leq i \leq r-1} X_i) = \emptyset$ .

The following proposition follows directly from the definitions and well known facts on path decompositions.

**Proposition 13** *Let  $(X_1, X_2, \dots, X_r)$  be a partial path decomposition of  $G$ . Each component of  $X_r$  is either a left or a right component of  $(X_1, X_2, \dots, X_r)$ .*

**Definition 14** *A partial path decomposition  $(X_1, X_2, \dots, X_s)$  of  $G = (V, E)$  is properly colored, if for all  $v, w \in V$ , if  $v \neq w$  and there exists an  $i$  with  $v, w \in X_i$ , then  $c(v) \neq c(w)$ .*

**Definition 15** *A (partial) path decomposition  $(Y_1, Y_2, \dots, Y_r)$  is an extension of a partial path decomposition  $(X_1, X_2, \dots, X_s)$  if  $r \leq s$  and for all  $i$ ,  $1 \leq i \leq r$ ,  $Y_i = X_i$ .*

**Definition 16** *We define an equivalence relation on partial path decompositions as follows. We say that the partial path decomposition  $(X_1, X_2, \dots, X_r)$  is equivalent to the partial path decomposition  $(Y_1, Y_2, \dots, Y_s)$ , if the following two conditions hold:*

1.  $X_r = Y_s$ .
2. *Suppose  $W_1, W_2, \dots, W_q$  are the components of  $X_r$ . There is a bijective function  $g : \{1, \dots, q\} \rightarrow \{1, \dots, q\}$ , such that for all  $i$ ,  $1 \leq i \leq q$ :  $W_i$  is a left component of  $(X_1, X_2, \dots, X_r)$ , if and only if  $W_{g(i)}$  is a left component of  $(Y_1, Y_2, \dots, Y_s)$  and  $W_i$  and  $W_{g(i)}$  are isomorphic.*

The main insight behind our dynamic programming algorithm is the following result.

**Proposition 17** *If  $(X_1, X_2, \dots, X_r)$  and  $(Y_1, Y_2, \dots, Y_s)$  are equivalent colored partial path decompositions, then  $(X_1, X_2, \dots, X_r)$  has an extension that is a properly colored path decomposition of  $G$ , if and only if,  $(Y_1, Y_2, \dots, Y_s)$  has an extension that is a properly colored path decomposition of  $G$ .*

**Proof:** Suppose  $(X_1, X_2, \dots, X_r, Z_1, Z_2, \dots, Z_{r'})$  is a properly colored path decomposition of  $G$  that is an extension of  $(X_1, X_2, \dots, X_r)$ . Let  $g$  be the bijective function as in the definition of equivalence. Let  $f_i$  be a color preserving graph isomorphism from  $G[X_r \cup W_i]$  to  $G[X_r \cup W_{g(i)}]$  that is the identity on  $X_r$ , as implied by the definition of equivalence.

Let  $f : V \rightarrow V$  be the function defined in the following way.

- for  $v \in W_i$ ,  $1 \leq i \leq r$ :  $f(v) = f_i(v)$ ;
- for  $v \in X_r$ ,  $f(v) = v$ .

Simple case analysis (no, one or both endpoints in  $X_r$ ) shows that  $f$  is an automorphism of  $G$ . Define for  $i$ ,  $1 \leq i \leq r'$ ,  $Z'_i = \{f(v) \mid v \in Z_i\}$ .

**Claim 18**  *$(Y_1, Y_2, \dots, Y_s, Z'_1, Z'_2, \dots, Z'_{r'})$  is a properly colored path decomposition.*

**Proof:** We first prove that  $(Y_1, Y_2, \dots, Y_s, Z'_1, Z'_2, \dots, Z'_{r'})$  is a path decomposition. Clearly,  $(\bigcup_{1 \leq i \leq s} Y_i) \cup (\bigcup_{1 \leq i \leq r'} Z'_i) = V$ .

Second, we show that every edge  $\{v, w\} \in E$  is contained in some bag of  $(Y_1, Y_2, \dots, Y_s, Z'_1, Z'_2, \dots, Z'_{r'})$ . If  $v, w \in Y_s$ , then we can take the bag  $Y_s$ ; so w.l.o.g., let  $v \notin Y_s$ . If  $v$  belongs to a left component  $W_i$ , then there must be a bag  $Y_j$ ,  $1 \leq j \leq s-1$  that contains  $v$  and  $w$  as  $(Y_1, Y_2, \dots, Y_s)$  is a partial path decomposition. If  $v$  belongs to a right component  $W_i$ , then  $\{f^{-1}(v), f^{-1}(w)\} \in E$ . It is not hard to see that the bag in  $(X_1, X_2, \dots, X_r, Z_1, Z_2, \dots, Z_{r'})$  that contains both  $v$  and  $w$  must be one of the  $Z_j$ ,  $1 \leq j \leq r'$ , and thus  $v, w \in Z'_j$ .

Third, we show that every  $v \in V$  only occurs in a series of consecutive bags. For a vertex  $v \in Y_s = X_r$ , we note that there are  $1 \leq \alpha \leq s$ ,  $0 \leq \beta \leq r'$ , such that  $v$  belongs to bags  $Y_\alpha, Y_{\alpha+1}, \dots, Y_s$ , and  $v$  belongs to bags  $Z_1, Z_2, \dots, Z_\beta$ , and no other bags. As  $f(v) = v$ ,  $v$  also belongs to bags  $Z'_1, Z'_2, \dots, Z'_\beta$ , and no later bags. So, for a vertex  $v \in Y_s = X_r$ , we are done.

If  $v \in W_{g(i)}$  where  $W_{g(i)}$  is a left component of  $(Y_1, Y_2, \dots, Y_s)$ . Then,  $f^{-1}(v) \in W_i$  with  $W_i$  a left component of  $(X_1, X_2, \dots, X_r)$ . Thus,  $f^{-1}(v)$  belongs to one or more consecutive bags in  $(X_1, X_2, \dots, X_{r-1})$ , and, as  $f^{-1}(v)$  does not belong to  $X_r$ ,  $f^{-1}(v)$  does not belong to  $Z_1, Z_2, \dots, Z_{r'}$  because otherwise  $(X_1, X_2, \dots, X_r, Z_1, Z_2, \dots, Z_{r'})$  is not a path decomposition. So,  $v$  belongs to one or more consecutive bags in  $(Y_1, Y_2, \dots, Y_{s-1})$  and no others. And, if  $v \in W_{g(i)}$  where  $W_{g(i)}$  is a right component of  $(Y_1, Y_2, \dots, Y_s)$ , then the required result follows from a similar analysis.

Finally, by assumption all vertices in a bag  $Y_i$  have a different color, and, as  $f$  is color preserving, as all vertices in a bag  $Z_i$  have a different color, also all vertices in a bag  $Z'_i$  have a different color.  $\square$

So,  $(Y_1, Y_2, \dots, Y_s)$  has an extension that is a properly colored path decomposition of  $G$ . This shows one direction of implication of the proposition; the proof of the other direction is identical.  $\square$

**Definition 19** *Assume some ordering on the vertices. The characteristic of a partial path decomposition  $(X_1, X_2, \dots, X_r)$  is the following pair:*

$$(X_r, \bigcup_{1 \leq i \leq r-1} X_i - X_r),$$

where it is assumed that both vertex sets are given as an ordered list of vertices.

Two properly colored partial path decompositions with the same characteristic are trivially equivalent, using the identity for  $g$ . We remark that one can obtain an  $\mathcal{O}^*(2^n)$  time algorithm for INTERVALIZING  $k$ -COLORED GRAPHS by tabulating all different characteristics of properly colored partial path decompositions; this is somewhat similar to the Held-Karp algorithm for TSP [19]. The isomorphism check for components is the main ingredient of our improvement upon this idea.



## 4 An Exact Algorithm for Intervalizing $k$ -Colored Graphs

In this section, we give the algorithm for INTERVALIZING  $k$ -COLORED GRAPHS, building upon the notions and preliminary results of the previous sections.

First, we note that a positive instance has a path decomposition in which each bag has size at most  $k$  (all vertices in a bag have a different color and there are  $k$  colors). Thus, as a first step we use the linear time algorithm (for fixed  $k$ ), that tests if the pathwidth of the input graph is at most  $k - 1$  from [4, 11]. If not, we are done, and can decide negatively. Thus, we can assume that  $G$  has pathwidth at most  $k$  in the remainder. We consider  $k$  to be a constant.

We introduce some further notions.

**Definition 20** *We define the progress of a partial path decomposition  $(X_1, X_2, \dots, X_r)$  to be  $2 \cdot |\bigcup_{1 \leq i \leq r} X_i| - |X_r|$ . Note that when we extend a nice partial path decomposition with one additional introduce or one additional forget node, then the progress always increases by exactly one. Also note that for a partial path decomposition with characteristic  $(X, Z)$  and progress  $\alpha$ , we have that  $\alpha = 2|Z| - |X|$ .*

**Definition 21** *The canonical characteristic of a properly colored partial path decomposition is the lexicographically minimal characteristic over all characteristics of equivalent properly colored partial path decompositions.*

**Proposition 22** *Given a characteristic of a properly colored partial path decomposition, we can compute in polynomial time its canonical characteristic.*

**Proof:** The GRAPH ISOMORPHISM problem is polynomial time solvable on graphs of bounded treewidth, and thus also on graphs of bounded pathwidth [3]. It is straightforward to modify the algorithm of [3] such that it also works on colored graphs while using the same running time.

Given a characteristic  $(X_r, Z)$ , we first compute (with depth first search) the connected components of  $G[V - X_r]$ , say  $W_1, W_2, \dots, W_q$ . For each pair  $W_i, W_j$ , we can check in polynomial time if they are isomorphic: use the isomorphism algorithm on colored graphs of bounded pathwidth discussed above, and take a new, different color for each vertex in  $X_r$ . (Note the definition of isomorphism for components, as given in Section 3.)

Thus, we can partition the components in equivalence classes dictated by isomorphism. We can sort each component lexicographically, and then each class lexicographically. Then, for each class, we determine how many components from the class are a subset of  $Z$  (i.e., left components). In the canonical characteristic, we take the same number of left components from the class, but now take this number of lexicographically smallest elements. A simple last sorting step gives the desired result.  $\square$

We can now describe our algorithm.

- Check if the pathwidth of  $G$  is at most  $k - 1$ . If not, answer no and terminate.
- Otherwise, for  $\alpha = 1 \cdots 2n$ , compute a table  $T_\alpha$  of all canonical characteristics of partial path decompositions of progress  $\alpha$ .
- If  $T_{2n}$  is empty, then answer no; otherwise, answer yes.

The output of the algorithm clearly is correct as a partial path decomposition is a path decomposition, if and only if, its progress equals  $2n$ .

We now describe how the tables  $T_i$  are computed. Computing  $T_1$  is simple: for all  $v \in V$ , we have an entry in  $T_1$  of the form  $(\{v\}, \emptyset)$ . Given a table  $T_\alpha$ ,  $1 \leq \alpha < 2n$ , we compute table  $T_{\alpha+1}$  as follows. Initialize  $T_{\alpha+1}$  as empty set. For each entry  $(X, Z)$  from  $T_\alpha$ , do the following:

- Compute the new characteristics that result when the next node in the partial path decomposition is an introduce node: for each  $v \in V - Z$  such that there is no  $x \in X$  with  $c(v) = c(x)$ , compute the canonical characteristic of  $(X \cup \{v\}, Z)$  and put it in  $T_{\alpha+1}$ .
- Compute the new characteristics that result when the next node in the partial path decomposition is a forget node: for each  $x \in X$  such that there is no  $v \in Z - V$  with  $\{v, x\} \in E$ , compute the canonical characteristic of  $(X - \{v\}, Z \cup \{v\})$ .

**Proposition 23** *The procedure correctly computes table  $T_{\alpha+1}$ .*

**Proof:** Note that the characteristic of a partial path decomposition remains the same when we apply the procedure of Proposition 6. So, we may assume that we compute the canonical characteristics of the properly colored nice partial path decompositions  $(X_1, X_2, \dots, X_r)$  with progress  $\alpha + 1$ . Of these, we consider two cases: the last node  $X_r$  can be an introduce node or a forget node.

If  $X_r$  is an introduce node with  $X_r = X_{r-1} \cup \{v\}$ , then  $(X_1, X_2, \dots, X_{r-1})$  is a properly colored partial path decomposition of progress  $\alpha$ . If  $(X_1, X_2, \dots, X_{r-1})$  has characteristic  $(X_{r-1}, Z)$ , then  $(X_1, X_2, \dots, X_r)$  has characteristic  $(X_{r-1} \cup \{v\}, Z)$ .  $v$  must have a color different from the colors of vertices in  $X_{r-1}$ .

If  $X_r$  is a forget node with  $X_r = X_{r-1} - \{v\}$ , then again  $(X_1, X_2, \dots, X_{r-1})$  is a properly colored partial path decomposition of progress  $\alpha$ . As  $v$  is forgotten, it cannot belong to bags right of  $X_r$ , and thus all neighbors of  $v$  must belong to  $\bigcup_{1 \leq i \leq r} X_i$ . If  $(X_1, X_2, \dots, X_{r-1})$  has characteristic  $(X_{r-1}, Z)$ , then the characteristic of  $(X_1, X_2, \dots, X_r)$  is  $(X_{r-1} - \{v\}, Z \cup \{v\})$ .  $\square$

This completes the description of the algorithm. From our discussion, we see that the algorithm indeed correctly decides if  $G$  has a properly colored interval completion.

We now will analyze the running time of the algorithm. We remark that our algorithm uses polynomial time per entry in a table  $T_i$ . Thus, the running time of the algorithm equals the product of a polynomial in  $n$  and the number of canonical characteristics of

properly colored partial path decompositions. So, we need to establish an upper bound on this number of canonical characteristics. First, we obtain an upper bound on the number of nonisomorphic components of a set  $X$ .

**Proposition 24** *Let  $(X_1, X_2, \dots, X_r)$  be a properly colored partial path decomposition of  $G$ . There are at most  $2^{3kl} \cdot k^\ell$  equivalence classes of the isomorphism relation on components of  $G[V - X_r]$  that contain components with  $\ell$  vertices.*

**Proof:** Each equivalence class can be identified by an uncolored unlabeled graph on  $\ell$  vertices of pathwidth at most  $k - 1$ , a coloring with at most  $k$  colors of the vertices of the graph, and the incidence relation between the vertices in the graph and the vertices in  $X_r$ . This gives at most the following number of equivalence classes:

$$(k - 1 + 2^{k-1} + 1)^{2\ell-1} \cdot k^\ell \cdot 2^{k\ell} \leq k^\ell \cdot 2^{3kl}$$

because the first gives at most  $(k - 1 + 2^{k-1} + 1)^{2\ell-1}$  possibilities by Proposition 7, the second at most  $k^\ell$  possibilities, and the last at most  $2^{k\ell}$  possibilities.  $\square$

We fix some integer  $\ell$ ,  $1 \leq \ell \leq n$ , which we will determine more precisely later.

First, for a given  $X \subseteq V$ , we derive an upper bound on the number of canonical characteristics of the form  $(X, Z)$ . Consider the equivalence classes of the isomorphism relation on the components of  $G[V - X]$ . The characteristic is completely determined if we know  $X$ , and for each of these classes how many left components it contains, i.e., how many of the components are a subset of  $Z$ . In counting the number of possibilities, we distinguish two cases:

- ‘Large’ components of  $G[V - X]$ , i.e., components that contain more than  $\ell$  vertices. For each, we have the possibility to be a left or a right component. As there are at most  $n/\ell$  large components, this gives in total at most  $2^{n/\ell}$  possibilities for the large components.
- ‘Small’ components of  $G[V - X]$ , i.e., components that contain at most  $\ell$  vertices. There are less than  $\ell \cdot 2^{3kl} \cdot k^\ell$  equivalence classes of the isomorphism relation that contain small components, by Proposition 24(iii). As there are less than  $n$  components, for each of these classes we have less than  $n$  possibilities for the canonical characteristic (each possibility has a different number of left components). So, we have less than  $n^{\ell 2^{3kl} \cdot k^\ell}$  possibilities for the small components.

This gives, for some fixed  $X$ , an upper bound of

$$2^{n/\ell} \cdot n^{\ell 2^{3kl} \cdot k^\ell}$$

characteristics of the form  $(X, Z)$ . As we never consider sets  $X$  with more than  $k$  vertices, we can multiply this number by  $(n + 1)^k$  to obtain the following result.

**Lemma 25** *The size of the table  $T_i$  is bounded by  $(n + 1)^k \cdot 2^{n/\ell} \cdot n^{\ell 2^{3kl} \cdot k^\ell}$ .*

It follows that the running time of our algorithm is bounded by

$$\mathcal{O}^*(2^{n/\ell} \cdot n^{\ell 2^{3kl} \cdot k^\ell})$$

We will now choose the value of  $\ell$ : set  $\ell = (\log^{1-\delta}(n))$ . Then:

$$\begin{aligned} 2^{n/\ell} \cdot n^{\ell 2^{3kl} \cdot k^\ell} &\leq 2^{n/\log^{1-\delta}(n)} \cdot 2^{\log(n) \cdot \log^{1-\delta}(n) \cdot 2^{3k \log^{1-\delta}(n)} \cdot k^{\log^{1-\delta}(n)}} \\ &\leq 2^{n/\log^{1-\delta}(n) + \log^2(n) \cdot 2^{3k \log^{1-\delta}(n)} \cdot k^{\log^{1-\delta}(n)}} \end{aligned}$$

Note that for fixed  $k$ ,

$$\log^2(n) \cdot 2^{3k \log^{1-\delta}(n)} \cdot k^{\log^{1-\delta}(n)} = o\left(\frac{n}{\log^{1-\delta}(n)}\right)$$

(This can be seen as follows. The logarithm of the left term is  $\Theta(\log^{1-\Delta} n)$ , for fixed  $k$ , while  $\log(\frac{n}{\log^{1-\delta} n})$  is  $\Theta(\log n)$ .)

Thus, for fixed  $k$  and for every  $\epsilon > 0$ , there is a  $\delta > 0$ , and an  $n_0 \in \mathbb{N}$  such that for all  $n \geq n_0$ :

$$2^{n/\ell} \cdot n^{\ell 2^{3kl} \cdot k^\ell} \leq 2^{\frac{n}{\log^{1-\epsilon} n}}$$

We have now shown that the size of the tables in our algorithm, and thus the running time of our algorithm is, for every fixed  $k$  and every  $\epsilon > 1$ ,  $\mathcal{O}^*(2^{\frac{n}{\log^{1-\epsilon} n}})$ .

**Theorem 26** *For every fixed  $k \geq 4$ , there is an algorithm for INTERVALIZING  $k$ -COLORED GRAPHS that runs in time  $\mathcal{O}^*(2^{\frac{n}{\log^{1-\epsilon} n}})$  for every  $\epsilon > 0$ .*

We remark that there are inputs on which the algorithm uses  $\Omega(2^{n/\log n})$  time: suppose  $G$  has a vertex  $v$  that is a separator such that  $G[V - \{v\}]$  has  $\Omega(n/\log n)$  non-isomorphic components each of size  $\lceil \log n \rceil$ .

## 5 An Algorithm for Intervalizing Colored Graphs with an Arbitrary Number of Colors

In this section, we consider the case that the number of colors is not fixed. We give a simple Held-Karp style dynamic programming algorithm for this problem.

**Definition 27** *Suppose we are given a properly colored graph  $G = (V, E)$ . For a given set of vertices  $W \subseteq V$ , the border of  $W$  is the set of vertices in  $W$  with at least one neighbor in  $V - W$ , i.e., we denote*

$$B(W) = \{v \in W \mid \exists w \in V - W : \{v, w\} \in E\}$$

**Definition 28** *A set of vertices  $W \subseteq V$  is said to be fine, if there exists a properly colored path decomposition  $(X_1, X_2, \dots, X_s)$  of  $G[W]$ , such that  $B[W] \subseteq X_s$ , i.e., the last bag contains all vertices in the border of  $W$ .*

**Lemma 29** *For all  $W \subseteq V$ ,  $W \neq \emptyset$ ,  $W$  is fine, if and only if, there exists a  $v \in W$ , such that  $W - \{v\}$  is fine and all vertices in  $B(W - \{v\}) \cup \{v\}$  have a different color.*

**Proof:** Suppose  $W$  is fine. Suppose  $(X_1, X_2, \dots, X_s)$  is a properly colored path decomposition of  $G[W]$  with  $B(W) \subseteq X_s$ . If  $s = 1$ , the result follows directly (any vertex in  $X_1$  can play the role of  $v$ ). Suppose  $s > 1$ . If  $X_s \subseteq X_{s-1}$ , then  $(X_1, \dots, X_{s-1})$  is also a properly colored path decomposition of  $G[W]$  with  $B(W) \subseteq X_s$ , and we look at this path decomposition instead. Repeat the step till  $X_s \not\subseteq X_{s-1}$  or  $s = 1$ . So, we may suppose that  $X_s \not\subseteq X_{s-1}$ .

Take a vertex  $v \in X_s - X_{s-1}$ .  $X_s$  must contain each vertex  $w \in B(W - \{v\})$ , as for each such  $w$ , either  $w \in B(W)$  or  $\{v, w\} \in E$ . So all vertices in  $B(W - \{v\}) \cup \{v\} \subseteq X_s$  have a different color.  $W - \{v\}$  is fine, as  $(X_1, X_2, \dots, X_{s-1}, X_s - \{v\})$  fulfills the stated condition.

For the other direction, suppose that  $W - \{v\}$  is fine, and all vertices in  $B(W - \{v\}) \cup \{v\}$  have a different color. Let  $(Y_1, Y_2, \dots, Y_r)$  be a properly colored path decomposition with  $B(W - \{v\}) \subseteq Y_r$ . A simple case analysis shows that  $(Y_1, Y_2, \dots, Y_r, B(W - \{v\}) \cup \{v\})$  is a properly colored path decomposition of  $G[W]$  with  $B(W) \subseteq B(W - \{v\}) \cup \{v\}$ . E.g., each neighbor in  $v$  that belongs to  $W - \{v\}$  belongs to  $B(W - \{v\})$ , and thus to the last bag.  $\square$

Lemma 29 directly implies the existence of a dynamic programming algorithm that uses  $\mathcal{O}^*(2^n)$  time. For  $i = 0, 1, \dots, n$ , we compute the collection of fine sets  $W$  with  $|W| = i$ ; call this collection  $F(i)$ . For  $i = 0$ , we note that the empty set is fine, i.e.,  $F(0) = \{\emptyset\}$ . If  $i > 0$ , initialize  $F(i)$  as an empty collection. Then, perform the following step for each fine set  $Y \in F(i - 1)$ :

- Compute the border of  $Y$ ,  $B[Y]$ . This can be done in linear time using depth first search.
- If  $B[Y]$  contains two vertices of the same color, we do not further process  $Y$ , otherwise continue with the next step.
- For all vertices  $v \in V - Y$ ,
  - Check if  $B[Y]$  contains a vertex with the same color as  $v$ .
  - If not, then  $Y \cup \{v\}$  is a fine set of size  $i$ . If  $F(i)$  does not yet contain  $Y \cup \{v\}$ , then add  $Y \cup \{v\}$  as a new element to  $F(i)$ .

It is easy to see that the amount of work per fine set of vertices is polynomial. Finally,  $G$  has a properly colored interval completion, if and only if  $F(n) \neq \emptyset$ . Thus, we have

**Theorem 30** *The INTERVALIZING COLORED GRAPHS problem can be solved in  $\mathcal{O}^*(2^n)$  time.*

## 6 Conclusions

In this paper, we gave dynamic programming algorithms for the INTERVALIZING  $k$ -COLORED GRAPHS problem. Our algorithm for the case that the number of colors  $k$  is fixed uses subexponential time of a somewhat unusual form, and thus, the result forms a somewhat curious exception to the types of results that are usually obtained in the field. The result is merely of theoretical interest, as values of  $n$  for which the algorithm can be run in practice can be expected to be rather small, say below 100. Experiments with a somewhat similar Held-Karp style algorithm for TREEWIDTH [8, 9] suggest that our algorithm can also be practical for small values of  $n$ ; probably a good modification would be to run the isomorphism test only for very small components, and with a usual graph isomorphism heuristic instead of the algorithm from [3].

A generalization of the INTERVALIZING  $k$ -COLORED GRAPHS problem is the INTERVAL GRAPH SANDWICH problem, in which we are given two graphs with  $G$  and  $H$  with the same vertex set, and ask whether there exists an interval graph  $G'$  that is a subgraph of  $H$  and contains  $G$  as a subgraph. A well studied variant has the additional condition that  $G'$  has maximum clique size  $k$ . See e.g., [17, 20]. The ideas of our paper seem not to give results better than an algorithm that uses  $\Theta^*(2^n)$  time for this problem however, still assuming that  $k$  is fixed.

Other related problems are the version where we ask to find a properly colored *proper interval graph*, which is polynomial for a fixed number of colors  $k$  [2], and the problem to find a properly colored *chordal graph*, which is also polynomial for a fixed number of colors [22].

An interesting open problem is whether it is possible to obtain faster exact algorithms for INTERVALIZING  $k$ -COLORED GRAPHS, e.g., is  $\mathcal{O}^*(c^{\sqrt{n}})$  possible? Also, are faster algorithms possible for the case without a bound on the number of colors?

## References

- [1] C. Àlvarex, J. Díaz, and M. Serna. The hardness of intervalizing four colored caterpillars. *Discrete Mathematics*, 235:19–27, 2001.
- [2] C. Àlvarex and M. Serna. On the proper intervalization of colored caterpillar trees. *Informatique Théorique et Applications*, 43:667–686, 2010.
- [3] H. L. Bodlaender. Polynomial algorithms for graph isomorphism and chromatic index on partial  $k$ -trees. *Journal of Algorithms*, 11:631–643, 1990.
- [4] H. L. Bodlaender. A linear time algorithm for finding tree-decompositions of small treewidth. *SIAM Journal on Computing*, 25:1305–1317, 1996.
- [5] H. L. Bodlaender and B. de Fluiter. Intervalizing  $k$ -colored graphs. In Z. Fülöp and F. Gécseg, editors, *Proceedings of the 22nd International Colloquium on Automata*,

- Languages and Programming, ICALP'95*, pages 87–98. Springer Verlag, Lecture Notes in Computer Science, vol. 944, 1995.
- [6] H. L. Bodlaender and B. de Fluiter. On intervalizing  $k$ -colored graphs for DNA physical mapping. *Discrete Applied Mathematics*, 71:55–77, 1996.
  - [7] H. L. Bodlaender and B. L. E. de Fluiter. Intervalizing  $k$ -colored graphs. Technical Report UU-CS-1995-15, Department of Computer Science, Utrecht University, Utrecht, the Netherlands, 1995.
  - [8] H. L. Bodlaender, F. V. Fomin, A. M. C. A. Koster, D. Kratsch, and D. M. Thilikos. On exact algorithms for treewidth. In Y. Azar and T. Erlebach, editors, *Proceedings of the 14th Annual European Symposium on Algorithms, ESA 2006*, pages 672–683. Springer Verlag, Lecture Notes in Computer Science, vol. 4168, 2006.
  - [9] H. L. Bodlaender, F. V. Fomin, A. M. C. A. Koster, D. Kratsch, and D. M. Thilikos. On exact algorithms for treewidth. Technical Report UU-CS-2006-032, Department of Information and Computing Sciences, Utrecht University, Utrecht, the Netherlands, 2006.
  - [10] H. L. Bodlaender, F. V. Fomin, A. M. C. A. Koster, D. Kratsch, and D. M. Thilikos. A note on exact algorithms for vertex ordering problems on graphs. Technical Report UU-CS-2009-023, Department of Information and Computer Sciences, Utrecht University, Utrecht, the Netherlands, 2009.
  - [11] H. L. Bodlaender and T. Kloks. Efficient and constructive algorithms for the pathwidth and treewidth of graphs. *Journal of Algorithms*, 21:358–402, 1996.
  - [12] E. D. Demaine and M. Hajiaghayi. The bidimensionality theory and its algorithmic applications. *The Computer Journal*, 51:292–302, 2008.
  - [13] M. R. Fellows, M. T. Hallett, and H. T. Wareham. DNA physical mapping: Three ways difficult (extended abstract). In T. Lengauer, editor, *Proceedings of the 1st Annual European Symposium on Algorithms, ESA '93*, pages 157–168. Springer Verlag, Lecture Notes in Computer Science, vol. 726, 1993.
  - [14] F. V. Fomin and D. M. Thilikos. A simple and fast approach for solving problems on planar graphs. In *Proceedings 21st International Symposium on Theoretical Aspects of Computer Science, STACS 2004*, pages 56–67. Springer Verlag, Lecture Notes in Computer Science, vol. 2996, 2004.
  - [15] M. C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs*. Academic Press, New York, 1980.
  - [16] M. C. Golumbic, H. Kaplan, and R. Shamir. On the complexity of DNA physical mapping. *Advances in Applied Mathematics*, 15:251–261, 1994.

- [17] M. C. Golumbic, H. Kaplan, and R. Shamir. Graph sandwich problems. *Journal of Algorithms*, 19:449–472, 1995.
- [18] P. Heggernes, K. Suchan, I. Todinca, and Y. Villanger. Minimal interval completions. In *Proceedings of the 13th Annual European Symposium on Algorithms, ESA 2005*, pages 403–414. Springer Verlag, Lecture Notes in Computer Science, vol. 3669, 2005.
- [19] M. Held and R. Karp. A dynamic programming approach to sequencing problems. *Journal of the Society for Industrial and Applied Mathematics*, 10:196–210, 1962.
- [20] H. Kaplan and R. Shamir. Bounded degree interval sandwich problems. *Algorithmica*, 24:96–104, 1999.
- [21] R. J. Lipton and R. E. Tarjan. Applications of a planar separator theorem. *SIAM Journal on Computing*, 9:615–627, 1980.
- [22] F. R. McMorris, T. Warnow, and T. Wimer. Triangulating vertex-colored graphs. *SIAM Journal on Discrete Mathematics*, 7(2):296–306, 1994.