Bounded Representations of Interval and Proper Interval Graphs*

Martin Balko**, Pavel Klavík***, and Yota Otachi†

Abstract. Klavík et al. [arXiv:1207.6960] recently introduced a generalization of recognition called the *bounded representation problem* which we study for the classes of interval and proper interval graphs. The input gives a graph G and in addition for each vertex v two intervals $\mathfrak{L}_v$ and $\mathfrak{R}_v$ called *bounds*. We ask whether there exists a bounded representation in which each interval I_v has its left endpoint in $\mathfrak{L}_v$ and its right endpoint in $\mathfrak{R}_v$. We show that the problem can be solved in linear time for interval graphs and in quadratic time for proper interval graphs.

Robert's Theorem states that the classes of proper interval graphs and unit interval graphs are equal. Surprisingly the bounded representation problem is polynomially solvable for proper interval graphs and NP-complete for unit interval graphs [Klavík et al., arxiv:1207.6960]. So unless $\mathsf{P} = \mathsf{NP}$, the proper and unit interval representations behave very differently.

The bounded representation problem belongs to a wider class of restricted representation problems. These problems are generalizations of the well-understood recognition problem, and they ask whether there exists a representation of G satisfying some additional constraints. The bounded representation problems generalize many of these problems.

1 Introduction

In the recent data-filled world, visualization and graph drawing is becoming an increasingly more important topic. One is frequently asked to work with a huge object and to understand its structure. In some cases, it is useful to visualize the object in a way which reveals its structure. A prime example of this is the class of *interval graphs* which is one of the oldest and best-understood graph classes. An interval graph G can contain many edges, so a standard drawing is not very understandable. But it has an *interval representation* $\mathcal{R}$ which is a collection of closed intervals $\{I_v : v \in V(G)\}$ representing the vertices of the graph such that

* The first two authors are supported by ESF Eurogiga project GraDR as GAČR GIG/11/E023.

** Department of Applied Mathematics, Faculty of Mathematics and Physics, Charles University, Malostranské náměstí 25, 118 00 Prague, Czech Republic. E-mail: `balko@kam.mff.cuni.cz`.

*** Supported by Charles University as GAUK 196213. Computer Science Institute, Faculty of Mathematics and Physics, Charles University, Malostranské náměstí 25, 118 00 Prague, Czech Republic. E-mail: `klavik@iuuk.mff.cuni.cz`.

† School of Information Science, Japan Advanced Institute of Science and Technology. Asahidai 1-1, Nomi, Ishikawa 923-1292, Japan. Email: `otachi@jaist.ac.jp`

$I_u \cap I_v \neq \emptyset$ if and only if $uv \in E(G)$. This representation nicely describes the structure of the edges. We denote the class of interval graphs by INT.

Interval graphs were first introduced by Hajós [10] in 1957. They caught quickly an attention of many researchers, for instance Benzer [1] used them in his experimental study of the DNA structure. The first polynomial-time recognition algorithms were given already in 1960's [9, 8]. After a decade, Booth and Lueker [3] finally described a linear-time recognition algorithm based on a new tree-structure called PQ-trees, applicable also to other problems such as planarity. Nowadays, there are over several hundred papers dealing with many aspects of interval graphs.

An interval representation is called *proper* if $I_u \subseteq I_v$ implies $I_u = I_v$, i.e., no interval is a proper subset of another interval. And it is called *unit* if all intervals are of unit length. We consider two important subclasses of interval graphs: *proper interval graphs* ($\mathsf{PROPER\ INT}$) are graphs which admit proper interval representations, and similarly for *unit interval graphs* ($\mathsf{UNIT\ INT}$). The well-known theorem of Roberts [18] states that $\mathsf{PROPER\ INT} = \mathsf{UNIT\ INT}$.

1.1 The Bounded Representation Problem

Several recent papers study *restricted representation problems* in which we ask whether there exists, say, an interval representation of an input graph G satisfying some additional constraints; see for example [16, 11, 12, 2, 17]. In this paper, we study for the classes INT and $\mathsf{PROPER\ INT}$ one such problem called *bounded representation*, recently introduced by Klavík et al. [13]. This problem is related to many other restricted representation problems; see Section 1.2 for details.

For an arbitrary interval I, we denote its left endpoint by $\ell(I)$ and its right endpoint by $r(I)$. Let $\mathfrak{L}_v$ and $\mathfrak{R}_v$ be two intervals defined for each $v \in V(G)$. A representation $\mathcal{R}$ is called a *bounded representation* if $\ell(I_v) \in \mathfrak{L}_v$ and $r(I_v) \in \mathfrak{R}_v$ for each $v \in V(G)$. The bounded representation problem is the following decision problem:

Problem:	The Bounded Representation Problem – BoundRep($\mathcal{C}$)
Input:	A graph G and two intervals $\mathfrak{L}_v$ and $\mathfrak{R}_v$ for each $v \in V(G)$.
Output:	Is there a bounded representation $\mathcal{R}$ of the class $\mathcal{C}$?

In the further text, we refer to the intervals $\mathfrak{L}_v$ as *left bounds* and to the intervals $\mathfrak{R}_v$ as *right bounds*, or just simply *bounds*. See Fig. 1a for an example of an BoundRep instance. It is easy to see that the bounded representation problem generalizes recognition; if all the bounds are set to $(-\infty, +\infty)$, they pose no restriction at all. We also allow trivial bounds consisting of single points.

1.2 Other Restricted Representation Problems

We review other restricted representation problems and discuss their relation to the bounded representation problem. The problems were considered for different intersection classes of graphs which we do not define formally; see the references for details. We note that all these problems generalize the recognition problem (Recog). See Fig. 1b for the relations between the problems.

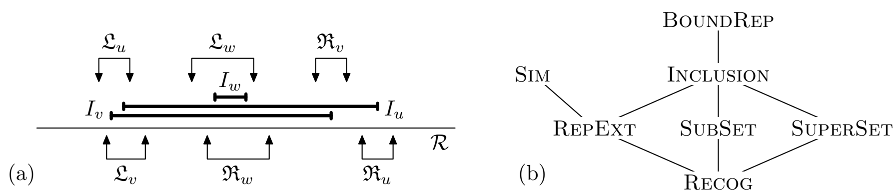

Fig. 1. (a) A bounded representation $\mathcal{R}$ of the class INT is given for a graph K_3. There exists no bounded proper interval representation since I_w is always a proper subset of I_u and I_v. (b) The Hasse diagram for different restricted representation problemsi. If $\mathcal{P} \le \mathcal{P}'$, then the problem $\mathcal{P}$ can be solved using the problem $\mathcal{P}'$.

Partial Representation Extension. This problem denoted by RepExt was introduced by Klavík et al. [16]. The input prescribes together with G an intersection representation $\mathcal{R}'$ of an induced subgraph G'. The goal is to find a representation $\mathcal{R}$ of the entire G which extends $\mathcal{R}'$, i.e., it assigns the same sets to the vertices of G' as $\mathcal{R}'$. The problem can be solved in polynomial time for interval graphs [16, 2, 15], proper and unit interval graphs [13], function and permutation graphs [12] and circle graphs [4]. For chordal graphs in the setting of subtree-in-a-tree graphs, several versions of the problem were considered in [14], and almost all of them are NP-complete. It is known that planar graphs have several intersection representations (contact representations of discs, etc.), but extending these representations is NP-hard [7].

The bounded representation problem generalizes partial representation extension, since one can prescribed singleton bounds for the intervals of G' and $(-\infty, +\infty)$ for the remaining bounds. (We note that the bounded representation problem can be considered also for many other classes of graphs.)

Inclusion Restrictions. Function graphs are intersection graphs of continuous functions defined on $[0, 1]$. In [12], the following problem was considered and proved to be NP-complete. The input prescribes some functions partially, i.e., on partial domains $[a, b] \subseteq [0, 1]$. The goal is to extend them to the full domain $[0, 1]$.

We consider more generally three different problems Inclusion, SubSet, and SuperSet for interval graphs. In all problems, the input gives two intervals A_v and B_v for each vertex $v \in V(G)$. The goal is to construct a representation such that $A_v \subseteq I_v \subseteq B_v$. Further for SubSet, we put all $A_v = \emptyset$, and for SuperSet, we put all $B_v = (-\infty, \infty)$. It is easy to see that these problems can be reduced to the bounded representation problems, and Inclusion can solve RepExt.

Simultaneous Representations. This problem denoted by Sim was introduced and solved for several classes by Jampani et al. [11]. The input consists of two graphs G_1 and G_2 with some common vertices. The goal is to construct their representations $\mathcal{R}_1$ and $\mathcal{R}_2$ such that the common vertices are represented the same. Bläsius et al. [2] reduce RepExt(INT) to Sim(INT), and thus solve the first problem in linear time. On the other hand, when the problem is generalized to k input graphs, the best known result for many classes is an FPT algorithm in the number of common vertices based on the partial representation extension [16, 4]. We are not aware of any relation of the simultaneous representations problem to the other considered problems.

Motivation. There are two very good motivations for studying the restricted representation problems. The first motivation is that they might be applicable. For instance, one might want to construct some specific representation of the given graph G. Using these restrictions, one can force the representation to be constructed in this way. The other motivation is that to solve these problems much better structural understanding is required. For classes like interval graphs, the structure of all representations is well understood and one can just use PQ-trees to solve the problems. For other classes like unit interval graphs [13] or circle graphs [4], the new structural results were developed which might be fruitful also for other purposes. In mathematics, it is generally desirable to have problems which force one to get better understanding of the objects.

1.3 Our Results

In this paper, we prove the following two theorems. For BoundRep(INT), we assume that the endpoints of the bounds are *sorted* from left to right, so we can work with the bounds efficiently. Otherwise, we need extra time $\mathcal{O}(n \log n)$ in the beginning.

Theorem 1. *The problem* BoundRep(INT) *with sorted endpoints of the bounds can be solved in time $\mathcal{O}(n+m)$ where n is the number of vertices and m is the number of edges.*

The algorithm of Theorem 1 is almost the same as the algorithm for RepExt(INT) of [16, 15]. So the techniques developed for the partial representation extension problem can be directly applied to more general problems.

Theorem 2. *The problem* BoundRep(PROPER INT) *can be solved in time $\mathcal{O}(n^2)$ where n is the number of vertices.*

We note that it was already observed in [16] that the classes of proper and unit interval graphs behave differently with respect to the partial representation problem; unit interval graphs put additional restrictions is the form of precise rational positions. In [13], the problem RepExt(UNIT INT) was solved in quadratic time by linear programming. So it seemed that this difference is only in some additional numerical problems posed by unit intervals. Theorem 2 shows together with the result of [13, Proposition 2] that this understanding is fundamentally wrong (unless $\mathsf{P} = \mathsf{NP}$, of course):

Theorem 3 (Klavík et al. [13]). *The problem* BoundRep(UNIT INT) *is* NP-*complete.*

The problem is reduced from 3-partition and the hard part is to derive a correct ordering $\blacktriangleleft$ of the components from left to right; if the ordering is prescribed, one can solve the problem in quadratic time. The main difference for proper interval graphs is Proposition 2 which allows us to derive this ordering

◀. The remainder of the algorithm works similarly as in [13], only some places are more technical since we have to deal with both left and right bounds; in the case of unit interval graphs, we can work only with left bounds since the position $\ell(I_v)$ determines the position $r(I_v)$.

2 Preliminaries

For a graph G, we denote by $V(G)$ the set of its vertices and by $E(G)$ the set of its edges. We use $N[u]$ for the *closed neighborhood* of the vertex u, i.e, $N[u] = \{v \in V(G) : uv \in E(G)\} \cup \{u\}$.

A *(partial) ordering* is a transitive, reflexive and antisymmetric relation. A *pre-ordering* is just a transitive and reflexive relation, so several elements can be equal in a pre-ordering. An ordering/pre-ordering is called *linear* if every two elements are comparable.

For an arbitrary subset S of the real line, we define $\ell(S) = \inf\{x : x \in S\}$ and $r = \sup\{x : x \in S\}$. By $\lessdot$, we denote the *subset ordering* where $S \lessdot T$ for two subsets S and T if and only if $r(S) \le \ell(T)$; in other words S is completely on the left of T.

Endpoint Pre-orderings. There are two very natural ways how one can work with intervals and interval representations. The first option is to assign to each interval I two rational numbers $\ell(I)$ and $r(I)$. The second option which we prefer in this paper is just to consider the ordering $<$ of the endpoints as they appear from left to right. The reason is that this ordering contains all information about intersections of intervals; precise rational positions are not needed, we can just work with a topology of the representation. We note that this is not the case of unit interval representations, for which one has to consider precise rational number positions.

In the case of general interval graphs, one can assume that no two endpoints share their positions. For bounded representations, this is not true anymore since the bounds might force shared positions. In this case, $\le$ is a linear pre-ordering, with some sets of endpoints being equal in it. We say that an endpoint z is *in between* of x and y if $x \le z \le y$. If two endpoints x and y share position, we denote it by $x = y$, and by $x < y$ we denote that x is strictly on the left of y. It is important to state that if $x < y$, then one can add in between of x and y an arbitrary number of endpoints in any pre-ordering. If $x = y$, then only endpoints sharing the position with x and y can be added in between.

If we work with representations just as with left-to-right pre-orderings of the endpoints, then how can we decide whether the endpoints lie in the bounds? Our assumption on the input is that we are given a linear pre-ordering of the endpoints of the bounds $\mathfrak{L}_v$ and $\mathfrak{R}_v$. The solution gives a bounded representation $\mathcal{R}$ in the form of a joined pre-ordering $\le$ of the endpoints of the bounds and the intervals. The bounds constraints just say that $\ell(\mathfrak{L}_v) \le \ell(I_v) \le r(\mathfrak{L}_v)$ and $\ell(\mathfrak{R}_v) \le r(I_v) \le r(\mathfrak{R}_v)$.

Simplifying Bounds. For each interval I_v, we want $\ell(I_v) \le r(I_v)$. So we assume each pair $\mathfrak{L}_v$ and $\mathfrak{R}_v$ satisfies $\ell(\mathfrak{L}_v) \le \ell(\mathfrak{R}_v)$ and $r(\mathfrak{L}_v) \le r(\mathfrak{R}_v)$. Otherwise we modify the instance by putting $\ell(\mathfrak{R}_v) := \ell(\mathfrak{L}_v)$, resp. $r(\mathfrak{L}_v) := r(\mathfrak{R}_v)$.

3 Bounded Representations of Interval Graphs

In this section we establish Theorem 1 which states that the problem BoundRep(INT) can be solved in time $\mathcal{O}(n+m)$ (given the pre-ordering of the endpoints of the bounds). First, we give a characterization of bounds for which the bounded representation exists. Then we describe the algorithm which checks this characterization, and since it is constructive, it can construct the bounded representation if it exists. We note that our approach is very similar to [15].

3.1 Characterization of Fulkerson and Gross

Fulkerson and Gross [8] gave the following characterization:

Lemma 1 (Fulkerson and Gross). *A graph G is an interval graph if and only if there exists a linear ordering $<$ of the maximal cliques of G such that for every vertex $v \in V(G)$ the cliques containing v appear consecutively in $<$.*

Proof (Sketch). We sketch this proof since it is important to understand the characterization. Let $\mathcal{R}$ be an interval representation. For each maximal clique C, we consider $\bigcap_{v \in C} I_v$, and according to Helly's theorem this intersection is non-empty. We pick an arbitrary point from this intersection, and we call it a *clique-point* and denote it by $\mathrm{cp}(C)$. Since these intersections are for different maximal cliques pairwise distinct, the clique-points are linearly ordered from left to right. It is routine to check that this is the ordering $<$ from Lemma 1.

On the other hand, given an ordering $<$ of the maximal cliques, we place clique-points arbitrarily in this ordering. Then for each vertex v, we put

$$\ell(I_v) = \min\{\mathrm{cp}(C) : v \in C\}, \qquad \text{and} \qquad r(I_v) = \max\{\mathrm{cp}(C) : v \in C\}, \tag{1}$$

i.e., we place I_v on top of the clique-points of cliques containing v. We obtain a correct interval representation of G. ☐

3.2 Orderings of Maximal Cliques Compatible with the Bounds

We want to construct a bounded representation in a similar manner, first by placing the clique-points from left to right and then by constructing the intervals using (1). But to ensure that the resulting representation is bounded, we cannot place the clique-points arbitrarily. For a maximal clique C, we denote by J_C the set of possible positions where $\mathrm{cp}(C)$ can be placed; see Appendix A for the precise definition.

But now if $J_C \lessdot J'_C$, we know that $\mathrm{cp}(C)$ has to be always placed on the left of $\mathrm{cp}(C')$; so $\lessdot$ on the sets J_C gives the partial ordering of the cliques from left to right which we denote $\lessdot$ as well.

Proposition 1. *There exists a bounded representation $\mathcal{R}$ if and only if there is an ordering $<$ of the maximal cliques which is consecutive in every vertex v and extends $\lessdot$.*

Proof (Sketch). The constraints are necessary. We place the clique-points greedily from left to right according to the ordering $<$. When we place $\mathrm{cp}(C)$, we place it on the right of the previously placed clique-point and in J_C. For contradiction suppose that no such point of J_C. We obtain a contradiction with the consecutivity property or the ordering $\lessdot$. The full proof is in Appendix A. $\square$

3.3 The Algorithm

To solve BoundRep(INT), we proceed in the following main steps.

(1) We find maximal cliques of G, using the algorithm of Rose et al. [19] in time $\mathcal{O}(n+m)$.
(2) We compute the sets J_C, this can be done by a single sweep from left to right in time $\mathcal{O}(n+m)$. This gives us the partial ordering $\lessdot$ of maximal cliques according to which the clique-points have to appear on the real line.
(3) Test whether there is a linear ordering $<$ of the maximal cliques which extends $\lessdot$ and for each vertex the maximal cliques containing it appear consecutively. This can be done using [15, Section 2].
(4) If there is a suitable reordering $<$ of $\lessdot$, then we place the clique-points as in the proof of Proposition 1. Using (1) we construct a correct bounded representation $\mathcal{R}$ of G.

Note that if we only want to decide BoundRep(INT) without constructing a representation, then the last step can be omitted. For the first step, the input graph has to be chordal and then the total size of all cliques is $\mathcal{O}(n+m)$.

The constructed representation with $\mathrm{cp}(C) \in J_C$ is correct since we have $\ell(I_v) \in \mathfrak{L}_v$ and $r(I_v) \in \mathfrak{R}_v$ for each $v \in V(G)$. Moreover $\mathcal{R}$ is an interval representation of G, since every clique-point lies exactly in the intervals representing the vertices of the corresponding maximal clique. Thus we can summarize the results.

Proof (Theorem 1). The proof follows the steps described in the beginning of the section. The correctness of the algorithm is ensured by Proposition 1. We already observed that for a given pre-ordering $\leq$ of the endpoints of the bounds from left to right, the construction of the reordering $<$ of $\lessdot$ can be done in time $\mathcal{O}(m+n)$. Since the representation $\mathcal{R}$ can be also constructed in linear time with respect to the size of G, we see that the whole algorithm runs in time $\mathcal{O}(m+n)$. $\square$

4 Bounded Representations of Proper Interval Graphs

In this section, we establish Theorem 2 which states that the bounded representation problem of proper interval graphs can be solved in time $\mathcal{O}(n^2)$. Proper interval representations give two important orderings: the ordering $\blacktriangleleft$ of the components, and the ordering $\lhd$ of the intervals of the components. We first describe them in details and then we show how they can be used in solving of the BoundRep(PROPER INT) problem.

4.1 Component Orderings $\blacktriangleleft$

Let $\mathcal{R}$ be any representation of G and let C be a connected component. Then $\bigcup_{v\in C} I_v$ is a closed interval of the real line. Since the intervals corresponding to the components are pairwise disjoint, the components are ordered as $C_1 \blacktriangleleft \cdots \blacktriangleleft C_c$. Notice that for different representations we may get different orderings $\blacktriangleleft$, and when no restriction is posed on the representation, we can use each of the $c!$ possible orderings.

Suppose that $uv \notin E(G)$. We ask what conditions the bounds have to satisfy to determine that $I_u \lessdot I_v$ in any bounded representation of G. Since the intervals I_u and I_v do not intersect, it is sufficient to prove that $\ell(I_u)$ is always to the left of $r(I_v)$. This is clearly satisfied if and only if $\mathfrak{L}_u \lessdot \mathfrak{R}_v$.

For a given instance of the bounded representation problem, our goal is to determine some ordering $\blacktriangleleft$ in which a bounded representation exists. To do so, we derive a relation $\blacktriangleleft'$ such that the ordering $\blacktriangleleft$ of every bounded representation $\mathcal{R}$ has to extend $\blacktriangleleft'$. Let C and C' be two distinct components of G. We put $C \blacktriangleleft' C'$ if there exists a pair $u \in C$ and $v \in C'$ such that $\mathfrak{L}_u \lessdot \mathfrak{R}_v$.

The following proposition (whose proof can be found in Appendix B) states that respecting the ordering $\blacktriangleleft'$ is already sufficient for solving the bounded representation problem:

Proposition 2. *A bounded representation of G in an ordering $\blacktriangleleft$ extending $\blacktriangleleft'$ exists if and only if there exists a bounded representation of G.*

Proof (Sketch). We argue only the non-obvious direction. Suppose that C and C' are two components incomparable in $\blacktriangleleft'$. In such a case, their bounds have to be hugely overlapping. There are two cases one has to deal with:

- All bounds of C and C' are pairwise intersecting. Then due to Helly's theorem, we can represent C and C' in any ordering in this intersection.
- Only bounds of, say, C are pairwise intersecting. But then due to Helly's theorem, we can represent C either on the left of the left-most bound of C', or on the right of the right-most bound of C. We still leave enough space for C' to be represented.

Then we repeatedly apply this local reordering of incomparable components till we modify the given bounded representation in the prescribed ordering $\blacktriangleleft$ which extends $\blacktriangleleft'$. See Appendix B for further details. $\square$

We note that a similar proposition is not correct for unit interval graphs. The problem is that a component has some minimal size which it requires in every representation, so it cannot be placed in this arbitrary small common intersection of the bounds. Actually Klavík et al. [13, Theorem 1] proved that finding the correct ordering $\blacktriangleleft$ is the NP-complete part of the problem BoundRep(UNIT INT). For a prescribed ordering $\blacktriangleleft$, one can solve the bounded representation problem of unit interval graphs in quadratic time.

4.2 Vertex Orderings $\trianglelefteq$

Two vertices u and v are called *indistinguishable* if $N[u] = N[v]$. So being indistinguishable defines an equivalence relation on $V(G)$, and the classes of this equivalence are called *groups of indistinguishable vertices*. For every intersection representation, the vertices of each group can be represented the same, and so indistinguishable vertices are not very interesting from the structural point of view. This is not the case for the bounded representation problem (or any other problem of restricted representation), in which indistinguishable vertices can be given distinct bounds and thus are forced to be represented differently.

Vertex Orderings. Let $\mathcal{R}$ be any proper interval representation, and assume for a second that no two intervals of $\mathcal{R}$ are the same. Then the intervals are ordered from left to right, and we denote this ordering by $\lhd$. The ordering $\lhd$ is the ordering of the left endpoints, and at the same time the ordering of the right endpoints. In $\lhd$, each group of indistinguishable vertices has to appear consecutively. Deng et al. [6] characterize possible orderings $\lhd$ for connected proper interval graphs:

Lemma 2 (Deng et al.). *For a connected proper interval graph, the ordering $\lhd$ is uniquely determined up to local reordering of the groups of indistinguishable vertices and the complete reversal.*

In other words, there exists a partial ordering $<$ in which exactly the pairs of indistinguishable vertices are incomparable. Then each $\lhd$ is a linear extension of $<$ or its reversal. Corneil et al. [5] describe a simple linear-time algorithm for computing $<$.

Now we allow having several same intervals in the representation $\mathcal{R}$ since the bounds might force this situation. The representation $\mathcal{R}$ then gives a linear pre-ordering $\unlhd$. When we construct bounded representations, we place intervals as the same if and only if this is forced by the bounds. It is easy to observe that if $I_u = I_v$, then the vertices u and v are indistinguishable.

Constraints Given by Bounds. In the case of bounded representations, the order of some pairs of the indistinguishable vertices can be prescribed by the bounds. Suppose that we restrict ourself to just a single component C of the input graph G and ignore the rest. Similarly to above, we produce a relation $\lhd'$ of the vertices of C.

Let u and v be two indistinguishable vertices of C. We put $u \unlhd' v$ if and only if $\mathfrak{L}_u \lessdot \mathfrak{L}_v$ or $\mathfrak{R}_u \lessdot \mathfrak{R}_v$; so $\unlhd'$ is a union of the subset order $\lessdot_\ell$ of the left bounds and the subset order $\lessdot_r$ of the right bounds. Notice that the pre-ordering $\unlhd'$ does not have to be a partial ordering and that $u \unlhd' v$ implies $u \unlhd v$ for any representation $\mathcal{R}$.

Now, since we do not want to work with pre-ordering $\unlhd$, we construct a reduced graph C' with modified bounds. The following proposition states that this construction does not change solution of the problem.

Proposition 3. *There exists a bounded representation of C with an ordering extending $<$ if and only if there exists a bounded representation of C' in an ordering $\lhd$ which extends both $<$ and $\lhd'$.*

Proof (Sketch). The construction of C' is done in two steps. First, we consider strongly connected components defined by $\unlhd'$, and they have to be represented by the same intervals. Therefore, we unify the bounds of their intervals to force this.

To prove the correctness of the reduction, we reorder groups in C according to $\lhd$. For each group, we apply a similar greedy procedure as in Proposition 1. For the detailed proof see Appendix C. $\square$

4.3 The Algorithm

The algorithm works as follows:

(1) We compute the ordering $\blacktriangleleft'$ of components, and construct a linear ordering $\blacktriangleleft$ extending $\blacktriangleleft'$.
(2) We proceed the components according to $\blacktriangleleft$ from left to right: $C_1 \blacktriangleleft \cdots \blacktriangleleft C_c$.
(3) When processing the component C_i:
  - Compute the partial ordering $<$, using [5].
  - For $<$ and its reversal do the following: for each group Γ of indistinguishable vertices, compute $\lhd'$, its strongly connected components, the reduced graph C_i' and its ordering $\lhd$.
  - Place the endpoints according to $\lhd$ from left to right, on the right side of the representation of C_{i-1} greedily as far to the left as possible.
  - Construct a representation of C_i, by copying the intervals I_{S_i}.

It remains to argue details concerning specific implementation and correctness which is easily implied by Proposition 2 and Proposition 3. See Appendix D for details.

5 Conclusions

In this paper, we give a polynomial time algorithm for the classes of interval and proper interval graphs for a recently introduced problem BoundRep. The main result of this paper is a rather surprising discovery that the bounded representation problem distinguishes the classes of proper and unit interval graphs: BoundRep(PROPER INT) is polynomially solvable but BoundRep(UNIT INT) is NP-complete [13]. We believe that is a very interesting problem to further investigate differences between the structures of proper and unit interval representations; this paper gives a good reason to do so.

Open Problems. We conclude with two open problems.

Problem 1. Is it possible to solve BoundRep(PROPER INT) in time $\mathcal{O}(n+m)$ (with a given left-to-right ordering of the bounds)?

The current bottleneck of our algorithm is the computation of $\unlhd$ from $\unlhd'$ which is the only step requiring time $\mathcal{O}(n^2)$.

Problem 2. What is the complexity of the BoundRep problem for other classes such as circular-arc graphs, circle graphs?

Currently, the only known results are for the classes INT, PROPER INT, and UNIT INT. Even attacking some simpler problems for these classes might be very interesting. For instance, solving the partial representation extension problem for circular-arc graphs could be a major advancement in the area of the restricted representation problems.

References

1. Benzer, S.: On the topology of the genetic fine structure. Proc. Nat. Acad. Sci. U.S.A. 45, 1607–1620 (1959)
2. Bläsius, T., Rutter, I.: Simultaneous PQ-ordering with applications to constrained embedding problems. In: SODA '13 (2013)
3. Booth, K.S., Lueker, G.S.: Testing for the consecutive ones property, interval graphs, and planarity using PQ-tree algorithms. J. Comput. System Sci. 13, 335–379 (1976)
4. Chaplick, S., Fulek, R., Klavík, P.: Extending partial representations of circle graphs. Accepted to GD. (2013)
5. Corneil, D.G., Kim, H., Natarajan, S., Olariu, S., Sprague, A.P.: Simple linear time recognition of unit interval graphs. Inf.Process.Lett 55(2), 99–104 (1995)
6. Deng, X., Hell, P., Huang, J.: Linear-time representation algorithms for proper circular-arc graphs and proper interval graphs. SIAM J. Comput. 25(2), 390–403 (1996)
7. Dorbec, P., Kratochvíl, J., Montassier, M.: Contact representations of planar graph: Rebuilding is hard. Submitted. (2013)
8. Fulkerson, D.R., Gross, O.A.: Incidence matrices and interval graphs. Pac. J. Math. 15, 835–855 (1965)
9. Gilmore, P.C., Hoffman, A.J.: A characterization of comparability graphs and of interval graphs. Can. J. Math. 16, 539–548 (1964)
10. Hajós, G.: Über eine Art von Graphen. Internationale Mathematische Nachrichten 11, 65 (1957)
11. Jampani, K., Lubiw, A.: Simultaneous interval graphs. In: Algorithms and Computation, LNCS, vol. 6506, pp. 206–217 (2010)
12. Klavík, P., Kratochvíl, J., Krawczyk, T., Walczak, B.: Extending partial representations of function graphs and permutation graphs. In: Algorithms – ESA 2012. Lecture Notes in Computer Science, vol. 7501, pp. 671–682 (2012)
13. Klavík, P., Kratochvíl, J., Otachi, Y., Rutter, I., Saitoh, T., Saumell, M., Vyskočil, T.: Extending partial representations of proper and unit interval graphs. In preparation. (2012)
14. Klavík, P., Kratochvíl, J., Otachi, Y., Saitoh, T.: Extending partial representations of subclasses of chordal graphs. In: Algorithms and Computation – ISAAC. Lecture Notes in Computer Science, vol. 7676, pp. 444–454 (2012)
15. Klavík, P., Kratochvíl, J., Otachi, Y., Saitoh, T., Vyskočil, T.: Linear-time algorithm for partial representation extension of interval graphs. CoRR abs/1306.2182 (2013)
16. Klavík, P., Kratochvíl, J., Vyskočil, T.: Extending partial representations of interval graphs. In: Theory and Applications of Models of Computation - 8th Annual Conference, TAMC 2011. LNCS, vol. 6648, pp. 276–285 (2011)
17. Kobler, J., Kuhnert, S., Watanabe, O.: Interval graph representation with given interval and intersection lengths. In: Algorithms and Computation, Lecture Notes in Computer Science, vol. 7676, pp. 517–526 (2012)
18. Roberts, F.S.: Indifference graphs. Proof techniques in graph theory pp. 139–146 (1969)

19. Rose, D.J., Tarjan, R.E., Lueker, G.S.: Algorithmic aspects of vertex elimination on graphs. SICOMP 5(2), 266–283 (1976)

A Appendix: Proof of Proposition 1

Let G be a given graph. For a maximal clique $C \subseteq G$, we set

$$J_C = \left(\bigcap_{u \in V(C)} [\ell(\mathfrak{L}_u), r(\mathfrak{R}_u)] \right) \setminus \left(\bigcup_{v \notin V(C)} [r(\mathfrak{L}_v), \ell(\mathfrak{R}_v)] \right)$$

i to denote the set of all the possible candidates for clique points $\mathrm{cp}(C)$ with respect to the given restrictions on the intervals of the representation. (Where the interval $[r(\mathfrak{L}_v), \ell(\mathfrak{R}_v)]$ is empty if $r(\mathfrak{L}_v) > \ell(\mathfrak{R}_v)$.) If the set J_C is empty, then it is clearly not possible to place $\mathrm{cp}(C)$ and thus there is no interval representation of G which satisfies the given restrictions.

We now prove Proposition 1 which says that there exists a bounded representation $\mathcal{R}$ if and only if there is an ordering $<$ of maximal cliques which is consecutive in every vertex v and extends the subset ordering $\lessdot$ on the sets J_C (which can be also understood as a partial ordering of the maximal cliques).

Proof (Proposition 1). If there is no such extension $<$, then using Lemma 1 we see that G does not have an interval representation which satisfies the bounds, because the conditions forced by $\lessdot$ are clearly necessary.

On the other hand, suppose that G is a graph which has an interval representation satisfying the bounds. Then according to Lemma 1 there is a reordering $<$ which extends $\lessdot$. To construct an interval representation $\mathcal{R}$ of G we first greedily place the clique points according to $<$ from left to right always as far to the left as possible.

All that is left is to show that the greedy procedure cannot fail. Assume, for the sake of contradiction, that the procedure fails for the clique point $\mathrm{cp}(C)$. Since $\mathrm{cp}(C)$ cannot be placed at all, there are some clique points placed on the right of $r(J_C)$ (or possibly on $r(J_C)$). Let $\mathrm{cp}(B)$ be the leftmost of them. Since $\mathrm{cp}(B)$ was placed before $\mathrm{cp}(C)$ we have $B < C$ and thus $C \lessdot B$ cannot hold. Therefore we know that $\ell(J_B) < r(J_C)$.

Following the greedy procedure we see that $\mathrm{cp}(B)$ was not placed to the left of $r(J_C)$, because all the possible locations there were blocked by previously placed clique points or by intervals $[r(\mathfrak{L}_v), \ell(\mathfrak{R}_v)]$ where $v \notin V(B)$. There is at least one clique point placed to the right of $\ell(J_B)$, because otherwise we would place $\mathrm{cp}(B)$ to $\ell(J_B)$ or right next to it. Let $\mathrm{cp}(A)$ be the rightmost clique-point placed between $\ell(J_B)$ and $\mathrm{cp}(B)$.

We claim that every point between $\mathrm{cp}(A)$ and $r(J_C)$ has to be covered by intervals $[r(\mathfrak{L}_v), \ell(\mathfrak{R}_v)]$ where $v \notin V(B)$. Otherwise we would place $\mathrm{cp}(B)$ to the uncovered point, which is in J_B, since $\mathrm{cp}(A)$ is between $\ell(J_B)$ and $\mathrm{cp}(B)$. Let S be the set of such intervals and let $\mathcal{C}$ be the set of maximal cliques containing at least one vertex from S. Note that since S induces a connected subgraph of G, all the cliques in $\mathcal{C}$ appear consecutively in $<$, because every pair of adjacent vertices from S is contained in some maximal clique from $\mathcal{C}$.

From the assumptions made, it is clear that A and C are in $\mathcal{C}$, but B is not and $A < B < C$ holds. However consecutiveness of $\mathcal{C}$ and $A < B$ imply $C < B$ which gives us a contradiction. $\square$

All that is left is to describe the construction of a suitable interval representation of G provided that the partial ordering $\lessdot$ was extended to $<$ in the third step of the algorithm.

To obtain such representation $\mathcal{R}$ it suffices to let

$$\ell(I_v) = \min\left(\{r(\mathfrak{L}_v)\} \cup \{\mathrm{cp}(C) \mid v \in V(C)\}\right)$$

and

$$r(I_v) = \max\left(\{\ell(\mathfrak{R}_v)\} \cup \{\mathrm{cp}(C) \mid v \in V(C)\}\right)$$

for every vertex v of G.

Then from the fact that $\mathrm{cp}(C) \in J_C$ we have $\ell(I_v) \in \mathfrak{L}_v$ and $r(I_v) \in \mathfrak{R}_v$. Therefore the restrictions on the representing intervals are satisfied. Moreover $\mathcal{R}$ is an interval representation of G, since every clique-point lies exactly in the intervals representing the vertices of the corresponding maximal clique.

B Appendix: Proof of Proposition 2

In this part we prove that a bounded representation of G in an ordering $\blacktriangleleft$ extending $\blacktriangleleft'$ exists if and only if there exists a bounded representation of G.

An interval I is called *trivial* if $\ell(I) = r(I)$ and *non-trivial* otherwise. Two non-trivial intervals I and J are *intersecting non-trivially*, if $I \cap J$ is non-trivial. If I is trivial and J is non-trivial, then $I \cap J$ is *non-trivial* if $\ell(J) < \ell(I) = r(I) < r(J)$.

Before proving Proposition 2, we need to establish some basic properties. For the following, let C and C' be two incomparable components in $\blacktriangleleft'$. Let $\mathcal{B}$ be the collection of all bounds of intervals in C, and similarly $\mathcal{B}'$ for C'.

Lemma 3. *Every bound $B \in \mathcal{B}$ non-trivially intersects every bound $B' \in \mathcal{B}'$.*

Proof. If B is $\mathfrak{L}_u$ and B' is $\mathfrak{R}_v$, or vice versa, then the statement clearly holds; if B and B' would not intersect, or would intersect trivially, we get that $\mathfrak{L}_u \lessdot \mathfrak{R}_v$ and $C \blacktriangleleft' C'$. Now, say we have two left bounds $B = \mathfrak{L}_u$ and $B' = \mathfrak{L}_v$, and suppose for contradiction $\mathfrak{L}_u \lessdot \mathfrak{L}_v$. Since $\ell(\mathfrak{L}_v) \le \ell(\mathfrak{R}_v)$, we get that also $\mathfrak{L}_u \lessdot \mathfrak{R}_v$ and so $C \blacktriangleleft' C'$, contradiction. We proceed similarly for the two right bounds. $\square$

Lemma 4. *If the bounds $\mathcal{B}$ are not pairwise non-trivially intersecting, then the bounds $\mathcal{B}'$ are pairwise non-trivially intersecting.*

Proof. Let $\mathfrak{L}_u$ be the leftmost left bound of C (minimizing $r(\mathfrak{L}_u)$) and let $\mathfrak{R}_v$ be the rightmost right bound of C (maximizing $\ell(\mathfrak{R}_v)$). We know that $\mathfrak{L}_u \lessdot \mathfrak{R}_v$. If $B' \in \mathcal{B}'$, then according to Lemma 3 we have $\ell(B') < r(\mathfrak{L}_u) \le \ell(\mathfrak{R}_v) < r(B')$. So all bounds of $\mathcal{B}'$ are intersecting non-trivially. $\square$

Lemma 5. *Suppose that there exists a bounded representation $\mathcal{R}$ which places C and C' next to each other in $\blacktriangleleft$. Then there exists another bounded representation $\mathcal{R}'$ with the only difference that the order of C and C' is swapped.*

Proof. We represent the remaining components in $\mathcal{R}'$ exactly as in $\mathcal{R}$, so we only need to deal with C and C'. We can assume that $C \blacktriangleleft C'$ in $\mathcal{R}$. Let x be the rightmost endpoint of the component on the left of C, and let y be the leftmost endpoint of the component on the right of C. (Or $-\infty$, resp. ∞, if such an endpoint does not exist.)

There are three possible cases how the bounds $\mathcal{B}$ and $\mathcal{B}'$ are intersecting.

- **Case 1:** The bounds $\mathcal{B}$ are pairwise non-trivially intersecting and the bounds $\mathcal{B}'$ are as well. Then by Lemma 3 all bounds of $\mathcal{B} \cup \mathcal{B}'$ are pairwise non-trivially intersecting. So by the Helly property, there exists a non-trivial interval J contained in all bounds $\mathcal{B} \cup \mathcal{B}'$ such that $x < \ell(J) < r(J) < y$. Since $\mathcal{R}$ exists, there exists some ordering of the endpoints in which the components C and C' are representable. We can represent in $\mathcal{R}'$ the components C and C' in this ordering inside J which gives a correct bounded representation of G.
- **Case 2:** The bounds $\mathcal{B}$ are not pairwise non-trivially intersecting and the bounds $\mathcal{B}'$ are. Let $\mathfrak{L}_u$ and $\mathfrak{R}_v$ be the same as in the proof of Lemma 4. Also, in the proof we argued that there exists a non-trivial interval J contained in every bound of $\mathcal{B}'$ such that $\ell(J) < r(\mathfrak{L}_u) \le \ell(\mathfrak{R}_v) < r(J)$. So it is possible to place the entire representation of C' strictly in between of x and $r(\mathfrak{L}_u)$, in the same ordering as in $\mathcal{R}$. And we represent C on the right of it, again in the same ordering. (This is possible since each bound ends on the right of $r(\mathfrak{L}_u)$.)
- **Case 3:** The same as above, but we represent C on the right of J, and we compress C' on the left of it.

For the cases 1 to 3, we construct a correct bounded representation $\mathcal{R}'$ in the required ordering $\blacktriangleleft$. Lemma 4 states that Case 4 in which both $\mathcal{B}$ and $\mathcal{B}'$ are not pairwise non-trivially intersecting cannot occur. ◻

Proof (Proposition 2). The implication from left to right is obvious, $\mathcal{R}$ is one of the bounded representations of G. For the other implication, let $\overline{\blacktriangleleft}$ be the ordering of the components in $\overline{\mathcal{R}}$, and modify $\overline{\mathcal{R}}$ by repeated application of Lemma 5 to make it in the ordering $C_1 \blacktriangleleft \cdots \blacktriangleleft C_c$. Now suppose that $\overline{\mathcal{R}}$ starts by $C_1, \dots, C_{i-1}$ but the following is not C_i as in $\blacktriangleleft$. We take C_i in $\overline{\blacktriangleleft}$ and by repeated application of Lemma 5, we shift it next to C_{i-1}. To do so, we need to show that C_i is incomparable to all components between C_{i-1} and C_i in $\overline{\blacktriangleleft}$. But if there would be C_j which would be comparable, then $C_j \blacktriangleleft' C_i$, and so the ordering $\blacktriangleleft$ would not extend $\blacktriangleleft'$. ◻

C Appendix: Proof of Proposition 3

Let us remind the statement of Proposition 3. It says that there exists a bounded representation of C with an ordering extending $<$ if and only if there exists a bounded representation of C' in an ordering $\lhd$ which extends both $<$ and $\lhd'$. Before the proof of this statement we need to introduce some notation.

Reduced Graph. We have a pre-ordering $\unlhd'$, and some intervals might be forced to be represented the same. We are going to construct a reduced graph C' such that it can be represented in a strict ordering $\lhd$. We consider one group Γ of indistinguishable vertices and use the relation $\lhd'$ to define an oriented graph H on Γ; we put $(u,v) \in E(H)$ if and only if $u \lhd' v$. Let S be a strongly connected component of H. Then its vertices have to be represented by the same interval in $\mathcal{R}$. The strongly connected components are partially ordered by the remaining edges of H going in between of them, let $S_1 < \cdots < S_k$ be an arbitrary topological sorting. Then the constructed pre-ordering $\lhd$ orders the vertices of Γ:

$$S_1 \lhd S_2 \lhd \cdots \lhd S_k,$$

where all vertices of each S_i are equal in $\unlhd$.

Let C' be the contracted graph, in which vertices of each group are replaced by the vertices $S_1, \dots, S_k$. The idea is to force the vertices of each S_i to be represented by the same interval. We are going to represent the entire S_i by a single interval I_{S_i}. To do that we need to define bounds compatible with all vertices of S_i, and so we put

$$\mathfrak{L}_{S_i} = \bigcap_{u \in S_i} \mathfrak{L}_u, \qquad \text{and} \qquad \mathfrak{R}_{S_i} = \bigcap_{u \in S_i} \mathfrak{R}_u.$$

Then the constructed pre-ordering $\unlhd$ becomes a linear ordering $\lhd$ for C', so the representation of C' has pairwise distinct intervals. So in the constructed representation of C', we can replace the interval I_{S_i} with several equal intervals, representing the vertices of S_i, and we obtain a correct bounded representation of C.

Group Reordering. Before proving this Proposition 3, we state one important property. Let Γ be a group of indistinguishable vertices and let $\mathcal{R}$ be a representation with the ordering $\lhd$. Notice that the vertices of Γ appear consecutively in $\lhd$. We study which points of the real line are taken by all intervals of Γ and which by some interval of Γ. We define

$$\cap\Gamma = \bigcap_{v \in \Gamma} I_v, \qquad \text{and} \qquad \cup\Gamma = \bigcup_{v \in \Gamma} I_v.$$

Since intervals of the real line satisfy the Helly property and Γ forms a clique in G, we get that both $\cap\Gamma$ and $\cup\Gamma$ are closed intervals such that $\cap\Gamma \subseteq \cup\Gamma$.

Lemma 6. *If u is adjacent to Γ, then I_u intersects $\cap\Gamma$. And if u is non-adjacent to Γ, then $I_u \cap \cup\Gamma = \emptyset$.*

Proof. Since Γ is a group of indistinguishable vertices, then u is either adjacent to everything or nothing, the rest easily follows. $\square$

Let $\varGamma$ and $\varGamma'$ be two groups. If the vertices in these groups are pairwise adjacent, we get that $\cap\varGamma$ intersects $\cap\varGamma'$, and if they are pairwise non-adjacent, we get that $\cup\varGamma \cap \cup\varGamma'$ is empty.

Lemma 7. *Let $\mathcal{R}$ be a representation of G and let $\varGamma$ be a group of indistinguishable vertices. Let $\mathcal{R}'$ be a representation constructed from $\mathcal{R}$ by placing intervals I'_u of $\varGamma$ such that $\cap\varGamma \subseteq I'_u \subseteq \cup\varGamma$ with the same ordering of the left and right endpoints. Then $\mathcal{R}'$ is a correct proper interval representation of G.*

Proof. The representation $\mathcal{R}'$ is proper, and it is easy to see that intersections are preserved. □

Proof (Proposition 3). Again, one direction is clear. We can construct representation of C as above by copying the intervals I_{S_i}. For the other direction, let $\lhd$ be an ordering from the statement and let $\mathcal{R}$ be a representation of C. We proceed the groups of C in an arbitrary order and construct a representation of C' according to $\lhd$ using Lemma 7.

For a group $\varGamma$ with the vertices $S_1 \lhd \cdots \lhd S_k$, we want to place the intervals I_{S_i} in between of $\cap\varGamma$ and $\cup\varGamma$ in this order. Let ℓ_i (resp. r_i) be the left (resp. right) endpoint of I_{S_i}. We want to place the left endpoints in $[\ell(\cup\varGamma), \ell(\cap\varGamma)]$ and the right endpoint in $[r(\cap\varGamma), r(\cup\varGamma)]$ both according to the ordering $\lhd$, and we can do this independently.

We consider the following only for the left endpoints $\ell_1, \dots, \ell_k$ and we place the right endpoints $r_1, \dots, r_k$ in $[r(\cap\varGamma), r(\cup\varGamma)]$ using exacly the same argument. Let $\mathfrak{L}_1, \dots, \mathfrak{L}_k$ be the left bounds of $S_1, \dots, S_k$. We assume that each $\mathfrak{L}_i$ is a subset of $[\ell(\cup\varGamma), \ell(\cap\varGamma)]$; otherwise we consider only the intersections of the bounds with this interval. (Notice that even for these restricted bounds we get the constraints $\lhd'$.) We place the left endpoints in $[\ell(\cup\varGamma), \ell(\cap\varGamma)]$ greedily from left to right from ℓ_1 to ℓ_k while respecting their bounds.

Suppose that this placing procedure fails when we attempt to place ℓ_i. Then we show that there is a contradiction with $\lhd$ extending $\lhd'$. Since we successfully placed ℓ_{i-1} and we cannot place ℓ_i, we have $r(\mathfrak{L}_i) \le \ell_{i-1}$. The reason why ℓ_{i-1} was not placed more to the left by the greedy algorithm is that this position is blocked by some previously placed left endpoint. Let ℓ_j be the leftmost placed endpoint such that $r(\mathfrak{L}_i) \le \ell_j$. Since ℓ_j was placed as far to the left as possible and the position immediately on the left of $r(\mathfrak{L}_i)$ is not blocked by any other left endpoint, we get $r(\mathfrak{L}_i) \le \ell(\mathfrak{L}_j)$.

It remains to relate this to the original bounds of C, and to show that it contradicts the definition of $\unlhd$. Let $u \in S_i$ be a vertex such that $r(\mathfrak{L}_u)$ is minimal and choose $v \in S_j$ such that $\ell(\mathfrak{L}_v)$ is maximal. Then we get $r(\mathfrak{L}_u) \le \ell(\mathfrak{L}_v)$, and so $u \lhd' v$. This contradicts the constructed topological sort which places S_j before S_i, since there is an edge going from S_i to S_j. □

D Appendix: Proof of Theorem 2

In this part the reader can find details of the proof of Theorem 2, which says that the problem BoundRep(PROPER INT) can be solved in time $\mathcal{O}(n^2)$ where n is the number of vertices.

Lemma 8. *A linear ordering $\blacktriangleleft$ extending $\blacktriangleleft'$ can be computed in time $\mathcal{O}(n+m)$.*

Proof. Let $C_1, \ldots, C_c$ be the components of G. We define for each component two numbers called a *lower handle* and an *upper handle*:

$$\mathrm{LH}(C_i) = \min\{r(\mathfrak{L}_v) : v \in V(C_i)\}, \quad \text{and} \quad \mathrm{UH}(C_i) = \max\{\ell(\mathfrak{R}_v) : v \in V(C_i)\}.$$

In this setting, we get that $C_i \blacktriangleleft' C_j$ if and only if $\mathrm{LH}(C_i) \le \mathrm{UH}(C_j)$.

It is proved in [15, Section 2.2] that one can find a linear ordering $\blacktriangleleft$ extending $\blacktriangleleft'$ using these handles. We first compute a linear ordering $\prec$ of all the handles from left to right, and to deal with ties we first place the lower handles in any order and then the upper handles in any order. Since the endpoints of the bounds are given sorted, we can compute this using a single sweep from left to right.

Now we use this ordering and construct $\blacktriangleleft$ by repeated finding of minimal elements. We look at the first element in $\prec$. If it is a lower handle $\mathrm{LH}(C_i)$, then there cannot be any other lower handle $\mathrm{LH}(C_j)$ such that $\mathrm{LH}(C_j) \prec \mathrm{UH}(C_i)$, and then C_i is a minimal element. And if the first element is an upper handle $\mathrm{UH}(C_i)$, then C_i is a minimal element. If there is no minimal element, the algorithm fails. If there is some minimal element C_i, we append it to the constructed $\blacktriangleleft$ and remove both handles of C_i from $\prec$. In total, this algorithm can be implemented in time $\mathcal{O}(n + m)$. For details and a proof of correctness, see [15]. $\square$

Lemma 9. *For a partial ordering $<$ (resp. its reversal) of [5], we can compute any $\lhd$ extending $<$ (resp. its reversal) and $\lhd'$ in time $\mathcal{O}(n^2)$.*

Proof. We argue only for $<$, for reversal the argument is the same. We proceed separately for each group Γ, for which $<$ gives no restriction. We represent the constraints given by $\lhd'$ as an oriented graph H over Γ, as described above. We start by finding strongly connected components $S_1, \ldots, S_k$. Then we contract the strongly connected components, and find any topological sorting of the contracted graph. Everything can be done in linear time with respect to the size of the graph H which is $\mathcal{O}(n^2)$. $\square$

Consider all representations of the reduced graph C_i' in an ordering $\lhd$. We call a bounded representation $\mathcal{R}$ of C_i' in the ordering $\lhd$ the *left-most representation*, if it minimizes the right-most endpoint of C_i' over all bounded representations $\mathcal{R}'$ of C_i'. To be more precise, we denote the right-most endpoint of the component C_i' by $r(C_i')$ in $\mathcal{R}$ and by $r'(C_i')$ in $\mathcal{R}'$. Then $\mathcal{R}$ is the left-most representation if $r(C_i') \le r'(C_i')$ for every bounded representation of $\mathcal{R}'$. Since we are working with left-to-right orderings instead of precise rational positions, we just require that for any bound x such that $r'(C_i') \le x$, we also have $r(C_i') \le x$.

Lemma 10. *If there exists a representation of the reduced graph C_i' in an ordering $\lhd$, then the left-most representation exists, it is unique and it can be computed in time $\mathcal{O}(n)$.*

Proof. For a given ordering $\lhd$, there are only finitely many ways how one can insert the bounds into the left-to-right ordering of the endpoints of the bounds;

so there are finitely many different bounded representations. We consider the structure $\mathfrak{Rep}$ of all bounded representations in the ordering $\lhd$, and we know that there is at least one such representation. We are going to show that they form a meet semilattice with infimum operation defined as follows. Then the left-most representation corresponds to the infimum of the entire semilattice, and thus we know it always exists and it is unique.

For two representations $\mathcal{R}, \mathcal{R}' \in \mathfrak{Rep}$, we define the infimum $\mathcal{R} \wedge \mathcal{R}'$ as the representation $\overline{\mathcal{R}}$ such that $\overline{\ell}_i = \min\{\ell_i, \ell'_i\}$ and $\overline{r}_i = \min\{r_i, r'_i\}$. In other words, the lattice is defined by the ordering $\leq$ in which $\mathcal{R} \leq \mathcal{R}'$ if and only if it is less or equal in each endpoint, and then this is the natural way to define infimum.

It is quite clear that $\overline{\mathcal{R}}$ satisfies the bounds. Also, it is straightforward to check that $\overline{\mathcal{R}}$ is a correct proper interval representation of C'_i; for each vertex we have two possible orderings $\ell\ell rr$ or $\ell r\ell r$ if we place $\mathcal{R}$ together with $\mathcal{R}'$. So for a pair of vertices u and v, we have four possibilities in total and we just need to check that the intersections are preserved for all of them.

We can construct the left-most representation in time $\mathcal{O}(n)$ as follows. First, we compute a common ordering of the left and right endpoints from left to right. We know from $\trianglelefteq$ that $\ell_1 \leq \cdots \leq \ell_n$. Into these orderings, we insert right endpoints one-by-one, and we insert r_i right before the left endpoint ℓ_j where v_j is the smallest non-neighbor of v_i such that $v_i \lhd v_j$. This ordering is uniquely determined by $\trianglelefteq$ and any representation constructed in this ordering is a correct representation of G.

Now, we process the endpoints from left to right and we always place them greedily as far to the left as possible. Doing so, we respect the bounds, so ℓ_i is inserted only in $\mathfrak{L}_{v_i}$ and r_i only in $\mathfrak{R}_{v_i}$. If we insert a right endpoint r_i, and the previously inserted endpoint was ℓ_j, then we put $r_i = \min\{\ell_j, \ell(\mathfrak{R}_i)\}$. Otherwise, we insert the endpoint x strictly on the right of the previously inserted endpoint y such that $x \geq \ell(B)$ where B is the bound for x; so we might insert it immediately next to y, or at the position $\ell(B)$ if the bound is further to the right.

Since the representation is constructed according to the common ordering, it is a correct proper interval representation of G. We prove by induction that the representation is the left-most representation, and thus it satisfies the right endpoints of the bounds. (The left endpoints are clearly satisfied by the construction.) The first endpoint is placed directly on the bound, so it is clearly the left-most. Now suppose that after placing an endpoint x the representation is not the left-most anymore, but it was the left-most right before placing the endpoint x. But this is not possible since we clearly place x as far to the left as possible, while satisfying the common ordering and the left endpoints of the bounds. $\square$

Proof (Theorem 2). We are ready to prove the main theorem. Clearly, if no bounded representation exists, the algorithm has to fail in some step and does not construct it. On the other hand, suppose that some convenient representation exists. First, we compute the ordering $\blacktriangleleft$ extending $\blacktriangleleft'$ using Lemma 8. According to Proposition 2, if there exists a bounded representation of G, then there exists a bounded representation in this ordering $\blacktriangleleft$. If $\mathcal{R}$ is a such representation, then the algorithm constructs another bounded representation $\mathcal{R}'$.

We are going to show by induction that $r'(C_i) \le r(C_i)$ for each component C_i. (Since we are working in the context of left-to-right orderings, this condition just means that for any endpoint x of any bound, if $r(C_i) \le x$, then also $r'(C_i) \le x$.) This condition says that the representation $\mathcal{R}'$ of C_i leaves at least as much space as C_i in $\mathcal{R}$ for the remaining components.

The first induction step is clearly satisfied for non-existent component C_0. (This means that we can place C_1 in both $\mathcal{R}'$ and $\mathcal{R}$ arbitrarily to the left.) We proceed the components from left to right. Let C_i be the processed component, so we have represented $C_1, \dots, C_{i-1}$ in $\mathcal{R}'$, and the induction hypothesis states that $r'(C_{i-1}) \le r(C_i)$. We test for C_i both $<$ and its reversal and choose the representation which minimizes $r'(C_i)$.

The component C_i is represented in $\mathcal{R}$ in an ordering extending one of these orderings, without loss of generality $<$. Using Lemma 9, we can compute in time $\mathcal{O}(n^2)$ the reduced graph C_i' and some other ordering $\lhd$. According to Proposition 3, there exists a representation $\overline{\mathcal{R}}$ of C_i', and by copying the intervals I_{S_i}, we obtain a representation $\overline{\mathcal{R}}$ of C_i. Since we represent each group $\mathit{\Gamma}$ in between of $\cap \mathit{\Gamma}$ and $\cup \mathit{\Gamma}$, the representation $\overline{\mathcal{R}}$ of C_i is at most as large as the representation $\mathcal{R}$, and thus $\overline{r}(C_i) \le r(C_i)$.

The proof of Proposition 3 is just existential, since it requires a representation $\mathcal{R}$. However using Lemma 10, we can construct the left-most representation $\mathcal{R}'$ of C_i' which satisfies $r'(C_i') \le \overline{r}(C_i') = \overline{r}(C_i)$. So by copying the intervals I_{S_i}, we obtain a representation $\mathcal{R}'$ of C_i which satisfies $r'(C_i) = r'(C_i') \le r(C_i)$ as required. Since at least for $<$ or its reversal we obtain a representation $\mathcal{R}'$ satisfying $r'(C_i) \le r(C_i)$, the induction step is correct and we construct a correct bounded representation of G if it exists. $\square$