THE RECOGNITION OF SIMPLE-TRIANGLE GRAPHS AND OF LINEAR-INTERVAL ORDERS IS POLYNOMIAL*

GEORGE B. MERTZIOS†

Abstract. Intersection graphs of geometric objects have been extensively studied, both due to their interesting structure and their numerous applications; prominent examples include interval graphs and permutation graphs. In this paper we study a natural graph class that generalizes both interval and permutation graphs, namely *simple-triangle* graphs. Simple-triangle graphs – also known as *PI* graphs (for Point-Interval) – are the intersection graphs of triangles that are defined by a point on a line L_1 and an interval on a parallel line L_2. They lie naturally between permutation and trapezoid graphs, which are the intersection graphs of line segments between L_1 and L_2 and of trapezoids between L_1 and L_2, respectively. Although various efficient recognition algorithms for permutation and trapezoid graphs are well known to exist, the recognition of simple-triangle graphs has remained an open problem since their introduction by Corneil and Kamula three decades ago. In this paper we resolve this problem by proving that simple-triangle graphs can be recognized in polynomial time. Given a graph G with n vertices, such that its complement $\overline{G}$ has m edges, our algorithm runs in $O(n^2m)$ time. As a consequence, our algorithm also solves a longstanding open problem in the area of partial orders, namely the recognition of *linear-interval orders*, i.e. of partial orders $P = P_1 \cap P_2$, where P_1 is a linear order and P_2 is an interval order. This is one of the first results on recognizing partial orders P that are the intersection of orders from two different classes $\mathcal{P}_1$ and $\mathcal{P}_2$. In complete contrast to this, partial orders P which are the intersection of orders from the same class $\mathcal{P}$ have been extensively investigated, and in most cases the complexity status of these recognition problems has been already established.

Keywords: Intersection graph, PI graph, recognition problem, partial order, polynomial algorithm.

AMS subject classification (2010). Primary 05C85; Secondary 06A07, 05C62, 68R10.

1. Introduction.

A graph G is the *intersection graph* of a family $\mathcal{F}$ of sets if we can bijectively assign sets of $\mathcal{F}$ to vertices of G such that two vertices of G are adjacent if and only if the corresponding sets have a non-empty intersection. It turns out that many graph classes with important applications can be described as intersection graphs of set families that are derived from some kind of geometric configuration. One of the most prominent examples is that of *interval* graphs, i.e. the intersection graphs of intervals on the real line, which have natural applications in several fields, including bioinformatics and involving the physical mapping of DNA and the genome reconstruction[1] [4, 9, 10].

Generalizing the intersections on the real line, consider two parallel horizontal lines on the plane, L_1 (the upper line) and L_2 (the lower line). A graph G is a *simple-triangle* graph if it is the intersection graph of triangles that have one endpoint on L_1 and the other two on L_2. Furthermore, G is a *triangle* graph if it is the intersection graph of triangles with endpoints on L_1 and L_2, but now there is no restriction on which line contains one endpoint of every triangle and which contains the other two. Simple-triangle and triangle graphs are also known as *PI* and *PI** graphs, respectively [3, 6, 22], where PI stands for "Point-Interval" . Such representations of simple-triangle and of triangle graphs are called *simple-triangle* (or *PI*) and *triangle*

*This work was partially supported by the EPSRC Grant EP/K022660/1. A preliminary conference version of this work appeared in the *Proceedings of the 21st European Symposium on Algorithms (ESA)*, Sophia Antipolis, France, pages 719–730, 2013.

†School of Engineering and Computing Sciences, Durham University, United Kingdom. Email: george.mertzios@durham.ac.uk

[1] Benzer [1] earned the prestigious Lasker Award (1971) and Crafoord Prize (1993) partly for showing that the set of intersections of a large number of fragments of genetic material in a virus form an interval graph.

(or *PI**) *representations*, respectively. Simple-triangle and triangle graphs lie naturally between *permutation* graphs (i.e. the intersection graphs of line segments with one endpoint on L_1 and one on L_2) and *trapezoid* graphs (i.e. the intersection graphs of trapezoids with one interval on L_1 and the opposite interval on L_2) [3, 22]. Note that, using the notation *PI* for simple-triangle graphs, permutation graphs are *PP* (for "Point-Point") graphs, while trapezoid graphs are *II* (for "Interval-Interval") graphs [6].

A *partial order* is a pair $P = (U, R)$, where U is a finite set and R is an irreflexive transitive binary relation on U. Whenever $(x, y) \in R$ for two elements $x, y \in U$, we write $x <_P y$. If $x <_P y$ or $y <_P x$, then x and y are *comparable*, otherwise they are *incomparable*. P is a *linear order* if every pair of elements in U are comparable. Furthermore, P is an *interval order* if each element $x \in U$ is assigned to an interval I_x on the real line such that $x <_P y$ if and only if I_x lies completely to the left of I_y. One of the most fundamental notions on partial orders is *dimension*. For any partial order P and any class $\mathcal{P}$ of partial orders (e.g. linear order, interval order, semiorder, etc.), the $\mathcal{P}$*-dimension* of P is the smallest k such that P is the intersection of k orders from $\mathcal{P}$. In particular, when $\mathcal{P}$ is the class of linear orders, the $\mathcal{P}$-dimension of P is known as the *dimension* of P. Although in most cases we can efficiently recognize whether a partial order belongs to a class $\mathcal{P}$, this is not the case for higher dimensions. Due to a classical result of Yannakakis [23], it is NP-complete to decide whether the dimension, or the interval dimension, of a partial order is at most k, where $k \geq 3$.

There is a natural correspondence between graphs and partial orders. For a partial order $P = (U, R)$, the *comparability* (resp. *incomparability*) *graph* $G(P)$ of P has elements of U as vertices and an edge between every pair of comparable (resp. incomparable) elements. A graph G is a *(co)comparability graph* if G is the (in)comparability graph of a partial order P. There has been a long line of research in order to establish the complexity of recognizing partial orders of $\mathcal{P}$-dimension at most 2 (e.g. where $\mathcal{P}$ is linear orders [22] or interval orders [15]). In particular, since permutation (resp. trapezoid) graphs are the incomparability graphs of partial orders with dimension (resp. interval dimension) at most 2 [7, 22], permutation and trapezoid graphs can be recognized efficiently by the corresponding partial order algorithms [15, 22].

In contrast, not much is known so far for the recognition of partial orders P that are the intersection of orders from different classes $\mathcal{P}_1$ and $\mathcal{P}_2$. One of the longstanding open problems in this area is the recognition of *linear-interval orders* P, i.e. of partial orders $P = P_1 \cap P_2$, where P_1 is a linear order and P_2 is an interval order. In terms of graphs, this problem is equivalent to the recognition of simple-triangle (i.e. PI) graphs, since PI graphs are the incomparability graphs of linear-interval orders; this problem is well known and remains open since the introduction of PI graphs in 1987 [6] (cf. for instance the books [3, 22]).

Our contribution. In this article we establish the complexity of recognizing simple-triangle (PI) graphs, and therefore also the complexity of recognizing linear-interval orders. Given a graph G with n vertices, such that its complement $\overline{G}$ has m edges, we provide an algorithm with running time $O(n^2 m)$ that either computes a PI representation of G, or it announces that G is not a PI graph. Equivalently, given a partial order $P = (U, R)$ with $|U| = n$ and $|R| = m$, our algorithm either computes in $O(n^2 m)$ time a linear order P_1 and an interval order P_2 such that $P = P_1 \cap P_2$, or it announces that such orders P_1, P_2 do not exist. Surprisingly, it turns out that the seemingly small difference in the definition of simple-triangle (PI) graphs and triangle (PI*) graphs results in a very different behavior of their recognition problems; only recently it has been proved that the recognition of triangle graphs is NP-complete [17]. In addition, our polynomial time algorithm is in contrast to the recognition problems for the related classes of *bounded tolerance* (i.e. *parallelogram*) graphs [19] and of *max-tolerance* graphs [14], which have already been proved to be NP-complete.

As the main tool for our algorithm we introduce the notion of a *linear-interval cover* of bipartite graphs. As a second tool we identify a new tractable subclass of 3SAT, called *gradually mixed* formulas, for which we provide a linear time algorithm. The class of gradually mixed formulas is *hybrid*, i.e. it is characterized by both *relational* and *structural* restrictions on the clauses. Then, using the notion of a linear-interval cover, we are able to reduce our problem to the satisfiability problem of gradually mixed formulas.

Our algorithm proceeds as follows. First, it computes from the given graph G a bipartite graph $\widetilde{G}$, such that G is a PI graph if and only if $\widetilde{G}$ has a linear-interval cover. Second, it computes a gradually mixed Boolean formula ϕ such that ϕ is satisfiable if and only if $\widetilde{G}$ has a linear-interval cover. This formula ϕ can be written as $\phi = \phi_1 \wedge \phi_2$, where every clause of ϕ_1 has 3 literals and every clause of ϕ_2 has 2 literals. The construction of ϕ_1 and ϕ_2 is based on the fact that a necessary condition for $\widetilde{G}$ to admit a linear-interval cover is that its edges can be colored with two different colors (according to some restrictions). Then the edges of $\widetilde{G}$ correspond to literals of ϕ, while the two edge colors encode the truth value of the corresponding variables. Furthermore every clause of ϕ_1 corresponds to the edges of an *alternating cycle* in $\widetilde{G}$ (i.e. a closed walk that alternately visits edges and non-edges) of length 6, while the clauses of ϕ_2 correspond to specific pairs of edges of $\widetilde{G}$ that are not allowed to receive the same color. Finally, the equivalence between the existence of a linear-interval cover of $\widetilde{G}$ and a satisfying truth assignment for ϕ allows us to use our linear algorithm to solve satisfiability on gradually mixed formulas in order to complete our recognition algorithm.

Organization of the paper. We present in Section 2 the class of gradually mixed formulas and a linear time algorithm to solve satisfiability on this class. In Section 3 we provide the necessary notation and preliminaries on threshold graphs and alternating cycles. Then in Section 4 we introduce the notion of a linear-interval cover of bipartite graphs to characterize PI graphs, and in Section 5 we translate the linear-interval cover problem to the satisfiability problem on a gradually mixed formula. Finally, in Section 6 we present our PI graph recognition algorithm.

2. A tractable subclass of 3SAT. In this section we introduce the class of *gradually mixed* formulas and we provide a linear time algorithm for solving satisfiability on this class. Any gradually mixed formula ϕ is a mix of binary and ternary clauses. That is, there exist a 3-CNF formula ϕ_1 (i.e. a formula in conjunctive normal form with at most 3 literals per clause) and a 2-CNF formula ϕ_2 (i.e. with at most 2 literals per clause) such that $\phi = \phi_1 \wedge \phi_2$, while ϕ satisfies some constraints among its clauses. Before we define gradually mixed formulas (cf. Definition 2.2), we first define *dual* clauses.

DEFINITION 2.1. *Let ϕ_1 be a 3-CNF formula. If $\alpha = (\ell_1 \vee \ell_2 \vee \ell_3)$ is a clause of ϕ_1, then $\overline{\alpha} = (\overline{\ell_1} \vee \overline{\ell_2} \vee \overline{\ell_3})$ is the* dual *clause of α.*

Note by Definition 2.1 that, whenever α is a clause of a formula ϕ_1, the dual clause $\overline{\alpha}$ of α may belong, or may not belong, to ϕ_1.

DEFINITION 2.2. *Let ϕ_1 and ϕ_2 be CNF formulas with 3 literals and 2 literals in each clause, respectively. The mixed formula $\phi = \phi_1 \wedge \phi_2$ is* gradually mixed *if the next two conditions are satisfied:*

1. *Let α and β be two clauses of ϕ_1. Then α does not share exactly one literal with either the clause β or the clause $\overline{\beta}$.*
2. *If $\alpha = (\ell_1 \vee \ell_2 \vee \ell_3)$ is a clause of ϕ_1 and $(\ell_0 \vee \overline{\ell_1})$ is a clause of ϕ_2, then ϕ_2 contains also (at least) one of the clauses $\{(\ell_0 \vee \ell_2), (\ell_0 \vee \ell_3)\}$.*

As an example of a gradually mixed formula, consider the formula $\phi = \phi_1 \wedge \phi_2$, where $\phi_1 = (x_1 \vee \overline{x_2} \vee x_3) \wedge (\overline{x_1} \vee x_2 \vee x_4) \wedge (x_5 \vee x_6 \vee \overline{x_7})$ and $\phi_2 = (x_8 \vee \overline{x_3}) \wedge (x_8 \vee x_1) \wedge (x_8 \vee x_4) \wedge (\overline{x_8} \vee x_9) \wedge (x_5 \vee x_{10}) \wedge (\overline{x_6} \vee x_{10})$.

Note by Definition 2.2 that the class of gradually mixed formulas contains 2SAT as a proper subclass, since every 2-CNF formula ϕ_2 can be written as a gradually mixed formula $\phi = \phi_1 \wedge \phi_2$ where $\phi_1 = \emptyset$. Furthermore the class of gradually mixed formulas ϕ is a *hybrid* class, since the conditions of Definition 2.2 concern simultaneously *relational* restrictions (i.e. where the clauses are restricted to be of certain types) and *structural* restrictions (i.e. where there are restrictions on how different clauses interact with each other). The intuition for the term *gradually mixed* in Definition 2.2 is that, whenever the sub-formulas ϕ_1 and ϕ_2 share more variables, the number of clauses of ϕ_2 that are imposed by condition 2 of Definition 2.2 increases. In the next theorem we use resolution to prove that satisfiability can be solved in linear time on gradually mixed formulas.

THEOREM 2.3. *There exists a linear time algorithm which decides whether a given gradually mixed formula ϕ is satisfiable and computes a satisfying truth assignment of ϕ, if one exists.*

Proof. Let $\phi = \phi_1 \wedge \phi_2$, where ϕ_1 is a 3-CNF formula and ϕ_2 is a 2-CNF formula. We first scan through all clauses of ϕ to remove all tautologies, i.e. all clauses which contain both a literal and its negation, since such clauses are always satisfiable. Furthermore we eliminate all double literal occurrences in every clause. In the remainder of the proof we denote by ϕ the resulting formula after the removal of tautologies and the elimination of double literal occurrences in the clauses. Note that, during this elimination procedure, some clauses of ϕ_1 may become 2-CNF clauses. In the resulting formula we denote by ϕ_1' the conjunction of the clauses that have 3 literals each, and by ϕ_1'' the conjunction of the clauses of ϕ_1 that remain with 1 or 2 literals each. In particular, since also in every clause of ϕ_1 no literal is the negation of another one (as we removed from ϕ all tautologies), the literals of every clause in ϕ_1' correspond to three distinct variables.

Then we compute a 2-CNF formula ϕ_0 (in time linear to the size of ϕ) as follows. Initially ϕ_0 is empty. First we mark all literals ℓ for which the 2-CNF formula $\phi_1'' \wedge \phi_2$ includes the clause (ℓ). Then we scan through all clauses of the 3-CNF formula ϕ_1'. For every clause $(\ell_1 \vee \ell_2 \vee \ell_3)$ of ϕ_1', such that the literal $\overline{\ell_1}$ (resp. $\overline{\ell_2}$ or $\overline{\ell_3}$) has been marked, we add to ϕ_0 the clause $(\ell_2 \vee \ell_3)$ (resp. the clause $(\ell_1 \vee \ell_3)$ or $(\ell_1 \vee \ell_2)$).

If $\phi \wedge \phi_0$ is satisfiable then clearly ϕ is also satisfiable as a sub-formula of $\phi \wedge \phi_0$. Conversely, suppose that ϕ is satisfied by the truth assignment τ. Let $\gamma = (\ell_1 \vee \ell_2)$ be an arbitrary clause of ϕ_0. The existence of γ in ϕ_0 implies the existence of some clauses $\alpha = (\overline{\ell_3})$ and $\beta = (\ell_1 \vee \ell_2 \vee \ell_3)$ in ϕ. Therefore, since $\alpha = \beta = 1$ in τ by assumption, it follows that $\ell_3 = 0$ in τ. Thus the clause β equals $(\ell_1 \vee \ell_2)$ in τ, and therefore $\gamma = 1$ in τ. That is, τ satisfies also ϕ_0. Therefore ϕ is satisfiable if and only if $\phi \wedge \phi_0$ is satisfiable.

In the remainder of the proof, we prove that $\phi \wedge \phi_0$ is satisfiable if and only if the 2-CNF formula $\phi_1'' \wedge \phi_2 \wedge \phi_0$ is satisfiable. The one direction is immediate, i.e. if $\phi \wedge \phi_0$ is satisfiable then $\phi_1'' \wedge \phi_2 \wedge \phi_0$ is also satisfiable as a sub-formula of $\phi \wedge \phi_0$. Conversely, suppose that $\phi_1'' \wedge \phi_2 \wedge \phi_0$ is satisfiable and let τ be a satisfying truth assignment of this formula. If τ satisfies all clauses of ϕ'_1, then clearly τ is also a satisfying truth assignment of $\phi \wedge \phi_0$. Otherwise let $\alpha = (\ell_1 \vee \ell_2 \vee \ell_3)$ be a clause of ϕ'_1 that is not satisfied by τ. Then $\ell_1 = \ell_2 = \ell_3 = 0$ in τ. In this case, we construct the truth assignment τ' from τ by flipping the value of one (arbitrary) literal of $\{\ell_1, \ell_2, \ell_3\}$ in τ. Assume without loss of generality that the value of ℓ_1 flips from τ to τ', while the values of all other variables remain the same in both τ and τ'. Recall that the literals $\{\ell_1, \ell_2, \ell_3\}$ correspond to three distinct variables, since we eliminated all double occurrences of literals in all clauses in ϕ_1. Therefore $\ell_1 = \overline{\ell_2} = \overline{\ell_3} = 1$ in τ', and thus $\alpha = 1$ in τ'.

Suppose that there exists a clause $\beta = (\ell_4 \vee \ell_5 \vee \ell_6)$ of ϕ'_1 where $\beta = 1$ in τ and $\beta = 0$ in τ'. Then clearly one of the literals of β equals $\overline{\ell_1}$, since $\overline{\ell_1}$ is the only literal whose value changes in τ' from 1 to 0. Assume without loss of generality that $\ell_4 = \overline{\ell_1}$, i.e. α shares at least one literal with $\overline{\beta} = (\overline{\ell_4} \vee \overline{\ell_5} \vee \overline{\ell_6})$. Therefore, since ϕ is a gradually mixed formula by assumption, it follows by Definition 2.2 that α shares at least one more literal with $\overline{\beta}$. Assume without loss of generality that $\ell_5 = \overline{\ell_2}$. Then, since by assumption $\ell_2 = 0$ in both τ and τ', it follows that the clause $\beta = (\ell_4 \vee \ell_5 \vee \ell_6) = (\overline{\ell_1} \vee \overline{\ell_2} \vee \ell_6)$ is satisfied in τ', which is a contradiction to our assumption. Therefore for every clause β of ϕ'_1, if $\beta = 1$ in τ then also $\beta = 1$ in τ'.

We now prove that all clauses of the 2-CNF formula $\phi''_1 \wedge \phi_2 \wedge \phi_0$ remain satisfied in τ'. First consider an arbitrary clause γ of ϕ_0 that contains one of the literals $\{\ell_1, \overline{\ell_1}\}$. If γ contains the literal ℓ_1 then $\gamma = 1$ in τ', since $\ell_1 = 1$ in τ'. Let γ contain the literal $\overline{\ell_1}$, and let $\gamma = (\overline{\ell_1} \vee \ell_4)$. Then it follows by the construction of the formula ϕ_0 that there exists a literal ℓ_5, such that $(\overline{\ell_1} \vee \ell_4 \vee \ell_5)$ is a clause of ϕ'_1 and $(\overline{\ell_5})$ is a clause of $\phi''_1 \wedge \phi_2$. Note that $(\overline{\ell_1} \vee \ell_4 \vee \ell_5) = 1$ in τ, since $\ell_1 = 0$ in τ by assumption. Therefore also $(\overline{\ell_1} \vee \ell_4 \vee \ell_5) = 1$ in τ' by the previous paragraph. Thus, since $\overline{\ell_1} = 0$ in τ', it follows that $(\ell_4 \vee \ell_5) = 1$ in τ'. Furthermore, since τ satisfies $\phi''_1 \wedge \phi_2$ by assumption, it follows that $(\overline{\ell_5}) = 1$ in τ, and thus $\ell_5 = 0$ in both τ and τ'. Therefore $\ell_4 = 1$ in τ', since $(\ell_4 \vee \ell_5) = 1$ in τ', and thus $\gamma = (\overline{\ell_1} \vee \ell_4) = 1$ in τ'. That is, all clauses γ of ϕ_0 remain satisfied in the assignment τ'.

Now consider a clause γ of ϕ_2 that contains one of the literals $\{\ell_1, \overline{\ell_1}\}$. If γ contains ℓ_1 then $\gamma = 1$ in τ', since $\ell_1 = 1$ in τ'. Let γ contain the literal $\overline{\ell_1}$, and let $\gamma = (\overline{\ell_1} \vee \ell_4)$. Note that $\ell_4 \neq \ell_1$, since we removed all tautologies from ϕ. Suppose that $\ell_4 = \overline{\ell_1}$, i.e. $\gamma = (\overline{\ell_1})$. Then, since $\alpha = (\ell_1 \vee \ell_2 \vee \ell_3)$ is a clause of ϕ_1 by assumption, the formula ϕ_0 contains (by construction) the clause $(\ell_2 \vee \ell_3)$. Thus, since τ satisfies ϕ_0 by assumption, it follows that $\ell_2 = 1$ or $\ell_3 = 1$ in τ. This is a contradiction, since $\ell_1 = \ell_2 = \ell_3 = 0$ in τ. Therefore $\ell_4 \notin \{\ell_1, \overline{\ell_1}\}$. Thus, since ϕ is a gradually mixed formula by assumption, it follows by Definition 2.2 that ϕ_2 has also one of the clauses $\{(\ell_4 \vee \ell_2), (\ell_4 \vee \ell_3)\}$. Assume without loss of generality that ϕ_2 has the clause $(\ell_4 \vee \ell_2)$. Then, since τ satisfies ϕ_2 by assumption and $\ell_2 = 0$ in τ, it follows that $\ell_4 = 1$ in τ. Furthermore, since $\ell_4 \notin \{\ell_1, \overline{\ell_1}\}$, it remains $\ell_4 = 1$ in τ', and thus $\gamma = (\overline{\ell_1} \vee \ell_4) = 1$ in τ'. That is, all clauses γ of ϕ_2 remain satisfied in the assignment τ'.

Finally consider a clause γ of ϕ''_1 that contains one of the literals $\{\ell_1, \overline{\ell_1}\}$. If γ contains ℓ_1 then $\gamma = 1$ in τ', since $\ell_1 = 1$ in τ'. Let γ contain the literal $\overline{\ell_1}$, and let $\gamma = (\overline{\ell_1} \vee \ell_4)$. Note that $\ell_4 \neq \ell_1$, since we removed all tautologies from ϕ. Suppose that $\ell_4 = \overline{\ell_1}$, i.e. $\gamma = (\overline{\ell_1})$. Then, since $\alpha = (\ell_1 \vee \ell_2 \vee \ell_3)$ is a clause of ϕ_1 by assumption, the formula ϕ_0 contains by construction the clause $(\ell_2 \vee \ell_3)$. Thus $\ell_2 = 1$ or $\ell_3 = 1$ in τ, since τ satisfies ϕ_0 by assumption. This is a contradiction, since $\ell_1 = \ell_2 = \ell_3 = 0$ in τ. Therefore $\ell_4 \notin \{\ell_1, \overline{\ell_1}\}$. Recall that ϕ''_1 contains exactly those clauses of ϕ_1 which remain with 1 or 2 literals each, after eliminating all double literal occurrences in every clause of ϕ. That is, the clause γ was before the double literal elimination one of the clauses $(\overline{\ell_1} \vee \ell_4 \vee \ell_4)$ and $(\overline{\ell_1} \vee \overline{\ell_1} \vee \ell_4)$. Furthermore $\alpha = (\ell_1 \vee \ell_2 \vee \ell_3)$ and γ are two different clauses of ϕ_1, since α belongs to ϕ_1' and γ belongs to ϕ_1''. Moreover α shares the literal ℓ_1 with the dual clause $\overline{\gamma}$ of γ. If γ was the clause $(\overline{\ell_1} \vee \ell_4 \vee \ell_4)$ before the double literal elimination, then Definition 2.2 implies that $\ell_4 = \overline{\ell_2}$ or $\ell_4 = \overline{\ell_3}$. Therefore $\ell_4 = 1$ in τ', since $\ell_2 = \ell_3 = 0$ in both τ and τ', and thus $\gamma = (\overline{\ell_1} \vee \ell_4) = 1$ in τ'. Otherwise, if γ was the clause $(\overline{\ell_1} \vee \overline{\ell_1} \vee \ell_4)$ before the double literal elimination, then Definition 2.2 implies that $\ell_1 = \ell_2$, or $\ell_1 = \ell_3$, or $\ell_4 = \overline{\ell_2}$, or $\ell_4 = \overline{\ell_3}$. Recall that α is a clause of ϕ_1' by assumption, and thus $\ell_1 \neq \ell_2$ and $\ell_1 \neq \ell_3$. Therefore $\ell_4 = \overline{\ell_2}$ or $\ell_4 = \overline{\ell_3}$, and thus $\ell_4 = 1$ in τ', since $\ell_2 = \ell_3 = 0$ in both τ and τ'. Therefore $\gamma = (\overline{\ell_1} \vee \ell_4) = 1$ in τ'. That is, all clauses γ of ϕ_1'' remain satisfied in the assignment τ'.

Summarizing, all clauses of the 2-CNF formula $\phi_1'' \wedge \phi_2 \wedge \phi_0$ remain satisfied in τ'. Furthermore, $\alpha = 1$ in τ', while for every clause β of ϕ_1', if $\beta = 1$ in τ then also $\beta = 1$ in τ'. Thus, according to the above transition from τ to τ', we can modify iteratively the truth assignment τ to a truth assignment τ'' that satisfies all clauses of $\phi \wedge \phi_0$. Therefore $\phi \wedge \phi_0$ is satisfiable if and only if the 2-CNF formula $\phi_1'' \wedge \phi_2 \wedge \phi_0$ is satisfiable.

Since the transition from the assignment τ to the assignment τ' can be done in constant time (we only need to flip locally the value of one literal ℓ_1 in the clause $\alpha = (\ell_1 \vee \ell_2 \vee \ell_3)$ of ϕ_1'), the computation of τ'' from τ can be done in time linear to the size of $\phi \wedge \phi_0$. Therefore, since a satisfying truth assignment τ of the 2-CNF formula $\phi_1'' \wedge \phi_2 \wedge \phi_0$ (if one exists) can be computed in linear time using any standard linear time algorithm for the 2-SAT problem (e.g. [8]), a satisfying truth assignment $\tau''$of $\phi \wedge \phi_0$ (if one exists) can be also computed in time linear to the size of $\phi \wedge \phi_0$ (and thus also in time linear to the size of ϕ). This completes the proof of the theorem. $\square$

The conditions of Definition 2.2 which guarantee the tractability of gradually mixed formulas are *minimal*, in the sense that, if we remove any of these two conditions, the resulting subclass of 3SAT is NP-complete.

Indeed, assume that we impose only the *first* condition of Definition 2.2 to the mixed formula $\phi = \phi_1 \wedge \phi_2$. Then we can reduce 3SAT to this subclass as follows. Let ϕ_0 be an instance of 3SAT. We define ϕ_1 to be the formula obtained by ϕ_0 if we replace every literal ℓ of ϕ_0 by a new variable x_ℓ. For every two of these new variables x_ℓ and $x_{\ell'}$ in ϕ_1, we add to ϕ_2 the clauses $(x_\ell \vee \overline{x_{\ell'}}) \wedge (\overline{x_\ell} \vee x_{\ell'})$ if $\ell = \ell'$ in ϕ_0, and we add to ϕ_2 the clauses $(x_\ell \vee x_{\ell'}) \wedge (\overline{x_\ell} \vee \overline{x_{\ell'}})$ if $\ell = \overline{\ell'}$ in ϕ_0. Then $\phi = \phi_1 \wedge \phi_2$ satisfies the first condition of Definition 2.2 (since no two clauses of ϕ_1 share any variable), while ϕ_0 is satisfiable if and only if ϕ is satisfiable.

On the other hand, assume that we impose only the *second* condition of Definition 2.2 to the mixed formula $\phi = \phi_1 \wedge \phi_2$. Then, by setting $\phi_2 = \emptyset$, we can include in the resulting class *every* 3-CNF formula, and thus this class is NP-complete.

3. Preliminaries.

3.1. Notation. In the remainder of this article we consider finite, simple, and undirected graphs. Given a graph G, we denote by $V(G)$ and $E(G)$ the sets of its vertices and edges, respectively. An edge between two vertices u and v of a graph $G = (V, E)$ is denoted by uv, and in this case u and v are said to be *adjacent*. The *neighborhood* of a vertex $u \in V$ is the set $N(u) = \{v \in V \mid uv \in E\}$ of its adjacent vertices. The complement of G is denoted by $\overline{G}$, i.e. $\overline{G} = (V, \overline{E})$, where $uv \in \overline{E}$ if and only if $uv \notin E$. For any subset $E_0 \subseteq E$ of the edges of G, we denote for simplicity

$G - E_0 = (V, E \setminus E_0)$. A subset $S \subseteq V$ of its vertices induces an *independent set* in G if $uv \notin E$ for every pair of vertices $u, v \in S$. Furthermore, S induces a *clique* in G if $uv \in E$ for every pair $u, v \in S$. For two graphs $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$, we denote $G_1 \subseteq G_2$ whenever $E_1 \subseteq E_2$. Moreover, we denote for simplicity by $G_1 \cup G_2$ and $G_1 \cap G_2$ the graphs $(V, E_1 \cup E_2)$ and $(V, E_1 \cap E_2)$, respectively. A graph G is a *split graph* if its vertices can be partitioned into a clique K and an independent set I. Furthermore, $G = (V, E)$ is a *threshold graph* if we can assign to each vertex $v \in V$ a real weight a_v, such that $uv \in E$ if and only if $a_u + a_v \geq 1$.

A *proper k-coloring* of a graph G is an assignment of k colors to the vertices of G, such that adjacent vertices are assigned different colors. The smallest k for which there exists a proper k-coloring of G is the *chromatic number* of G, denoted by $\chi(G)$. If $\chi(G) = 2$ then G is a *bipartite* graph; in this case the vertices of G are partitioned into two independent sets, the *color classes*. A bipartite graph G is denoted by $G = (U, V, E)$, where U and V are its color classes and E is the set of edges between them. For a bipartite graph $G = (U, V, E)$, its *bipartite complement* is the graph $\widehat{G} = (U, V, \widehat{E})$, where for two vertices $u \in U$ and $v \in V$, $uv \in \widehat{E}$ if and only if $uv \notin E$. A bipartite graph $G = (U, V, E)$ is a *chain graph* if the vertices of each color class can be ordered by inclusion of their neighborhoods, i.e. $N(u) \subseteq N(v)$ or $N(v) \subseteq N(u)$ for any two vertices u, v in the same color class. Note that chain graphs are closed under bipartite complementation, i.e. G is a chain graph if and only if $\widehat{G}$ is a chain graph.

For any graph $G = (V, E)$ and any graph class $\mathcal{G}$, the *$\mathcal{G}$-cover number* of G is the smallest k such that $E = \bigcup_{i=1}^{k} E_i$, where $G_i = (V, E_i) \in \mathcal{G}$, $1 \leq i \leq k$; in this case the graphs $\{G_i\}_{i=1}^{k}$ are a *$\mathcal{G}$-cover* of G. For several graph classes $\mathcal{G}$ it is NP-complete to decide whether the $\mathcal{G}$-cover number of a graph is at most k, where $k \geq 3$, see e.g. [23]. Throughout the paper, whenever a set of the chain graphs $\{G_i\}_{i=1}^{k}$ form a chain-cover of a bipartite graph G, then all these graphs are assumed to have the same color classes as G.

For any partial order $P = (U, R)$, we denote by $\overline{P} = (U, \overline{R})$ the *inverse* partial order of P, i.e. for any two elements $u, v \in U$, $u <_{\overline{P}} v$ if and only if $v <_P u$. For any two partial orders $P_1 = (U, R_1)$ and $P_2 = (U, R_2)$, we denote $P_1 \subseteq P_2$ whenever $R_1 \subseteq R_2$. Moreover, we denote for simplicity $P_1 \cup P_2$ and $P_1 \cap P_2$ for the partial orders $(U, R_1 \cup R_2)$ and $(U, R_1 \cap R_2)$, respectively. If P_2 is a linear order and $P_1 \subseteq P_2$, then P_2 is a *linear extension* of P_1. The orders P_1 and P_2 *contradict each other* if there exist two elements $u, v \in U$ such that $u <_{P_1} v$ and $v <_{P_2} u$. The *linear-interval dimension* of a partial order P (denoted $lidim(P)$) is the lexicographically smallest pair (k, ℓ) such that $P = \bigcap_{i=1}^{k} P_i$, where $\{P_i\}_{i=1}^{k}$ are interval orders and exactly ℓ among them are not linear orders. In particular, P is a *linear-interval order* if its linear-interval dimension is at most $(2, 1)$, i.e. $P = P_1 \cap P_2$, where P_1 is a linear order and P_2 is an interval order.

3.2. Threshold graphs and alternating cycles. In this section we provide preliminary definitions and known results on alternating cycles and on threshold graphs, which will be useful for the remainder of the paper.

Definition 3.1. *Let $G = (V, E)$ be a graph, $\widetilde{E} \subseteq E$ be an edge subset, and $k \geq 2$. A set of $2k$ (not necessarily distinct) vertices $v_1, v_2, \ldots, v_{2k} \in V$ builds an* alternating cycle AC_{2k} in $\widetilde{E}$*, if $v_i v_{i+1} \in \widetilde{E}$ whenever i is even and $v_i v_{i+1} \notin E$ whenever i is odd (where indices are* mod $2k$*). Furthermore, we say that G has an alternating cycle AC_{2k}, whenever G has an AC_{2k} in the edge set $\widetilde{E} = E$.*

For instance, for $k = 3$, there exist two different possibilities for an AC_6, which are illustrated in Figures 1(a) and 1(b). These two types of an AC_6 are called an *alternating path of length* 5 or *of length* 6, respectively (AP_5 and AP_6 for short, respectively). In an AP_6 on vertices $v_1, v_2, v_3, v_4, v_5, v_6$, if there exist the edges v_1v_3 and v_2v_6 (or, symmetrically, the edges v_3v_5 and v_4v_2, or the edges v_5v_1 and v_6v_4), then this AP_6 is called a *double* AP_6, cf. Figure 1(c).

DEFINITION 3.2. *Let $G = (V, E)$ be a graph and $v_1, \ldots, v_6$ be the vertices of an AP_6. Then the non-edge v_1v_2 (resp. the non-edge v_3v_4, v_5v_6) is a* base *of the AP_6 and the edge v_4v_5 (resp. the edge v_6v_1, v_2v_3) is the corresponding* ceiling *of this AP_6.*

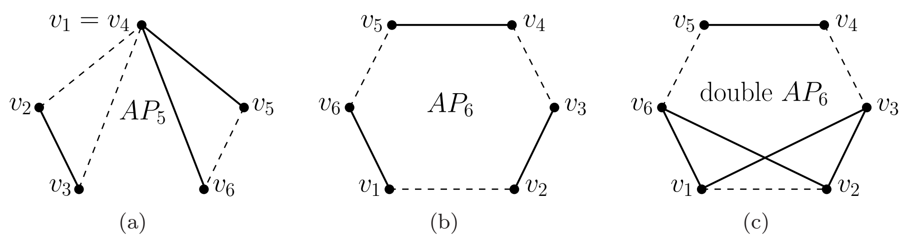

FIGURE 1. *All possibilities for an AC_6: (a) an alternating path AP_5 of length 5, (b) an alternating path AP_6 of length 6, and (c) a double AP_6. The solid lines denote edges of the graph and the dashed lines denote non-edges of the graph.*

Furthermore, note that for $k = 2$, a set of four vertices $v_1, v_2, v_3, v_4 \in V$ builds an alternating cycle AC_4 if $v_1v_2, v_3v_4 \in E$ and $v_1v_4, v_2v_3 \notin E$. There are three possible graphs on four vertices that build an alternating cycle AC_4, namely $2K_2$, P_4, and C_4, which are illustrated in Figure 2.

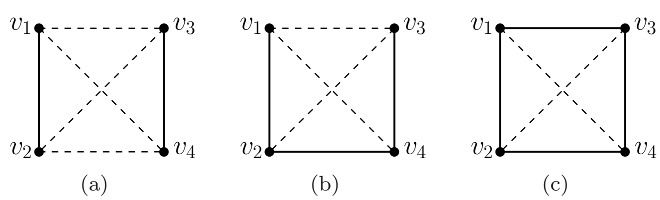

FIGURE 2. *The three possible AC_4's: (a) a $2K_2$, (b) a P_4, and (c) a C_4.*

Alternating cycles can be used to characterize threshold and chain graphs. In particular, threshold graphs are the graphs with no induced AC_4, and chain graphs are the bipartite graphs with no induced $2K_2$ [16]. We define now for any bipartite graph G the associated split graph of G, which we will use extensively in the remainder of the paper.

DEFINITION 3.3. *Let $G = (U, V, E)$ be a bipartite graph. The* associated split graph *of G is the split graph $H_G = (U \cup V, E')$, where $E' = E \cup (V \times V)$, i.e. H_G is the split graph made by G by replacing the independent set V of G by a clique.*

OBSERVATION 1. *Let G be a bipartite graph and H_G be the associated split graph of G. Then, G has an induced $2K_2$ if and only if H_G has an induced AC_4, and in this case this AC_4 is a P_4.*

The next lemma connects the chain cover number $ch(G)$ of a bipartite graph G with the threshold cover number $t(H_G)$ of the associated split graph H_G of G. Recall that the problem of deciding whether a graph G has threshold cover number at most a given number k is NP-complete for $k \geq 3$ [23], while it is polynomial for $k = 2$ [21].

Lemma 3.4 ([16]). *Let $G = (U, V, E)$ be a bipartite graph. Then $ch(G) = t(H_G)$.*

The next two definitions of a *conflict* between two edges and the *conflict graph* are essential for our results.

Definition 3.5. *Let $G = (V, E)$ be a graph and $e_1, e_2 \in E$. If the vertices of e_1 and e_2 build an AC_4 in G, then e_1 and e_2 are in* conflict*, and in this case we denote $e_1 || e_2$ in G. Furthermore, an edge $e \in E$ is* committed *if there exists an edge $e' \in E$ such that $e || e'$; otherwise e is* uncommitted.

Definition 3.6 ([21]). *Let $G = (V, E)$ be a graph. The* conflict graph *$G^* = (V^*, E^*)$ of G is defined by*
- *$V^* = E$ and*
- *for every $e_1, e_2 \in E$, $e_1 e_2 \in E^*$ if and only if $e_1 || e_2$ in G.*

Observation 2. *Let $G = (V, E)$ be a graph and let $e \in E$. If e is uncommitted, then e is an isolated vertex in the conflict graph G^* of G.*

Observation 3. *Let $G = (V, E)$ be a split graph. Let K and I be a partition of V, such that K is a clique and I is an independent set (such a partition always exists for split graphs). Then, every edge of K is uncommitted.*

Lemma 3.7. *Let G be a graph and let the vertices $v_1, \ldots, v_6$ of G build an AP_6 (an alternating path of length 6). Assume that among the three edges $\{v_2v_3, v_4v_5, v_6v_1\}$ of this AP_6, no pair of edges is in conflict. Then the edges v_3v_6, v_4v_1, v_5v_2 exist in G and $v_4v_5 || v_3v_6$, $v_2v_3 || v_4v_1$, and $v_6v_1 || v_5v_2$.*

Proof. Suppose that v_3v_6 is not an edge of G. Then the edges v_2v_3 and v_6v_1 are in conflict, since v_1v_2 is not an edge of G (cf. Figure 1(b)), which is a contradiction to the assumption of the lemma. Therefore v_3v_6 is an edge of G. By symmetry, it follows that also v_4v_1 and v_5v_2 are edges in G. Note now that the edges $v_4v_5 || v_3v_6$ are in conflict, since v_3v_4 and v_5v_6 are not edges of G. By symmetry, it follows that also $v_2v_3 || v_4v_1$, and $v_6v_1 || v_5v_2$. □

Note that the threshold cover number $t(G)$ of a graph $G = (V, E)$ equals the smallest k, such that the edge set E of G can be partitioned into k sets $E_1, E_2, \ldots, E_k$, each having a *threshold completion in* G (that is, there exists for every $i = 1, 2, \ldots, k$ an edge set E'_i, such that $E_i \subseteq E'_i \subseteq E$ and (V, E'_i) is a threshold graph). The following characterization of subgraphs that admit a threshold completion in a given graph G has been proved in [12].

Lemma 3.8 ([12]). *Let H be a subgraph of a graph $G = (V, E)$. Then H has a threshold completion in G if and only if G has no AC_{2k}, $k \geq 2$, on the edges of H.*

If the conditions of Lemma 3.8 are satisfied, then such a threshold completion of H in G can be computed in linear time, as the next lemma states.

Lemma 3.9 ([21]). *If a subgraph H of $G = (V, E)$ has a threshold completion in G, then it can be computed in $O(|V| + |E|)$ time.*

Corollary 3.10. *Let $G = (V, E)$ be a graph. Then, $t(G) = 1$ if and only if G has no AC_{2k}, $k \geq 2$. Furthermore, $t(G) \leq 2$ if and only if the set E of edges can be partitioned into two sets E_1 and E_2, such that G has no AC_{2k}, $k \geq 2$, in each E_i, $i = 1, 2$.*

Proof. First note that $t(G) = 1$ if and only if G is a threshold graph. Therefore, Lemma 3.8 implies that $t(G) = 1$ if and only if G has no AC_{2k}, $k \geq 2$.

Recall that the threshold cover number $t(G)$ of a graph $G = (V, E)$ equals the smallest k, such that the edge set E of G can be partitioned into k sets $E_1, E_2, \ldots, E_k$, each having a *threshold completion in* G. Therefore, if $t(G) \leq 2$, Lemma 3.8 implies that E can be partitioned into two sets E_1 and E_2, such that G has no AC_{2k}, $k \geq 2$, in each E_i, $i = 1, 2$. Note that, in the case where $t(G) = 1$ (i.e. G is a threshold graph), we can set $E_1 = E$ and $E_2 = \emptyset$. Conversely, suppose that E can be partitioned into two such sets E_1 and E_2. Then Lemma 3.8 implies that both graphs $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ have a threshold completion in G, where $G_1 \cup G_2 = G$. Therefore $t(G) \leq 2$. $\square$

It can be easily proved that, for every graph G, the chromatic number $\chi(G^*)$ of its conflict graph G^* provides a lower bound for the threshold cover number $t(G)$ of G, as the next lemma states.

LEMMA 3.11 ([16]). *Let G be a graph. Then $\chi(G^*) \leq t(G)$.*

Lemma 3.11 immediately implies that a necessary condition for a graph G to have threshold cover number $t(G) \leq 2$ is that $\chi(G^*) \leq 2$, i.e. that G^* is a bipartite graph. The main result of [21] is the next theorem, which proves that this is also a sufficient condition for graphs G with $\chi(G^*) \leq 2$.

THEOREM 3.12 ([21]). *If the conflict graph G^* of a graph $G = (V, E)$ is bipartite (i.e. $\chi(G^*) \leq 2$), then $t(G) \leq 2$. Moreover, E can be partitioned in $O(|E|(|V| + |E|))$ time into two sets E_1 and E_2, such that G has no AC_{2k}, $k \geq 2$, in each E_i, $i = 1, 2$.*

Due to the next theorem, it suffices for bipartite conflict graphs G^* to consider only small alternating cycles AC_{2k} with $k \leq 3$.

THEOREM 3.13 ([12]). *Suppose that the conflict graph G^* of a graph $G = (V, E)$ is bipartite (i.e. $\chi(G^*) \leq 2$), with (vertex) color classes E_1 and E_2. If G has an AC_{2k} on the edges of E_1 (resp. of E_2), where $k \geq 3$, then G has also an AC_6 in E_1 (resp. of E_2).*

LEMMA 3.14 ([13]). *Let $G = (V, E)$ be a split graph. Let K and I be a partition of V such that K induces a clique and I induces an independent set in G. Assume that the vertices $v_1, \ldots, v_6$ build an AP_6 in G. Then either $v_1, v_3, v_5 \in K$ and $v_2, v_4, v_6 \in I$, or $v_1, v_3, v_5 \in I$ and $v_2, v_4, v_6 \in K$.*

LEMMA 3.15. *Any split graph G does not contain any AP_5 or any double AP_6.*

Proof. Let $G = (V, E)$ be a split graph and let K and I be a partition of V, such that K induces a clique and I induces an independent set in G; such a partition exists by the definition of split graphs. The fact that a split graph does not contain any AP_5 has been proved in [13]. However, for the sake of completeness we present here a simple proof of this fact. Assume that G contains an AP_5 on the vertices $v_1, v_2, v_3, v_4, v_5, v_6$ where $v_1 = v_4$, cf. Figure 1(a). Suppose first that $v_1 = v_4 \in K$. Then, since $v_2, v_3 \notin N(v_1)$, it follows that $v_2, v_3 \in I$. This is a contradiction, since $v_2v_3 \in E$. Suppose now that $v_1 = v_4 \in I$. Then, since $v_5, v_6 \in N(v_1)$, it follows that $v_5, v_6 \in K$. This is a contradiction, since $v_5v_6 \notin E$. Therefore G does not contain any AP_5.

Now assume that G contains an AP_6 on the vertices $v_1, v_2, v_3, v_4, v_5, v_6$, cf. Figure 1(b). We will prove that this is not a double AP_6 (cf. Figure 1(c)). Indeed, Lemma 3.14 implies that either $v_1, v_3, v_5 \in K$ and $v_2, v_4, v_6 \in I$, or $v_1, v_3, v_5 \in I$ and $v_2, v_4, v_6 \in K$. In both cases, none of the pairs of edges $\{v_1v_3, v_2v_6\}$, $\{v_3v_5, v_4v_2\}$, and $\{v_5v_1, v_6v_4\}$ can exist simultaneously in G. Therefore, G has no double AP_6. This completes the proof of the lemma. $\square$

4. Linear-Interval covers of bipartite graphs. In this section we introduce the crucial notion of a *linear-interval cover* of bipartite graphs (cf. Definition 4.6). Then we use linear-interval covers to provide a new characterization of PI graphs (cf. Theorem 4.8), which is one of the main tools for our PI graph recognition algorithm. First we provide in the next theorem the characterization of PI graphs using linear orders and interval orders.

THEOREM 4.1. *Let $G = (V, E)$ be a cocomparability graph and P be a partial order of $\overline{G}$. Then G is a PI graph if and only if $P = P_1 \cap P_2$, where P_1 is a linear order and P_2 is an interval order.*

Proof. For the purposes of the proof, a partial order $P = (U, R)$ is called a PI order [5], if there exists a PI representation (i.e. a simple-triangle representation) R, such that for any two $u, v \in U$, $u <_P v$ if and only if the triangle associated to u lies in R entirely to the left of the triangle associated to v.

Suppose that $P = P_1 \cap P_2$ for two partial orders P_1 and P_2, where P_1 is a linear order and P_2 is an interval order. Then P is a PI order [5], and thus G is a PI graph. Conversely, suppose that G is a PI graph. Equivalently, P is a PI order, and thus the linear-interval dimension of P is $lidim(P) \leq (2, 1)$ [5]. That is, $P = P_1 \cap P_2$ for two partial orders P_1 and P_2, where P_1 is a linear order and P_2 is an interval order. Moreover, whenever we are given a partial order P such that $P = P_1 \cap P_2$, where P_1 is a linear order and P_2 is an interval order, it is straightforward to compute a PI model for P (cf. [5]). Equivalently, we can easily construct in this case a PI representation of the incomparability graph G of P (cf. lines 13-15 of Algorithm 1 below). ◻

For every partial order P we define now the domination bipartite graph $C(P)$, which has been used to characterize interval orders [15]. Here "C" stands for "Comparable", since the definition of $C(P)$ uses the comparable elements of P.

DEFINITION 4.2 ([15]). *Let $P = (U, R)$ be a partial order, where $U = \{u_1, u_2, \ldots, u_n\}$. Furthermore let $V = \{v_1, v_2, \ldots, v_n\}$. The* domination bipartite graph *$C(P) = (U, V, E)$ is defined such that $u_i v_j \in E$ if and only if $u_i <_P u_j$.*

LEMMA 4.3 ([15]). *Let $P = (U, R)$ be a partial order. Then, P is an interval order if and only if $C(P)$ is a chain graph.*

Extending the notion of $C(P)$, we now introduce the bipartite graph $NC(P)$ to characterize linear orders (cf. Lemma 4.5). Here "NC" stands for "Non-strictly Comparable". Namely, this graph can be obtained by adding to the graph $C(P)$ the perfect matching $\{u_i v_i \mid i = 1, 2, \ldots, n\}$ on the vertices of U and V.

DEFINITION 4.4. *Let $P = (U, R)$ be a partial order, where $U = \{u_1, u_2, \ldots, u_n\}$. Furthermore let $V = \{v_1, v_2, \ldots, v_n\}$. Then, $NC(P) = (U, V, E)$ is the bipartite graph, such that $u_i v_j \in E$ if and only if $u_i \leq_P u_j$.*

LEMMA 4.5. *Let $P = (U, R)$ be a partial order. Then, P is a linear order if and only if $NC(P)$ is a chain graph.*

Proof. Let $U = \{u_1, u_2, \ldots, u_n\}$. Suppose that P is a linear order, i.e. $u_1 <_P u_2 <_P \ldots <_P u_n$. Then, by Definition 4.4, the set of neighbors of a vertex $u_i \in U$ in the graph $NC(P)$ is $N(u_i) = \{v_i, v_{i+1}, \ldots, v_n\}$. Therefore, $N(u_n) \subset N(u_{n-1}) \subset \ldots \subset N(u_1)$, and thus $NC(P)$ is a chain graph.

Suppose now that $NC(P)$ is a chain graph. Then the sets of neighbors of the vertices of U in the graph $NC(P)$ can be linearly ordered by inclusion. Let without loss of generality $N(u_1) \subseteq N(u_2) \subseteq \ldots \subseteq N(u_n)$. Therefore, since $v_i \in N(u_i)$ in $NC(P)$ for every $i = 1, 2, \ldots, n$, it follows that $v_i \in N(u_j)$ in $NC(P)$ whenever $i < j$. Therefore, by Definition 4.4, $u_j <_P u_i$ whenever $i < j$. That is, $u_n <_P u_{n-1} <_P \ldots <_P u_1$, i.e. P is a linear order. ◻

We introduce now the notion of a *linear-interval cover* of a bipartite graph. This notion is crucial for our main result of this section, cf. Theorem 4.8.

DEFINITION 4.6. *Let $G = (U, V, E)$ be a bipartite graph, where $U = \{u_1, u_2, \ldots, u_n\}$ and $V = \{v_1, v_2, \ldots, v_n\}$. Let $E_0 = \{u_i v_i \mid 1 \leq i \leq n\}$ and suppose that $E_0 \subseteq E$. Then, G is* linear-interval coverable *if there exist two chain graphs*

$G_1 = (U, V, E_1)$ *and* $G_2 = (U, V, E_2)$*, such that* $G = G_1 \cup G_2$ *and* $E_0 \subseteq E_2 \setminus E_1$*. In this case, the sets* $\{E_1, E_2\}$ *are a* linear-interval cover *of* G*.*

Before we proceed with Theorem 4.8, we first provide the next auxiliary lemma.

LEMMA 4.7. *Let* $Q_1 = (U, R_1)$ *be an interval order and* $Q_2 = (U, R_2)$ *be a partial order, such that* Q_1 *and* Q_2 *do not contradict each other. Then there exists a linear order* Q_0 *that is a linear extension of both* Q_1 *and* Q_2*.*

Proof. Let $U = \{u_1, u_2, \ldots, u_n\}$ be the ground set of Q_1 and Q_2. Furthermore let $C(Q_1) = (U, V, E_1)$ be the domination bipartite graph of Q_1, where $V = \{v_1, v_2, \ldots, v_n\}$, cf. Definition 4.2. Since Q_1 is an interval order by assumption, $C(Q_1)$ is a chain graph by Lemma 4.3, i.e. $C(Q_1)$ does not contain an induced $2K_2$. Consider now two edges $u_i v_j$ and $u_k v_\ell$ of $C(Q_1)$, where $\{i, j\} \cap \{k, \ell\} = \emptyset$. Then $u_i <_{Q_1} u_j$ and $u_k <_{Q_1} u_\ell$ by Definition 4.2. Furthermore, at least one of the edges $u_i v_\ell$ and $u_k v_j$ exists in $C(Q_1)$, since otherwise the edges $u_i v_j$ and $u_k v_\ell$ induce a $2K_2$ in $C(Q_1)$, which is a contradiction. Therefore $u_i <_{Q_1} u_\ell$ or $u_k <_{Q_1} u_j$.

Since Q_1 and Q_2 do not contradict each other by assumption, we can define the simple directed graph $G_0 = (U, E)$, such that $\overrightarrow{u_i u_j} \in E$ if and only if $u_i <_{Q_1} u_j$ or $u_i <_{Q_2} u_j$. We will prove that G_0 is acyclic. Suppose otherwise that G_0 has at least one directed cycle, and let C be a directed cycle of G_0 with the smallest possible length. Assume first that C has length 3, and let its edges be $\overrightarrow{u_i u_j}$, $\overrightarrow{u_j u_k}$, and $\overrightarrow{u_k u_i}$. Then at least two of these edges belong to Q_1 or to Q_2. Let without loss of generality $\overrightarrow{u_i u_j}$ and $\overrightarrow{u_j u_k}$ belong to Q_1, i.e. $u_i <_{Q_1} u_j$ and $u_j <_{Q_1} u_k$. Then also $u_i <_{Q_1} u_k$, since Q_1 is transitive, and thus $\overrightarrow{u_i u_k} \in E$. This contradicts the assumption that $\overrightarrow{u_k u_i}$ is an edge of C. Assume now that C has length greater than 3. Suppose that two consecutive edges $\overrightarrow{u_i u_j}$ and $\overrightarrow{u_j u_k}$ of C belong to Q_1, i.e. $u_i <_{Q_1} u_j$ and $u_j <_{Q_1} u_k$. Then also $u_i <_{Q_1} u_k$, since Q_1 is transitive, and thus $\overrightarrow{u_i u_k} \in E$. Therefore we can replace in C the edges $\overrightarrow{u_i u_j}$ and $\overrightarrow{u_j u_k}$ by the edge $\overrightarrow{u_i u_k}$, obtaining thus a smaller directed cycle than C, which is a contradiction by the assumption on C. Thus no two consecutive edges of C belong to Q_1. Similarly, no two consecutive edges of C belong to Q_2, and thus the edges of C belong alternately to Q_1 and Q_2. In particular, C has even length.

Consider now three consecutive edges $\overrightarrow{u_i u_j}, \overrightarrow{u_j u_k}, \overrightarrow{u_k u_\ell}$ of C, where $\overrightarrow{u_i u_j}$ and $\overrightarrow{u_k u_\ell}$ belong to Q_1. Then $u_i <_{Q_1} u_j$ and $u_k <_{Q_1} u_\ell$, where $\{i, j\} \cap \{k, \ell\} = \emptyset$, and thus $u_i <_{Q_1} u_\ell$ or $u_k <_{Q_1} u_j$, as we proved above. That is, $\overrightarrow{u_i u_\ell} \in E$ or $\overrightarrow{u_k u_j} \in E$. Therefore, since we assumed that $\overrightarrow{u_j u_k}$ is an edge of C, it follows that $\overrightarrow{u_k u_j} \notin E$, and thus $\overrightarrow{u_i u_\ell} \in E$. Therefore, in particular, $\overrightarrow{u_\ell u_i} \notin E$, and thus C does not have length 4, i.e. it has length at least 6. Thus we can replace in C the edges $\overrightarrow{u_i u_j}, \overrightarrow{u_j u_k}, \overrightarrow{u_k u_\ell}$ by the edge $\overrightarrow{u_i u_\ell}$, obtaining thus a smaller directed cycle than C, which is a contradiction by the assumption on C.

Therefore, there exists no directed cycle in G_0, i.e. G_0 is a directed acyclic graph. Thus any topological ordering of G_0 corresponds to a linear order $Q_0 = (U, R_0)$ that is a linear extension of both Q_1 and Q_2. This completes the proof of the lemma. ∎

THEOREM 4.8. *Let* $P = (U, R)$ *be a partial order. In the bipartite complement* $\widehat{C}(P)$ *of the graph* $C(P)$*, denote* $E_0 = \{u_i v_i \mid 1 \leq i \leq n\}$*. The following statements are equivalent:*

(a) $P = P_1 \cap P_2$*, where* P_1 *is a linear order and* P_2 *is an interval order.*

(b) $\widehat{C}(P) = \widehat{NC}(P_1) \cup \widehat{C}(P_2)$ *for two partial orders* P_1 *and* P_2 *on* V*, where* $\widehat{NC}(P_1)$ *and* $\widehat{C}(P_2)$ *are chain graphs.*

(c) $\widehat{C}(P)$ *is linear-interval coverable, i.e.* $\widehat{C}(P) = G_1 \cup G_2$ *for two chain graphs* $G_1 = (U, V, E_1)$ *and* $G_2 = (U, V, E_2)$*, where* $E_0 \subseteq E_2 \setminus E_1$*.*

Proof. (a) $\Rightarrow$ (b). Since P_1 is a linear order, it follows by Lemma 4.5 that $NC(P_1)$ is a chain graph. Furthermore, sine P_2 is an interval order, it follows by Lemma 4.3 that $C(P_2)$ is a chain graph. Therefore, since the class of chain graphs is closed under bipartite complementation, it follows that $\widehat{NC}(P_1)$ and $\widehat{C}(P_2)$ are chain graphs.

Let $u_i, u_j \in U$ such that $u_iv_j \in E(C(P))$. Then $u_i <_P u_j$ by Definition 4.2. Furthermore, since $P = P_1 \cap P_2$ by assumption, it follows that $u_i <_{P_1} u_j$ and $u_i <_{P_2} u_j$, and thus also $u_iv_j \in E(NC(P_1))$ and $u_iv_j \in E(C(P_2))$ by Definitions 4.2 and 4.4, respectively. Therefore $C(P) \subseteq NC(P_1) \cap C(P_2)$.

Let now $u_i, u_j \in U$ such that $u_iv_j \in E(NC(P_1))$ and $u_iv_j \in E(C(P_2))$. Then, it follows in particular that $u_i \neq u_j$ (since otherwise $u_iv_j \notin E(C(P_2))$, a contradiction). Thus, $u_i <_{P_1} u_j$ and $u_i <_{P_2} u_j$ by Definitions 4.2 and 4.4. Therefore, since $P = P_1 \cap P_2$ by assumption, it follows that $u_i <_P u_j$, and thus $u_iv_j \in E(C(P))$ by Definition 4.2. That is, $NC(P_1) \cap C(P_2) \subseteq C(P)$. Summarizing, $C(P) = NC(P_1) \cap C(P_2)$, and thus also $\widehat{C}(P) = \widehat{NC}(P_1) \cup \widehat{C}(P_2)$.

(b) $\Rightarrow$ (a). Since $\widehat{C}(P) = \widehat{NC}(P_1) \cup \widehat{C}(P_2)$, it follows that $C(P) = NC(P_1) \cap C(P_2)$. Let $u_i, u_j \in U$ such that $u_i <_P u_j$. Then $u_iv_j \in E(C(P))$ by Definition 4.2. Therefore, since $C(P) = NC(P_1) \cap C(P_2)$, it follows that also $u_iv_j \in E(NC(P_1))$ and $u_iv_j \in E(C(P_2))$. Thus, in particular, $u_i \neq u_j$ (since otherwise $u_iv_j \notin E(C(P_2))$, a contradiction). Therefore $u_i <_{P_1} u_j$ and $u_i <_{P_2} u_j$ by Definitions 4.2 and 4.4. That is, $P \subseteq P_1 \cap P_2$.

Let now $u_i, u_j \in U$ such that $u_i <_{P_1} u_j$ and $u_i <_{P_2} u_j$. Then $u_iv_j \in E(NC(P_1))$ and $u_iv_j \in E(C(P_2))$ by Definitions 4.2 and 4.4. Therefore, since $C(P) = NC(P_1) \cap C(P_2)$, it follows that also $u_iv_j \in E(C(P))$. Thus $u_i <_P u_j$ by Definition 4.2. That is, $P_1 \cap P_2 \subseteq P$. Summarizing, $P = P_1 \cap P_2$. Furthermore, since by assumption $\widehat{NC}(P_1)$ and $\widehat{C}(P_2)$ are chain graphs, it follows that also $NC(P_1)$ and $C(P_2)$ are chain graphs. Therefore P_1 is a linear order and P_2 is an interval order by Lemmas 4.5 and 4.3, respectively.

(b) $\Rightarrow$ (c). Define $G_1 = \widehat{NC}(P_1)$ and $G_2 = \widehat{C}(P_2)$. Then, it follows by (b) that G_1 and G_2 are chain graphs and that $\widehat{C}(P) = G_1 \cup G_2$. Note now by Definitions 4.2 and 4.4 that $E_0 \cap E(C(P_2)) = \emptyset$ and that $E_0 \subseteq E(NC(P_1))$, respectively. Therefore $E_0 \subseteq E(\widehat{C}(P_2)) \setminus E(\widehat{NC}(P_1))$. Thus, since $E_2 = E(G_2) = E(\widehat{C}(P_2))$ and $E_1 = E(G_1) = E(\widehat{NC}(P_1))$, it follows that $E_0 \subseteq E_2 \setminus E_1$. That is, $\widehat{C}(P)$ is linear-interval coverable by Definition 4.6.

(c) $\Rightarrow$ (b). We will construct from the edge sets E_1 and E_2 of G_1 and G_2, respectively, a linear order P_1 and an interval order P_2, such that $\widehat{C}(P) = \widehat{NC}(P_1) \cup \widehat{C}(P_2)$. Denote first the bipartite complement $\widehat{G}_2$ of G_2 as $\widehat{G}_2 = (U, V, \widehat{E}_2)$. Note that $\widehat{G}_2$ is a chain graph, since G_2 is also a chain graph by assumption.

The interval order P_2. We define P_2, such that $u_i <_{P_2} u_j$ if and only if $u_iv_j \in \widehat{E}_2$. We will now prove that P_2 is a partial order. Recall that $E_0 \subseteq E_2$ by assumption, and thus $E_0 \cap \widehat{E}_2 = \emptyset$. That is, $u_iv_i \notin \widehat{E}_2$ for every $i = 1, 2, \ldots, n$. Furthermore, $\widehat{G}_2$ is a chain graph, since G_2 is a chain graph by assumption. Therefore, for two distinct indices i, j, at most one of the edges u_iv_j and u_jv_i belongs to $\widehat{E}_2$, since otherwise these two edges would induce a $2K_2$ in $\widehat{G}_2$, which is a contradiction. Thus, according to our definition of P_2, whenever $i \neq j$, it follows that either $u_i <_{P_2} u_j$, or $u_j <_{P_2} u_i$, or u_i and u_j are incomparable in P_2. Suppose that $u_i <_{P_2} u_j$ and $u_j <_{P_2} u_k$ for three indices i, j, k. That is, $u_iv_j, u_jv_k \in \widehat{E}_2$ by definition of P_2. Since

$\widehat{G}_2 = (U, V, \widehat{E}_2)$ is a chain graph, the edges $u_i v_j$ and $u_j v_k$ do not build a $2K_2$ in $\widehat{G}_2$. Therefore, since $u_j v_j \notin \widehat{E}_2$, it follows that $u_i v_k \in \widehat{E}_2$, i.e. $u_i <_{P_2} u_k$. That is, P_2 is transitive, and thus P_2 is a partial order. Furthermore, note by the definition of P_2 and by Definition 4.2 that $\widehat{G}_2 = C(P_2)$. Therefore, since $\widehat{G}_2$ is a chain graph, it follows by Lemma 4.3 that P_2 is an interval order.

In order to define the linear order P_1, we first define two auxiliary orders Q_1 and Q_2, as follows.

The interval order Q_1. We define Q_1, such that $u_i <_{Q_1} u_j$ if and only if $u_i v_j \in E_1$. We will prove that Q_1 is a partial order. Recall that $E_0 \cap E_1 = \emptyset$ by assumption. That is, $u_i v_i \notin E_1$ for every $i = 1, 2, \ldots, n$. Furthermore, for two distinct indices i, j, at most one of the edges $u_i v_j$ and $u_j v_i$ belongs to E_1. Indeed, otherwise these two edges would induce a $2K_2$ in G_1, which is a contradiction since G_1 is a chain graph by assumption. Thus, according to our definition of Q_1, whenever $i \neq j$, it follows that either $u_i <_{Q_1} u_j$, or $u_j <_{Q_1} u_i$, or u_i and u_j are incomparable in Q_1. Suppose that $u_i <_{Q_1} u_j$ and $u_j <_{Q_1} u_k$ for three indices i, j, k. That is, $u_i v_j, u_j v_k \in E_1$ by definition of Q_1. Since G_1 is a chain graph by assumption, the edges $u_i v_j$ and $u_j v_k$ do not build a $2K_2$ in G_1. Therefore, since $u_j v_j \notin E_1$, it follows that $u_i v_k \in E_1$, i.e. $u_i <_{Q_1} u_k$. That is, Q_1 is transitive, and thus Q_1 is a partial order. Furthermore, note by the definition of Q_1 and by Definition 4.2 that $G_1 = C(Q_1)$. Therefore Q_1 is an interval order by Lemma 4.3, since G_1 is a chain graph by assumption.

The partial order Q_2. We define the partial order Q_2 as the inverse partial order $\overline{P}$ of P. That is, $u_i <_{Q_2} u_j$ if and only if $u_j <_P u_i$. Note that Q_2 is transitive, since P is transitive.

Before we define the linear order P_1, we first prove that the partial orders Q_1 and Q_2 do not contradict each other. Suppose otherwise that $u_i <_{Q_1} u_j$ and $u_j <_{Q_2} u_i$, for some pair u_i, u_j. Then, since $u_i <_{Q_1} u_j$, it follows that $u_i v_j \in E_1$ by definition of Q_1. Therefore $u_i v_j \in E(\widehat{C}(P))$, since $\widehat{C}(P) = G_1 \cup G_2$ by assumption. On the other hand, since $u_j <_{Q_2} u_i$, it follows that $u_i <_P u_j$ by definition of Q_2. Therefore $u_i v_j \in E(C(P))$ by Definition 4.2, and thus $u_i v_j \notin E(\widehat{C}(P))$, which is a contradiction. Therefore the partial orders Q_1 and Q_2 do not contradict each other.

The linear order P_1. Since the interval order Q_1 and the partial order Q_2 do not contradict each other, we can construct by Lemma 4.7 a common linear extension Q_0 of Q_1 and Q_2. That is, if $u_i <_{Q_1} u_j$ or $u_i <_{Q_2} u_j$, then $u_i <_{Q_0} u_j$. We define now the linear order P_1 as the inverse linear order $\overline{Q_0}$ of Q_0. Note that P_1 is also a linear extension of P, since $u_i <_P u_j$ implies that $u_j <_{Q_2} u_i$, which in turn implies that $u_i <_{P_1} u_j$.

Now we prove that $\widehat{C}(P) \subseteq \widehat{NC}(P_1) \cup \widehat{C}(P_2)$. Let $u_i v_j \in E_1$. Then $u_i <_{Q_1} u_j$ by the definition of Q_1, and thus $u_j <_{P_1} u_i$ by the definition of P_1. Therefore $u_i \nleq_{P_1} u_j$, and thus $u_i v_j \notin E(NC(P_1))$ by Definition 4.4. Therefore $u_i v_j \in E(\widehat{NC}(P_1))$. Thus $E_1 \subseteq E(\widehat{NC}(P_1))$, i.e. $G_1 \subseteq \widehat{NC}(P_1)$. Recall now that $\widehat{C}(P) = G_1 \cup G_2$ by assumption. Furthermore recall that $\widehat{G}_2 = C(P_2)$ as we proved above, and thus $G_2 = \widehat{C}(P_2)$. Therefore, since $G_1 \subseteq \widehat{NC}(P_1)$, it follows that $\widehat{C}(P) \subseteq \widehat{NC}(P_1) \cup \widehat{C}(P_2)$.

Finally we prove that $C(P) \subseteq NC(P_1) \cap C(P_2)$. Consider now an edge $u_i v_j \in E(C(P))$. Then $u_i <_P u_j$ by Definition 4.2, and thus $u_j <_{Q_2} u_i$ by the definition of Q_2. Furthermore $u_i <_{P_1} u_j$ by the definition of P_1, and thus $u_i v_j \in E(NC(P_1))$ by Definition 4.4. Note now that $C(P) = \widehat{G}_1 \cap \widehat{G}_2$, since $\widehat{C}(P) = G_1 \cup G_2$ by assumption.

Therefore, since $u_iv_j \in E(C(P))$ by assumption, it follows that also $u_iv_j \in \widehat{E}_2$. That is, if $u_iv_j \in E(C(P))$ then $u_iv_j \in E(NC(P_1))$ and $u_iv_j \in \widehat{E}_2$. Therefore, since $\widehat{G}_2 = C(P_2)$, it follows that $C(P) \subseteq NC(P_1) \cap C(P_2)$.

Summarizing, since $\widehat{C}(P) \subseteq \widehat{NC}(P_1) \cup \widehat{C}(P_2)$ and $C(P) \subseteq NC(P_1) \cap C(P_2)$, it follows that $\widehat{C}(P) = \widehat{NC}(P_1) \cup \widehat{C}(P_2)$. This completes the proof of the theorem. $\square$

The next corollary follows now easily by Theorems 4.1 and 4.8.

Corollary 4.9. *Let $G = (V, E)$ be a cocomparability graph and P be a partial order of $\overline{G}$. Then, G is a PI graph if and only if the bipartite graph $\widehat{C}(P)$ is linear-interval coverable.*

We now present Algorithm 1, which constructs a PI representation R of a cocomparability graph G by a linear-interval cover $\{E_1, E_2\}$ of the bipartite graph $\widehat{C}(P)$ (cf. Definition 4.6). Since $E_0 \subseteq E_2 \setminus E_1$ by Definition 4.6, where $E_0 = \{u_iv_i \mid 1 \leq i \leq n\}$ and n is the number of vertices of G, note that $i \neq j$ during the execution of each of the lines 6, 8, and 10 of Algorithm 1.

Algorithm 1 Construction of a PI representation, given a linear-interval cover

Input: A cocomparability graph G, a partial order P of $\overline{G}$, the domination bipartite graph $C(P) = (U, V, E)$, and a linear-interval cover $\{E_1, E_2\}$ of $\widehat{C}(P)$
Output: A PI representation R of G

1: Let $U = \{u_1, u_2, \ldots, u_n\}$, $V = \{v_1, v_2, \ldots, v_n\}$
2: $Q_1 \leftarrow \emptyset$; $Q_2 \leftarrow \emptyset$; $P_2 \leftarrow \emptyset$
3: **for** $i = 1, 2, \ldots, n$ **do** {construction of the partial orders Q_1, Q_2, P_2}
4:   **for** $j = 1, 2, \ldots, n$ **do**
5:     **if** $u_iv_j \notin E_2$ **then** $\{i \neq j\}$
6:       $u_i <_{P_2} u_j$
7:     **if** $u_iv_j \in E_1$ **then** $\{i \neq j\}$
8:       $u_i <_{Q_1} u_j$
9:     **if** $u_j <_P v_i$ **then** $\{i \neq j\}$
10:       $u_i <_{Q_2} u_j$
11: Compute a linear extension Q_0 of $Q_1 \cup Q_2$
12: $P_1 \leftarrow \overline{Q_0}$
13: Place the elements of U on a line L_1 according to the linear order P_1
14: Place a set of n intervals on a line L_2 (parallel to L_1) according to the interval order P_2
15: Build the PI representation R of G by connecting the endpoints of the intervals on L_2 with the corresponding points on L_1
16: **return** R

Theorem 4.10. *Let G be a cocomparability graph with n vertices and P be the partial order of $\overline{G}$. Let $\{E_1, E_2\}$ be a linear-interval cover of $\widehat{C}(P)$. Then Algorithm 1 constructs in $O(n^2)$ time a PI representation R of G.*

Proof. Since $\widehat{C}(P)$ admits a linear-interval cover $\{E_1, E_2\}$, Corollary 4.9 implies that G is a PI graph. Furthermore, it follows by the proof of the implication ((c) $\Rightarrow$ (b)) in Theorem 4.8 that the partial orders P_1 and P_2 that are constructed in lines 3-12 of Algorithm 1 are a linear order and an interval order, respectively, such that $\widehat{C}(P) = \widehat{NC}(P_1) \cup \widehat{C}(P_2)$. Furthermore, it follows by the proof of the implication ((b) $\Rightarrow$ (a)) in Theorem 4.8 that $P = P_1 \cap P_2$ for these two partial orders. Once we have computed in lines 3-12 the linear order P_1 and the interval order P_2, for which $P = P_1 \cap P_2$, it is now straightforward to construct a PI representation R of G as follows (cf. also [5] and the proof of Theorem 4.1). We arrange a set of n points (resp. n intervals) on a line L_1 (resp. on a line L_2, parallel to L_1) according to the linear order P_1 (resp. to the interval order P_2). Then we connect the endpoints of the intervals on L_2 with the corresponding points on L_1. Regarding the time complexity, each of the lines 5-10 of Algorithm 1 can be executed in constant time, and thus the lines 3-10 can be executed in total $O(n^2)$ time. Furthermore, since the lines 11-15 can be executed in a trivial way in at most $O(n^2)$ time each, it follows that the running time of Algorithm 1 is $O(n^2)$. $\square$

5. Detecting linear-interval covers using Boolean satisfiability. The natural algorithmic question that arizes from the characterization of PI graphs using linear-interval covers in Corollary 4.9, is the following: "Given a cocomparability graph G and a partial order P of $\overline{G}$, can we efficiently decide whether the bipartite graph $\widehat{C}(P)$ has a linear-interval cover?" We will answer this algorithmic question in the affirmative in Section 6. In this section we translate *every* instance of this decision problem (i.e. whether the bipartite graph $\widehat{C}(P)$ has a linear-interval cover) to a restricted instance of 3SAT (cf. Theorem 5.4). That is, for every such a bipartite graph $\widehat{C}(P)$, we construct a Boolean formula ϕ in conjunctive normal form (CNF), with size polynomial on the size of $\widehat{C}(P)$ (and thus also on G), such that $\widehat{C}(P)$ has a linear-interval cover if and only if ϕ is satisfiable. In particular, this formula ϕ can be written as $\phi = \phi_1 \wedge \phi_2$, where ϕ_1 has three literals in every clause and ϕ_2 has two literals in every clause. Moreover, as we will prove in Section 6, the satisfiability problem can be efficiently decided on the formula ϕ, by exploiting an appropriate sub-formula of ϕ which is gradually mixed (cf. Definition 2.2).

In the remainder of the paper, given a cocomparability graph G and a partial ordering P of its complement $\overline{G}$, we denote by $\widetilde{G} = \widehat{C}(P)$ the bipartite complement of the domination bipartite graph $C(P)$ of P. Furthermore we denote by H the associated split graph of $\widetilde{G}$ and by H^* the conflict graph of H. Moreover, we assume in the remainder of the paper without loss of generality that $\chi(H^*) \leq 2$, i.e. that H^* is bipartite. Indeed, as we formally prove in Lemma 5.1, if $\chi(H^*) > 2$ then $\widetilde{G}$ does not have a linear-interval cover, i.e. G is not a PI graph. Note that every proper 2-coloring of the vertices of the conflict graph H^* corresponds to exactly one 2-coloring of the edges of H that includes no monochromatic AC_4. We assume in the following that a proper 2-coloring (with colors blue and red) of the vertices of H^* is given as input; note that χ_0 can be computed in polynomial time.

Lemma 5.1. *Let G be a cocomparability graph and P be a partial order of $\overline{G}$. Let $\widetilde{G} = \widehat{C}(P)$, H be the associated split graph of $\widetilde{G}$, and H^* be the conflict graph of H. If $\widetilde{G}$ is linear-interval coverable, then $\chi(H^*) \leq 2$.*

Proof. Suppose otherwise that $\chi(H^*) > 2$. Then $t(H) > 2$, since $\chi(H^*) \leq t(H)$ by Lemma 3.11. Therefore, Lemma 3.4 implies that $ch(\widetilde{G}) > 2$, and thus G is not a trapezoid graph [15]. Therefore G is clearly not a PI graph, and thus $\widetilde{G}$ is not linear-interval coverable by Corollary 4.9, which is a contradiction to the assumption of the lemma. Therefore $\chi(H^*) \leq 2$. $\square$

Let $C_1, C_2, \ldots, C_k$ be the connected components of H^*. Some of these components of H^* may be isolated vertices, which correspond to uncommitted edges in H. We assign to every component C_i, where $1 \leq i \leq k$, the Boolean variable x_i. Since H^* is bipartite by assumption, the vertices of each connected component C_i of H^* can be

Algorithm 2 Construction of the 3-CNF Boolean formula ϕ_1

Input: The bipartite graph $\widetilde{G} = \widehat{C}(P)$, the associated split graph H of $\widetilde{G}$, its conflict graph H^*, and a proper 2-coloring χ_0 of the vertices of H^*

Output: The 3-CNF Boolean formula ϕ_1

$\phi_1 \leftarrow \emptyset$
for all triples of edges $\{e, e', e''\} \subseteq E(H)$, such that $\{e, e', e''\}$ build an AC_6 in $E(H)$ **do** {note that this is an AC_6 in the graph H itself and not in a color subclass of its edges}
  if $\ell_e \neq \overline{\ell_{e'}}$, $\ell_{e'} \neq \overline{\ell_{e''}}$, and $\ell_e \neq \overline{\ell_{e''}}$ **then**
    if ϕ_1 does not contain $(\ell_e \vee \ell_{e'} \vee \ell_{e''})$ and $(\overline{\ell_e} \vee \overline{\ell_{e'}} \vee \overline{\ell_{e''}})$ **then**
      $\phi_1 \leftarrow \phi_1 \wedge (\ell_e \vee \ell_{e'} \vee \ell_{e''}) \wedge (\overline{\ell_e} \vee \overline{\ell_{e'}} \vee \overline{\ell_{e''}})$
return ϕ_1

partitioned into two color classes $S_{i,1}$ and $S_{i,2}$. Without loss of generality, we assume that $S_{i,1}$ (resp. $S_{i,2}$) contains the vertices of C_i that are colored red (resp. blue) in χ_0. Note that, since vertices of H^* correspond to edges of H (cf. Definition 3.6), for every two edges e and e' of H that are in conflict (i.e. $e||e'$) there exists an index $i \in \{1, 2, \ldots, k\}$ such that one of these edges belongs to $S_{i,1}$ and the other belongs to $S_{i,2}$. We now assign a *literal* ℓ_e to every edge e of H as follows: if $e \in S_{i,1}$ for some $i \in \{1, 2, \ldots, k\}$, then $\ell_e = x_i$; otherwise, if $e \in S_{i,2}$, then $\ell_e = \overline{x_i}$. Note that, by construction, whenever two edges are in conflict in H, their assigned literals are one the negation of the other.

Observation 4. *Every truth assignment τ of the variables $x_1, x_2, \ldots, x_k$ corresponds bijectively to a proper 2-coloring χ_τ (with colors blue and red) of the vertices of H^*, as follows: $x_i = 0$ in τ (resp. $x_i = 1$ in τ), if and only if all vertices of the component C_i have in χ_τ the same color as in χ_0 (resp. opposite color than in χ_0). In particular, $\tau = (0, 0, \ldots, 0)$ corresponds to the coloring χ_0.*

We now present the construction of the Boolean formulas ϕ_1 and ϕ_2 from the graphs H and H^*, cf. Algorithms 2 and 3, respectively.

Description of the 3-CNF formula ϕ_1: Consider an AC_6 in the split graph H, and let e, e', e'' be its three edges in H, such that no two literals among $\{\ell_e, \ell_{e'}, \ell_{e''}\}$ are one the negation of the other. According to Algorithm 2, the Boolean formula ϕ_1 has for this triple $\{e, e', e''\}$ of edges exactly the two clauses $\alpha = (\ell_e \vee \ell_{e'} \vee \ell_{e''})$ and $\alpha' = (\overline{\ell_e} \vee \overline{\ell_{e'}} \vee \overline{\ell_{e''}})$. It is easy to check by the assignment of literals to edges that the clause α (resp. the clause α') of ϕ_1 is false in a truth assignment τ of the variables if and only if all edges $\{e, e', e''\}$ are colored red (resp. blue) in the 2-edge-coloring χ_τ of H (cf. Observation 4), as the following observation states.

Observation 5. *Let τ be any truth assignment of the variables $x_1, x_2, \ldots, x_k$. Let $\{e_1, e_2, e_3\}$ be the edges of an AC_6 in H and let $\alpha = (\ell_{e_1} \vee \ell_{e_2} \vee \ell_{e_3})$ and $\alpha' = (\overline{\ell_{e_1}} \vee \overline{\ell_{e_2}} \vee \overline{\ell_{e_3}})$ be a the corresponding clauses in ϕ_1. This AC_6 is monochromatic in the coloring χ_τ if and only if $\alpha = 0$ or $\alpha' = 0$ in τ.*

Consider now another AC_6 of H on the edges $\{e_1, e_2, e_3\}$, in which at least one literal among $\{\ell_{e_1}, \ell_{e_2}, \ell_{e_3}\}$ is the negation of another literal, for example $\ell_{e_1} = \overline{\ell_{e_2}}$. Then, for *any* proper 2-coloring of the vertices of H^*, the edges e and e' of H receive different colors, and thus this AC_6 is not monochromatic. Thus the next observation

Algorithm 3 Construction of the 2-CNF Boolean formula ϕ_2

Input: The bipartite graph $\widetilde{G} = \widehat{C}(P)$, the associated split graph H of $\widetilde{G}$, its conflict graph H^*, and a proper 2-coloring χ_0 of the vertices of H^*

Output: The 2-CNF Boolean formula ϕ_2

Let $H = (U, V, E_H)$, where $U = \{u_1, u_2, \ldots, u_n\}$ and $V = \{v_1, v_2, \ldots, v_n\}$

$E_0 \leftarrow \{u_i v_i \mid 1 \leq i \leq n\}$; $E' \leftarrow E_H \setminus E_0$; $H' \leftarrow H - E_0$

$\phi_2 \leftarrow \emptyset$

for every pair $\{i, j\} \subseteq \{1, 2, \ldots, n\}$ with $u_i v_j \notin E'$ **do**

  for $t = 1, 2, \ldots, n$ **do**

    if $u_i v_t, u_t v_j \in E'$ **then** {the edges $u_i v_t, u_t v_j$ are in conflict in H' but not in H}

      $e \leftarrow u_i v_t$; $e' \leftarrow u_t v_j$; $\phi_2 \leftarrow \phi_2 \wedge (\ell_e \vee \ell_{e'})$

return ϕ_2

follows by Observation 5.

Observation 6. *The formula ϕ_1 is satisfied by a truth assignment τ if and only if the corresponding 2-coloring χ_τ of the edges of H does not contain any monochromatic AC_6.*

Description of the 2-CNF formula ϕ_2: Denote for simplicity $H = (U, V, E_H)$, where $U = \{u_1, u_2, \ldots, u_n\}$ and $V = \{v_1, v_2, \ldots, v_n\}$. Furthermore denote $E_0 = \{u_i v_i \mid 1 \leq i \leq n\}$. Let $E' = E_H \setminus E_0$ and $H' = H - E_0$, i.e. H' is the split graph that we obtain if we remove from H all edges of E_0. Consider now a pair of edges $e = u_i v_t$ and $e' = u_t v_j$ of E', such that $u_i v_j \notin E'$. Note that i and j may be equal. However, since $E' \cap E_0 = \emptyset$, it follows that $i \neq t$ and $t \neq j$. Moreover, since the edge $u_t v_t$ belongs to E_H but not to E', it follows that the edges e and e' are in conflict in H' but not in H (for both cases where $i = j$ and $i \neq j$). That is, although e and e' are two non-adjacent vertices in the conflict graph H^* of H, they are adjacent vertices in the conflict graph of H'. For both cases where $i = j$ and $i \neq j$, an example of such a pair of edges $\{e, e'\}$ is illustrated in Figure 3. According to Algorithm 3, for every such pair $\{e, e'\}$ of edges in H, the Boolean formula ϕ_2 has the clause $(\ell_e \vee \ell_{e'})$. It is easy to check by the assignment of literals to edges of H that this clause $(\ell_e \vee \ell_{e'})$ of ϕ_2 is false in the truth assignment τ if and only if both e and e' are colored *red* in the 2-edge coloring χ_τ of H.

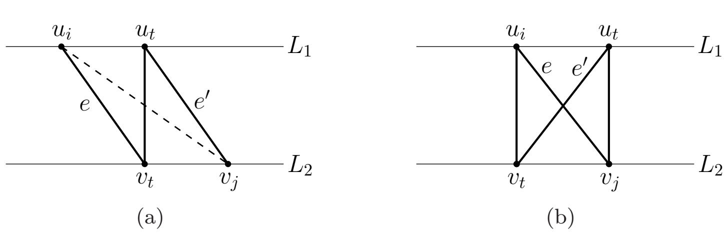

Figure 3. *Two edges $e = u_i v_t$ and $e' = u_t v_j$ of H, for which the formula ϕ_2 has the clause $(\ell_e \vee \ell_{e'})$, in the case where (a) $i \neq j$ and (b) $i = j$.*

Now we provide the main result of this section in Theorem 5.4, which relates the existence of a linear-interval cover in $\widetilde{G} = \widehat{C}(P)$ with the Boolean satisfiability of the formula $\phi_1 \wedge \phi_2$. Before we present Theorem 5.4, we first provide two auxiliary lemmas.

LEMMA 5.2. *Let G be a cocomparability graph and P be a partial order of $\overline{G}$. Let $\widetilde{G} = \widehat{C}(P)$, H be the associated split graph of $\widetilde{G}$, and H^* be the conflict graph of H. Denote $\widetilde{G} = (U, V, \widetilde{E})$ and $E_0 = \{u_i v_i \mid 1 \le i \le n\}$. Then, every $e \in E_0$ is an isolated vertex of H^*.*

Proof. Note by Definition 3.3 that $H = (U \cup V, E_H)$, where $E_H = \widetilde{E} \cup (V \times V)$. Furthermore all edges of $V \times V$ in E_H correspond to isolated vertices in the conflict graph H^* of H by Observations 2 and 3. Therefore all non-isolated vertices in H^* correspond to edges of $\widetilde{G}$ (i.e. they do not belong to $V \times V$). Consider now an edge $e_i = u_i v_i \in E_0 \subseteq \widetilde{E}$, where $1 \le i \le n$. Suppose that e_i is not an isolated vertex in the conflict graph H^*. Then the edge e_i of $\widetilde{G}$ builds with another edge $e = u_j v_k$ an induced AC_4 in H, i.e. $e_i = u_i v_i$ and $e = u_j v_k$ induce a $2K_2$ in $\widetilde{G}$. Therefore $u_j v_i, u_i v_k \notin \widetilde{E}$, i.e. $u_j v_i, u_i v_k \in E(C(P))$. Thus $u_j <_P u_i$ and $u_i <_P u_k$ by Definition 4.2. Therefore, since P is transitive (as a partial order), it follows that $u_j <_P u_k$, and thus $u_j v_k \in E(C(P))$, i.e. $u_j v_k \notin \widetilde{E}$. This is a contradiction, since we assumed that $e = u_j v_k$ is an edge of $\widetilde{G}$, i.e. $u_j v_k \in \widetilde{E}$. Therefore, $e_i = u_i v_i$ is an isolated vertex of H^*. ∎

LEMMA 5.3. *Let H be a split graph and H^* be the conflict graph of H, where H^* is bipartite with color classes E_1 and E_2. Let the vertices $v_1, \ldots, v_6$ of H build an AC_6 on the edges of E_i, where $i \in \{1, 2\}$. Then the edges $v_3 v_6, v_4 v_1, v_5 v_2$ exist in H and $v_4 v_5 || v_3 v_6$, $v_2 v_3 || v_4 v_1$, and $v_6 v_1 || v_5 v_2$.*

Proof. Since H is a split graph, Lemma 3.15 implies that H does not contain any AP_5 or any double AP_6. Therefore, the AC_6 of H is an AP_6, i.e. an alternating path of length 6, cf. Figure 1(b). Since E_1 and E_2 are the two color classes of H^*, any two vertices e and e' of H^* in the set E_i, where $i \in \{1, 2\}$, are not adjacent in H^*. Equivalently, any two edges e and e' of H in the set E_i are not in conflict, where $i \in \{1, 2\}$. Therefore, since by assumption, all edges $\{v_2 v_3, v_4 v_5, v_6 v_1\}$ of this AC_6 belong to the same color class E_i for some $i \in \{1, 2\}$, it follows that no pair of these edges is in conflict in H. Thus Lemma 3.7 implies that the edges $v_3 v_6, v_4 v_1, v_5 v_2$ exist in H and that $v_4 v_5 || v_3 v_6$, $v_2 v_3 || v_4 v_1$, and $v_6 v_1 || v_5 v_2$. ∎

We are now ready to provide Theorem 5.4.

THEOREM 5.4. *$\widetilde{G} = \widehat{C}(P)$ is linear-interval coverable if and only if $\phi_1 \wedge \phi_2$ is satisfiable. Given a satisfying assignment τ of $\phi_1 \wedge \phi_2$, Algorithm 4 computes a linear-interval cover of $\widetilde{G}$ in $O(n^2)$ time.*

Proof. Denote $\widetilde{G} = (U, V, \widetilde{E})$, where $U = \{u_1, u_2, \ldots, u_n\}$ and $V = \{v_1, v_2, \ldots, v_n\}$. Furthermore denote $H = (U, V, E_H)$, where $E_H = \widetilde{E} \cup (V \times V)$, cf. Definition 3.3. Let $E_0 = \{u_i v_i \mid 1 \le i \le n\}$. Since $\widetilde{G} = \widehat{C}(P)$, note by Definition 4.2 that $E_0 \subseteq \widetilde{E} \subseteq E_H$. Let χ_0 be the 2-coloring of the vertices of H^* (i.e. the edges of H) that is given as input to Algorithms 2 and 3. Moreover, let $C_1, C_2, \ldots, C_k$ be the connected components of H^*.

($\Rightarrow$) Suppose that $\widetilde{G}$ is linear-interval coverable. That is, there exist by Definition 4.6 two chain graphs $G_1 = (U, V, E_1)$ and $G_2 = (U, V, E_2)$, such that $\widetilde{G} = G_1 \cup G_2$ and $E_0 \subseteq E_2 \setminus E_1$. Let $H_1 = (U, V, E_{H_1})$ and $H_2 = (U, V, E_{H_2})$ be the associated split graphs of G_1 and G_2, respectively. Note that $H = H_1 \cup H_2$ and $E_0 \subseteq E_{H_2} \setminus E_{H_1}$. Since G_1 and G_2 are chain graphs, i.e. $ch(G_1) = ch(G_2) = 1$, Lemma 3.4 implies that $t(H_1) = t(H_2) = 1$, i.e. H_1 and H_2 are threshold graphs. Therefore, neither H_1 nor H_2 includes an AC_4.

Recall that the formulas ϕ_1 and ϕ_2 have one Boolean variable x_i for every connected component C_i of H^*, $i = 1, 2, \ldots, k$. We construct a 2-coloring χ_H of the edges of H as follows. For every edge e of H (i.e. a vertex of H^*), if $e \in E_{H_1}$ then we color e red in χ_H; otherwise, if $e \in E_{H_2} \setminus E_{H_1}$ then we color e blue in χ_H. Recall that $E_0 \subseteq E_{H_2} \setminus E_{H_1}$, and thus all edges of E_0 are colored blue in χ_H. Since both H_1 and H_2 do not include any AC_4, it follows by the definition of χ_H that there exists no monochromatic AC_4 in χ_H. Therefore, every two edges e and e' of H, which correspond to adjacent vertices in H^*, have different colors in χ_H, and thus χ_H constitutes a proper 2-coloring of the vertices of H^*. Therefore the coloring χ_H of the edges of H (i.e. vertices of H^*) defines a truth assignment τ of the variables $x_1, x_2, \ldots, x_k$ as follows (cf. Observation 4). For every connected component C_i of H^*, where $1 \leq i \leq k$, we define $x_i = 1$ (resp. $x_i = 0$) in τ if all vertices of C_i have in χ_H different (resp. the same) color as in χ_0. We will now prove that τ satisfies both formulas ϕ_1 and ϕ_2.

Satisfaction of the Boolean formula ϕ_1. Let α be a clause of ϕ_1. Recall that α corresponds to some triple $\{e, e', e''\}$ of edges of H that builds an AC_6 in H (cf. lines 2-5 of Algorithm 2). In particular, either $\alpha = (\ell_e \vee \ell_{e'} \vee \ell_{e''})$ or $\alpha = (\overline{\ell_e} \vee \overline{\ell_{e'}} \vee \overline{\ell_{e''}})$, where $\ell_e, \ell_{e'}, \ell_{e''}$ are the literals that have been assigned to the edges e, e', e'', respectively. Then, it follows from the description of the formula ϕ_1 (cf. also Observation 5) that the clause $(\ell_e \vee \ell_{e'} \vee \ell_{e''})$ (resp. the clause $(\overline{\ell_e} \vee \overline{\ell_{e'}} \vee \overline{\ell_{e''}})$) is not satisfied in the truth assignment τ if and only if the edges e, e', e'' of H are all red (resp. all blue) in χ_H.

Let $\alpha = (\ell_e \vee \ell_{e'} \vee \ell_{e''})$ (resp. $\alpha = (\overline{\ell_e} \vee \overline{\ell_{e'}} \vee \overline{\ell_{e''}})$). Suppose that α is not satisfied by τ, and thus the edges e, e', e'' of H are all red (resp. blue) in χ_H. Therefore all edges e, e', e'' belong to E_{H_1} (resp. to $E_{H_2} \setminus E_{H_1}$, and thus to E_{H_2}) by the definition of χ_H. Thus H has an AC_6 on the edges e, e', e'', which belong to H_1 (resp. to H_2). Therefore H_1 (resp. H_2) does not have a threshold completion in H by Lemma 3.8. This is a contradiction, since H_1 (resp. H_2) is a threshold graph. Therefore the clause $\alpha = (\ell_e \vee \ell_{e'} \vee \ell_{e''})$ (resp. $\alpha = (\overline{\ell_e} \vee \overline{\ell_{e'}} \vee \overline{\ell_{e''}})$) of ϕ_1 is satisfied by the truth assignment τ, and thus τ satisfies ϕ_1.

Satisfaction of the Boolean formula ϕ_2. Let $\alpha = (\ell_e \vee \ell_{e'})$ be a clause of ϕ_2. Recall that α corresponds to some pair of edges $e = u_i v_t$ and $e' = u_t v_j$ of $E_H \setminus E_0$, such that $u_i v_j \notin E_H \setminus E_0$ (cf. lines 4-7 of Algorithm 3). Therefore, since $u_t v_t \in E_0$, it follows that the edges $\{e, e'\}$ build an AC_4 in $H - E_0$ but not in H. Suppose that the clause $\alpha = (\ell_e \vee \ell_{e'})$ of ϕ_2 is not satisfied by the truth assignment τ, i.e. $\ell_e = \ell_{e'} = 0$ in τ. Then, it follows from the description of the formula ϕ_2 that both e and e' are colored red in the 2-edge coloring χ_H of H. Therefore both edges e and e' belong to H_1 by the definition of χ_H. However, as we noticed above, the edges $\{e, e'\}$ build an AC_4 in $H - E_0$, and thus they also build an AC_4 in $H_1 \subseteq H - E_0$. This is a contradiction by Corollary 3.10, since H_1 is a threshold graph. Therefore the clause $\alpha = (\ell_e \vee \ell_{e'})$ of ϕ_2 is satisfied by the truth assignment τ, and thus τ satisfies ϕ_2.

($\Leftarrow$) Suppose that $\phi_1 \wedge \phi_2$ is satisfiable, and let τ be a satisfying truth assignment of $\phi_1 \wedge \phi_2$. Recall that the formulas ϕ_1 and ϕ_2 have one Boolean variable x_i for every connected component C_i of H^*, $i = 1, 2, \ldots, k$. First, given the truth assignment τ, we construct the 2-coloring χ_τ of the vertices of H^* according to Observation 4. This 2-coloring of the vertices of H^* defines also a corresponding 2-coloring of the edges of H. Since ϕ_1 is satisfied by τ, it follows by Observation 6 that, in the coloring χ_τ of its edges, H does not contain any monochromatic AC_6. Therefore Theorem 3.13 implies that H does not contain any monochromatic AC_{2k} in χ_τ, where $k \geq 3$.

The vertex coloring χ'_τ of H^*. Now we modify the coloring χ_τ to the coloring

χ'_τ, as follows. For every trivial connected component C_i of H^* (i.e. when C_i has exactly one vertex), we color the vertex of C_i blue in χ'_τ, regardless of the color of C_i in χ_τ. On the other hand, for every non-trivial connected component C_i of H^* (i.e. when C_i has at least two vertices), the vertices of C_i have the same color in both χ_τ and χ'_τ. This new 2-coloring of the vertices of H^* defines also a corresponding 2-coloring of the edges of H. Note in particular by Lemma 5.2 that all edges of E_0 are colored blue in χ'_τ. Denote by E_{H_1} and E_{H_2} the sets of red and blue edges of H in χ'_τ, respectively. Note that $E_0 \subseteq E_{H_2}$. Moreover note that H does not have any AC_4 on the vertices of E_{H_1}, or on the vertices of E_{H_2}, since χ'_τ is a proper 2-coloring of the vertices of H^*. Define the subgraphs $H_1 = (U, V, E_{H_1})$ and $H_2 = (U, V, E_{H_2})$ of H. Note that $H = H_1 \cup H_2$.

***H_2* has a threshold completion in *H*.** Suppose now that H has an AC_{2k} on the edges of E_{H_2}, for some $k \geq 3$. Then Theorem 3.13 implies that H has also an AC_6 on the edges of E_{H_2}, i.e. H has an AC_6, in which all three edges are blue in χ'_τ. Since H does not have any monochromatic AC_6 in χ_τ, it follows that for at least one of the edges e of the blue AC_6 of H in χ'_τ, the color of e is different in χ_τ and in χ'_τ. Therefore, it follows by the construction of χ'_τ from χ_τ that the vertex of H^* that corresponds to e is an isolated vertex in H^*. That is, the edge e is uncommitted in H. This is a contradiction by Lemma 5.3, since e has been assumed to be an edge of a monochromatic AC_6 of H in χ'_τ. Therefore H does not have any AC_{2k} on the edges of E_{H_2}, where $k \geq 3$. Thus, since H does not have any AC_4 on the vertices of E_{H_2}, it follows that H does not have any AC_{2k} on the edges of E_{H_2}, where $k \geq 2$. Therefore H_2 has a threshold completion in H by Lemma 3.8.

***H_1* has a threshold completion in *$H - E_0$*.** Denote now $H' = H - E_0$. We will prove that H_1 has a threshold completion in H'. To this end, it suffices to prove by Lemma 3.8 that H' does not have any AC_{2k} on the edges of E_{H_1}, where $k \geq 2$.

For the sake of contradiction, suppose that H' includes an AC_4 on the edges of E_{H_1}. That is, there exist two edges $e, e' \in E_{H_1}$ that are in conflict in H'. Note by the definition of E_{H_1} that the edges e and e' are colored red in χ'_τ, and thus they are also colored red in χ_τ. If the edges $\{e, e'\}$ also build an AC_4 in H (i.e. before the removal of E_0), then the vertices e and e' of H^* are adjacent in H^*, and thus the edges e and e' of H have different colors in χ_τ, which is a contradiction. Thus the edges $\{e, e'\}$ are in conflict in H' but not in H. Recall now that for every such a pair $\{e, e'\}$ of edges of H' there exists a clause $\alpha = (\ell_e \vee \ell_{e'})$ in the formula ϕ_2 (cf. lines 4-7 of Algorithm 3). It follows from the description of the formula ϕ_2 that the clause α is not satisfied by the truth assignment τ if and only if both edges e, e' in H are red in χ_τ. However, since τ is a satisfying assignment of ϕ_2, every clause of ϕ_2 is satisfied by τ. Therefore at least one of the edges e and e' is colored blue in χ_τ, which is a contradiction. Therefore H' does not include any AC_4 on the edges of E_{H_1}.

Suppose now that H' includes an AC_{2k} on the edges of E_{H_1}, where $k \geq 3$. Consider the smallest such AC_{2k} on the edges of E_{H_1}, i.e. an AC_{2k} with the smallest $k \geq 3$. Let $w_1, w_2, \ldots, w_{2k}$ be the vertices of H' that build this AC_{2k}. Note by the definition of E_{H_1} that all edges of this AC_{2k} are colored red in the coloring χ'_τ, and thus they are also colored red in the coloring χ_τ. However, as we proved above, in the coloring χ_τ of its edges, H does not contain any monochromatic AC_{2k}, where $k \geq 3$. Therefore, at least one of the non-edges of the AC_{2k} in the graph H' is an edge of E_0 in the graph H. Assume without loss of generality that this edge of E_0 is w_1w_2. That is, assume that $w_1w_2 \in E_0$, i.e. $w_1w_2 = u_iv_i$ for some $i \in \{1, 2, \ldots, n\}$.

Suppose that w_3w_{2k} is not an edge of H'. Then, since $w_1w_2 \in E_0$, there exists (similarly to above) a clause α in the formula ϕ_2 such that α is not satisfied by the truth assignment τ if and only if both edges w_2w_3 and $w_{2k}w_1$ are colored red in χ_τ. However, τ is a satisfying truth assignment of ϕ_2 by assumption, and thus at least one edge of w_2w_3 and $w_{2k}w_1$ is colored blue in χ_τ, which is a contradiction. Therefore w_3w_{2k} is an edge of H'. Suppose now that the edge w_3w_{2k} of H' is colored red in χ'_τ, and thus $w_3w_{2k} \in E_{H_1}$ by the definition of E_{H_1}. Then the vertices $w_3, w_4, \ldots, w_{2k}$ build an AC_{2k-2} in H' on the edges of E_{H_1}, which is a contradiction to the minimality assumption of the AC_{2k} in H'. Therefore the edge w_3w_{2k} of H' is colored blue in χ'_τ, and thus $w_3w_{2k} \in E_{H_2}$.

Recall now that both the edges w_2w_3 and $w_{2k}w_1$ of H' are red in χ'_τ. Therefore, by the definition of the coloring χ'_τ from χ_τ, it follows that each of the edges w_2w_3 and $w_{2k}w_1$ participates to at least one AC_4 in H (or equivalently the corresponding vertices of w_2w_3 and $w_{2k}w_1$ in H^* are not isolated vertices). Let the edges w_2w_3 and $w'_2w'_3$ form an AC_4 in H, for some vertices w'_2 and w'_3, where $w_2w'_2$ and $w_3w'_3$ are not edges in H. Similarly, let the edges $w_{2k}w_1$ and $w'_{2k}w'_1$ form an AC_4 in H, for some vertices w'_{2k} and w'_1, where $w_{2k}w'_{2k}$ and $w_1w'_1$ are not edges in H. Note that some of the vertices $\{w'_2, w'_3, w'_{2k}, w'_1\}$ may coincide with each other, as well as with some of the vertices $\{w_2, w_3, w_{2k}, w_1\}$. Recall that χ'_τ is a proper 2-coloring of the vertices of H^*. Therefore, since w_2w_3 and $w_{2k}w_1$ are colored red in χ'_τ, it follows that $w'_2w'_3$ and $w'_{2k}w'_1$ are colored blue in χ'_τ. Therefore the vertices $w_1, w_2, w'_2, w'_3, w_3, w_{2k}, w'_{2k}, w'_1$ build an AC_8 in H on the edges of E_{H_2}. This is a contradiction, since we proved above that H does not have any AC_{2k} on the edges of E_{H_2}, where $k \geq 2$.

Therefore, it follows that H' does not include any AC_{2k} on the edges of E_{H_1}, where $k \geq 3$. Thus, since we already proved that H' does not include any AC_4 on the edges of E_{H_1}, it follows that H' does not include any AC_{2k} on the edges of E_{H_1}, where $k \geq 2$. Therefore H_1 has a threshold completion in $H' = H - E_0$ by Lemma 3.8.

Summarizing, H_1 has a threshold completion in $H' = H - E_0$, and H_2 has a threshold completion in H. Furthermore all edges of E_0 belong to the graph H, and $H = H_1 \cup H_2$. Let $\widetilde{H}_1$ be the threshold completion of H_1 in $H - E_0$, and let $\widetilde{H}_2$ be the threshold completion of H_2 in H. Then $\widetilde{H}_1$ and $\widetilde{H}_2$ are two threshold graphs, i.e. they do not include any AC_4. Furthermore, let $\widetilde{G}_1 = (U, V, \widetilde{E}_1)$ and $\widetilde{G}_2 = (U, V, \widetilde{E}_2)$ be the bipartite graphs obtained by $\widetilde{H}_1$ and $\widetilde{H}_2$, respectively, by removing from them all possible edges of $V \times V$. Note that $E_0 \subseteq \widetilde{E}_2 \setminus \widetilde{E}_1$, since every edge of E_0 belongs to $\widetilde{H}_2$ and not to $\widetilde{H}_1$. Furthermore, neither $\widetilde{G}_1$ nor $\widetilde{G}_2$ include any induced $2K_2$, since $\widetilde{H}_1$ and $\widetilde{H}_2$ do not include any AC_4. Therefore both $\widetilde{G}_1$ and $\widetilde{G}_2$ are chain graphs. Moreover, since $H = H_1 \cup H_2$, it follows that also $H = \widetilde{H}_1 \cup \widetilde{H}_2$ and $\widetilde{G} = \widetilde{G}_1 \cup \widetilde{G}_2$. Thus, since $E_0 \subseteq \widetilde{E}_2 \setminus \widetilde{E}_1$, it follows that $\widetilde{G}$ is linear-interval coverable by Definition 4.6 and $\{\widetilde{E}_1, \widetilde{E}_2\}$ is a linear-interval cover of $\widetilde{G}$. This construction of $\{\widetilde{E}_1, \widetilde{E}_2\}$ from the satisfying truth assignment τ of $\phi_1 \wedge \phi_2$ is shown in Algorithm 4.

Running time of Algorithm 4. First note that, since $|U| = |V| = n$, the split graph H has $O(n^2)$ edges. Therefore, since each edge of H is processed exactly once in the execution of lines 3-8 in Algorithm 4, these lines are executed in $O(n^2)$ time in total. Similarly, each of the lines 9, 10, and 13 is executed in $O(n^2)$ time. Now, each of the lines 11 and 12 is executed by Lemma 3.9 in time linear to the size of H, i.e. in $O(n^2)$ time each. Therefore the total running time of Algorithm 4 is $O(n^2)$. This completes the proof of the theorem. □

6. The recognition of linear-interval orders and PI graphs. In this section we investigate the structure of the formula $\phi_1 \wedge \phi_2$ that we computed in Section 5.

Algorithm 4 Construction of a linear-interval cover of $\widetilde{G} = \widehat{C}(P)$, if $\phi_1 \wedge \phi_2$ is satisfiable

Input: The bipartite graph $\widetilde{G} = \widehat{C}(P)$, the associated split graph H of $\widetilde{G}$, its conflict graph H^*, a proper 2-coloring χ_0 of the vertices of H^*, and a satisfying truth assignment τ of $\phi_1 \wedge \phi_2$

Output: A linear-interval cover $\{\widetilde{E}_1, \widetilde{E}_2\}$ of $\widetilde{G}$

Let $H = (U, V, E_H)$, where $U = \{u_1, u_2, \ldots, u_n\}$ and $V = \{v_1, v_2, \ldots, v_n\}$
$E_0 \leftarrow \{u_i v_i \mid 1 \leq i \leq n\}$
for every connected component $C_i, 1 \leq i \leq k$, of H^* **do**
  if C_i is an isolated vertex of H^* **then**
    color the vertex of C_i blue
  else
    if $x_i = 0$ in τ **then** color every vertex of C_i with the same color as in χ_0
    if $x_i = 1$ in τ **then** color every vertex of C_i with the opposite color than in χ_0
$E_{H_1} \leftarrow \{e \in E_H \mid e \text{ is red}\}$; $H_1 \leftarrow (U, V, E_{H_1})$
$E_{H_2} \leftarrow \{e \in E_H \mid e \text{ is blue}\}$; $H_2 \leftarrow (U, V, E_{H_2})$
Compute a threshold completion $\widetilde{H}_1$ of H_1 in $H - E_0$ (by Lemma 3.9)
Compute a threshold completion $\widetilde{H}_2$ of H_2 in H (by Lemma 3.9)
$\widetilde{E}_1 \leftarrow E(\widetilde{H}_1) \setminus (V \times V)$; $\widetilde{E}_2 \leftarrow E(\widetilde{H}_2) \setminus (V \times V)$
return $\{\widetilde{E}_1, \widetilde{E}_2\}$

In particular, we first prove in Section 6.1 some fundamental structural properties of $\phi_1 \wedge \phi_2$, which allow us to find an appropriate sub-formula of $\phi_1 \wedge \phi_2$ which is gradually mixed (cf. Definition 2.2). Then we exploit this sub-formula of $\phi_1 \wedge \phi_2$ in order to provide in Section 6.2 an algorithm that solves the satisfiability problem on $\phi_1 \wedge \phi_2$ in time linear to its size, cf. Theorem 6.8. Finally, using this satisfiability algorithm, we combine our results of Sections 4 and 5 in order to recognize efficiently PI graphs and linear-interval orders in Section 6.2.

6.1. Structural properties of the formula $\phi_1 \wedge \phi_2$. The three main structural properties of $\phi_1 \wedge \phi_2$ are proved in Lemmas 6.3, 6.5, and 6.6, respectively. We first provide two auxiliary technical lemmas.

LEMMA 6.1. *Let $\alpha = (\ell_1 \vee \ell_2 \vee \ell_3)$ be a clause of ϕ_1. Assume that α corresponds to the AP_6 of H on the vertices $v_1, \ldots, v_6$, which has the literals ℓ_1, ℓ_2, ℓ_3 on its edges (in this order). Then, for every edge e of H with literal $\ell_e = \ell_2$, there exists an AP_6 in H with $v_1 v_2$ as is its base and e as its ceiling, which has the literals ℓ_1, ℓ_2, ℓ_3 on its edges (in this order).*

Proof. First note that by the construction of ϕ_1 (cf. Section 5) no two literals among $\{\ell_1, \ell_2, \ell_3\}$ are one the negation of the other, i.e. $\ell_1 \neq \overline{\ell_2}$, $\ell_1 \neq \overline{\ell_3}$, and $\ell_2 \neq \overline{\ell_3}$. Therefore also no pair among the edges of the AP_6 on the vertices $v_1, \ldots, v_6$ is in conflict. Denote for simplicity $e' = v_4 v_5$. Since $\ell_{e'} = \ell_e = \ell_2$, the edges e' and e of H correspond to two vertices of the conflict graph H^* that lie in the same connected component of H^*. Thus there exists a path between these two vertices of H^*. That is, there exists a sequence of edges $e_1, e_2, \ldots, e_t$ in H, where $e_1 = e'$ and $e_t = e$, such that $e_i || e_{i+1}$ for every $i \in \{1, 2, \ldots, t-1\}$. Note that $\ell_{e_i} \in \{\ell_2, \overline{\ell_2}\}$ for all these edges

e_i. For every $1 \le i \le t$ denote $e_i = u_i w_i$, where $u_1 = v_4$ and $w_1 = v_5$. Furthermore let $u_i u_{i+1}$ and $w_i w_{i+1}$ be the non-edges between e_i and e_{i+1}, where $1 \le i \le t-1$. For simplicity of the presentation, denote $u_0 = v_3$ and $w_0 = v_6$.

We will prove by induction that for every $i \in \{1, 2, \ldots, t\}$ there exists an AP_6 in H on the vertices $v_1, v_2, u_{i-1}, u_i, w_i, w_{i-1}$ (if i is odd), or on the vertices $v_1, v_2, u_i, u_{i-1}, w_{i-1}, w_i$ (if i is even), which has the literals ℓ_1, ℓ_2, ℓ_3 on its edges (in this order). The induction basis (i.e. the case where $i = 1$) follows immediately by the assumption of the lemma.

For the induction step, let first $i \ge 2$ be even. Then $i-1$ is odd, and thus there exists by the induction hypothesis an AP_6 in H on the vertices $v_1, v_2, u_{i-2}, u_{i-1}, w_{i-1}, w_{i-2}$ which has the literals ℓ_1, ℓ_2, ℓ_3 on its edges (in this order). That is, $\ell_{v_2 u_{i-2}} = \ell_1$, $\ell_{u_{i-1} w_{i-1}} = \ell_2$, and $\ell_{w_{i-2} v_1} = \ell_3$. Therefore, since $\ell_{u_i w_i} \in \{\ell_2, \overline{\ell_2}\}$ and $u_i w_i || u_{i-1} w_{i-1}$ by assumption, it follows that $\ell_{u_i w_i} = \overline{\ell_2}$. Furthermore, since no pair among the edges of the AP_6 is in conflict, Lemma 3.7 implies in particular that the edges $v_1 u_{i-1}$ and $v_2 w_{i-1}$ exist in H and that $\ell_{v_1 u_{i-1}} = \overline{\ell_1}$ and $\ell_{v_2 w_{i-1}} = \overline{\ell_3}$.

Claim 1. *$v_1 \ne w_i$ and $v_2 \ne u_i$.*

Proof of Claim 1. Since H is a split graph, there exists a partition of its vertices into a clique K and an independent set I. Then, since H has an AP_6 on the vertices $v_1, v_2, u_{i-2}, u_{i-1}, w_{i-1}, w_{i-2}$, Lemma 3.14 implies that either $v_1, u_{i-2}, w_{i-1} \in K$ and $v_2, u_{i-1}, w_{i-2} \in I$, or $v_1, u_{i-2}, w_{i-1} \in I$ and $v_2, u_{i-1}, w_{i-2} \in K$. In the former case, since $w_{i-1} \in K$ and $w_{i-1} w_i$ is not an edge in H, it follows that $w_i \in I$. Thus $v_1 \ne w_i$, since $v_1 \in K$. Furthermore, since $w_i \in I$ and $u_i w_i$ is an edge in H, it follows that $u_i \in K$. Thus $v_2 \ne u_i$, since $v_2 \in I$. Similarly, in the latter case, since $u_{i-1} \in K$ and $u_{i-1} u_i$ is not an edge in H, it follows that $u_i \in I$. Thus $v_2 \ne u_i$, since $v_2 \in K$. Furthermore, since $u_i \in I$ and $u_i w_i$ is an edge in H, it follows that $w_i \in K$. Thus $v_1 \ne w_i$, since $v_1 \in I$. Summarizing, in both cases $v_1 \ne w_i$ and $v_2 \ne u_i$. ■

Suppose that $v_1 w_i$ is not an edge in H. Then $u_i w_i$ is in conflict with $v_1 u_{i-1}$, since also $u_{i-1} u_i$ is not an edge in H. Therefore $\ell_{u_i w_i} = \overline{\ell_{v_1 u_{i-1}}}$. Thus, since $\ell_{u_i w_i} = \overline{\ell_2}$ and $\ell_{v_1 u_{i-1}} = \overline{\ell_1}$, it follows that $\ell_1 = \overline{\ell_2}$, which is a contradiction, since no two literals among $\{\ell_1, \ell_2, \ell_3\}$ are one the negation of the other. Therefore $v_1 w_i$ is an edge in H. Furthermore $\ell_{v_1 w_i} = \ell_3$, since $\ell_{v_2 w_{i-1}} = \overline{\ell_3}$ and $w_{i-1} w_i, v_1 v_2$ are not edges in H. By symmetry it follows that also $v_2 u_i$ is an edge in H and that $\ell_{v_2 u_i} = \ell_1$. Thus the vertices $v_1, v_2, u_i, u_{i-1}, w_{i-1}, w_i$ build an AP_6 in H, which has the literals ℓ_1, ℓ_2, ℓ_3 on its edges (in this order). This completes the induction step whenever i is even.

Let now $i \ge 3$ be odd. Then $i-1$ is even, and thus there exists by the induction hypothesis an AP_6 in H on the vertices $v_1, v_2, u_{i-1}, u_{i-2}, w_{i-2}, w_{i-1}$ which has the literals ℓ_1, ℓ_2, ℓ_3 on its edges (in this order). That is, $\ell_{v_2 u_{i-1}} = \ell_1$, $\ell_{u_{i-2} w_{i-2}} = \ell_2$, and $\ell_{w_{i-1} v_1} = \ell_3$. Thus, since the edges $u_{i-2} w_{i-2}$ and $u_{i-1} w_{i-1}$ are in conflict by assumption, it follows that $\ell_{u_{i-1} w_{i-1}} = \overline{\ell_2}$. Furthermore, since the edges $u_{i-1} w_{i-1}$ and $u_i w_i$ are in conflict by assumption, it follows that $\ell_{u_i w_i} = \ell_2$. Thus the vertices $v_1, v_2, u_{i-1}, u_i, w_i, w_{i-1}$ build an AP_6 in H, which has the literals ℓ_1, ℓ_2, ℓ_3 on its edges (in this order). This completes the induction step whenever i is odd.

Summarizing, for $i = t$, there exists an AP_6 in H on the vertices $v_1, v_2, u_{t-1}, u_t, w_t, w_{t-1}$ (if t is odd), or on the vertices $v_1, v_2, u_t, u_{t-1}, w_{t-1}, w_t$ (if t is even), which has the literals ℓ_1, ℓ_2, ℓ_3 on its edges (in this order). In both cases where t is even or odd, this AP_6 has the non-edge $v_1 v_2$ as it base and the edge $e = u_t w_t$ as its ceiling. This completes the proof of the lemma. □

LEMMA 6.2. *Let $\alpha = (\ell_1 \vee \ell_2 \vee \ell_3)$ and $\beta = (\ell_1 \vee \ell_2 \vee \ell_4)$ be two clauses of ϕ_1 that share two literals ℓ_1 and ℓ_2. Then also $\ell_3 = \ell_4$.*

Proof. By the construction of the formula ϕ_1 (cf. Section 5), the clauses α and β correspond to two AC_6's in H. Since H is a split graph, Lemma 3.15 implies that each of these two AC_6's is an AP_6, i.e. an alternating path of length 6 (cf. Figure 1(b)). Let $v_1, v_2, v_3, v_4, v_5, v_6$ be the vertices of the first AP_6, which has the literals ℓ_1, ℓ_2, ℓ_3 on its edges (in this order). Note that, by the construction of ϕ_1, no two literals among $\{\ell_1, \ell_2, \ell_3\}$ are one the negation of the other, i.e. $\ell_1 \neq \overline{\ell_2}$, $\ell_1 \neq \overline{\ell_3}$, and $\ell_2 \neq \overline{\ell_3}$. Furthermore let $w_1, w_2, w_3, w_4, w_5, w_6$ be the vertices of the second AP_6, which has the literals ℓ_1, ℓ_2, ℓ_4 on its edges (in this order). Since H is a split graph, there exists a partition of its vertices into a clique K and an independent set I.

Consider now the base v_5v_6 and the ceiling v_2v_3 of the first AP_6 (cf. Definition 3.2). That is, the vertices of this AP_6 can be ordered as $v_5, v_6, v_1, v_2, v_3, v_4$ (where v_5v_6 is not an edge); then the literals on its edges are ℓ_3, ℓ_1, ℓ_2 (in this order). Since $\ell_{v_2v_3} = \ell_{w_2w_3} = \ell_1$, there exists by Lemma 6.1 an AP_6 with v_5v_6 as its base and w_2w_3 as its ceiling, which has the literals ℓ_3, ℓ_1, ℓ_2 on its edges (in this order). Note that the ordering of the vertices in this AP_6 can be either v_5, v_6, a, w_3, w_2, b, or v_5, v_6, a, w_2, w_3, b, for some vertices a and b of H. We distinguish now these two cases.

Case 1. The AP_6 with v_5v_6 as its base and w_2w_3 as its ceiling has vertex ordering v_5, v_6, a, w_3, w_2, b. Consider now the base aw_3 and the ceiling bv_5 of this AP_6. That is, its vertices can be ordered as a, w_3, w_2, b, v_5, v_6 (where aw_3 is not an edge); then the literals on its edges are ℓ_1, ℓ_2, ℓ_3 (in this order). Since $\ell_{bv_5} = \ell_{w_4w_5} = \ell_2$, there exists by Lemma 6.1 an AP_6 with aw_3 as its base and w_4w_5 as its ceiling, which has the literals ℓ_1, ℓ_2, ℓ_3 on its edges (in this order). Note that the ordering of the vertices in this AP_6 can be either a, w_3, c, w_5, w_4, d, or a, w_3, c, w_4, w_5, d, for some vertices c and d of H. We distinguish now these two cases.

Case 1.1. The AP_6 with aw_3 as its base and w_4w_5 as its ceiling has vertex ordering a, w_3, c, w_5, w_4, d. Since no two literals among $\{\ell_1, \ell_2, \ell_3\}$ are one the negation of the other, it follows that no pair among the edges of this AP_6 is in conflict. Thus Lemma 3.7 implies in particular that the edge w_3w_4 exists in H. This is a contradiction to our initial assumption that the vertices $w_1, w_2, w_3, w_4, w_5, w_6$ build an AC_6 (and thus w_3w_4 is not an edge).

Case 1.2. The AP_6 with aw_3 as its base and w_4w_5 as its ceiling has vertex ordering a, w_3, c, w_4, w_5, d. Then Lemma 3.14 implies that either $w_3 \in K$ and $w_5 \in I$, or $w_3 \in I$ and $w_5 \in K$. However, due to our initial assumption that the vertices $w_1, w_2, w_3, w_4, w_5, w_6$ build an AC_6, Lemma 3.14 implies that either $w_3, w_5 \in K$ or $w_3, w_5 \in I$, which is a contradiction.

Case 2. The AP_6 with v_5v_6 as its base and w_2w_3 as its ceiling has vertex ordering v_5, v_6, a, w_2, w_3, b. Consider now the base aw_2 and the ceiling bv_5 of this AP_6. That is, its vertices can be ordered as a, w_2, w_3, b, v_5, v_6 (where aw_2 is not an edge); then the literals on its edges are ℓ_1, ℓ_2, ℓ_3 (in this order). Since $\ell_{bv_5} = \ell_{w_4w_5} = \ell_2$, there exists by Lemma 6.1 an AP_6 with aw_2 as its base and w_4w_5 as its ceiling, which has the literals ℓ_1, ℓ_2, ℓ_3 on its edges (in this order). Note that the ordering of the vertices in this AP_6 can be either a, w_2, c, w_5, w_4, d, or a, w_2, c, w_4, w_5, d, for some vertices c and d of H. We distinguish now these two cases.

Case 2.1. The AP_6 with aw_2 as its base and w_4w_5 as its ceiling has vertex ordering a, w_2, c, w_5, w_4, d. Then Lemma 3.14 implies that either $w_2 \in K$ and $w_4 \in I$, or $w_2 \in I$ and $w_4 \in K$. However, due to our initial assumption that the vertices $w_1, w_2, w_3, w_4, w_5, w_6$ build an AC_6, Lemma 3.14 implies that either $w_2, w_4 \in K$ or $w_2, w_4 \in I$, which is a contradiction.

Case 2.2. The AP_6 with aw_2 as its base and w_4w_5 as its ceiling has vertex ordering a, w_2, c, w_4, w_5, d. Since no two literals among $\{\ell_1, \ell_2, \ell_3\}$ are one the negation of the other, it follows that no pair among the edges of this AP_6 is in conflict. Thus Lemma 3.7 implies in particular that the edge w_5w_2 exists in H and that $ad||w_5w_2$. Thus, since $\ell_{ad} = \ell_3$, it follows that $\ell_{w_5w_2} = \overline{\ell_3}$. Recall now that we initially assumed that the vertices $w_1, w_2, w_3, w_4, w_5, w_6$ build an AP_6 in H, which has the literals ℓ_1, ℓ_2, ℓ_4 on its edges (in this order). Similarly, Lemma 3.7 implies for this AP_6 that $w_6w_1||w_5w_2$. Thus, since $\ell_{w_6w_1} = \ell_4$, it follows that $\ell_{w_5w_2} = \overline{\ell_4}$. That is, $\ell_{w_5w_2} = \overline{\ell_3} = \overline{\ell_4}$, and thus $\ell_3 = \ell_4$. This completes the proof of the lemma. ☐

We are now ready to prove the three main structural properties of the formula $\phi_1 \wedge \phi_2$ in Lemmas 6.3, 6.5, and 6.6, respectively. The proof of the next lemma is a based on the results of [21].

LEMMA 6.3. *Let α and β be two clauses of ϕ_1. If α and β share at least one variable, then $\{\alpha, \overline{\alpha}\} = \{\beta, \overline{\beta}\}$.*

Proof. In Theorem 3.2 of [21], the authors consider an arbitrary graph G and its conflict graph G^*, which is bipartite. For every edge e of G, denote by $C^*(e)$ the connected component of G^* in which the vertex e belongs. For simplicity of the presentation, we will also refer in the following to $C^*(e)$ as the set of the corresponding edges in G. The authors of [21] assume an arbitrary 2-coloring of the vertices of G^* (i.e. of the edges of G), such that there is no monochromatic double AP_6, i.e. there is no double AP_6 on the edges of one edge-color class of G. Furthermore they assume that there is a monochromatic AP_6 in G on the vertices $v_1, \ldots, v_6$ (which is not a double AP_6). Since this AP_6 is monochromatic, it follows that no pair among its three edges is in conflict in G (since any two edges in conflict would have different colors). Thus the edges v_3v_6, v_4v_1, v_5v_2 exist in G and $v_4v_5||v_3v_6$, $v_2v_3||v_4v_1$, and $v_6v_1||v_5v_2$ by Lemma 3.7. The non-edge v_1v_2 is called the *base* of the AP_6 (cf. Definition 3.2); furthermore we call the edge v_3v_6 the *front* of the AP_6 [21]. Note here that the choice of the base v_1v_2 is arbitrary (the AP_6 has three bases v_1v_2, v_3v_4, and v_5v_6). Then, they prove[2] in Theorem 3.2 that, if we flip the colors of all edges of $C^*(v_3v_6)$ then in the new edge coloring of G no edge of $C^*(v_3v_6)$ participates in a monochromatic AP_6. Note furthermore that $v_4v_5 \in C^*(v_3v_6)$, since $v_4v_5||v_3v_6$, and thus also the color of v_4v_5 changes by flipping the colors of the edges in $C^*(v_3v_6)$.

We now apply the results of [21] in our case as follows. Consider two clauses α and β of ϕ_1 that share at least one variable. That is, each of the dual clauses $\{\alpha, \overline{\alpha}\}$ shares at least one literal with at least one of the dual clauses $\{\beta, \overline{\beta}\}$. If $\beta \in \{\alpha, \overline{\alpha}\}$ then clearly $\{\alpha, \overline{\alpha}\} = \{\beta, \overline{\beta}\}$, and thus the lemma follows.

Let now $\beta \notin \{\alpha, \overline{\alpha}\}$. Consider the AC_6 of H on the vertices $v_1, \ldots, v_6$ that corresponds to the dual clauses $\{\alpha, \overline{\alpha}\}$. Since H is a split graph, it follows by Lemma 3.15 that H does not contain any AP_5 or any double AP_6. Therefore this AC_6 of H on

[2]In [21], the authors prove within the proof of Theorem 3.2 a more general statement (cf. equations (2) and (3) in [21]). In particular, they flip the colors of all edges xy of G, for which there exists an AP_6 in G having v_1v_2 as its basis and xy as its front (cf. equation (2) in [21]); note here that all these edges, whose color is being flipped, may correspond to one or more connected components in the conflict graph G^*. Then they prove that in the new edge coloring of G no flipped edge participates in a monochromatic AP_6 (cf. equation (3) in [21]). In their proof, which is correct and technically involved, they actually prove that this happens also when we flip the colors of only one connected component $C^*(v_3v_6)$ of G^*, where v_3v_6 is the front of the initial monochromatic AP_6 on the vertices $v_1, \ldots, v_6$.

the vertices $v_1, \ldots, v_6$ is an AP_6 (but not a double AP_6). Let $e = v_2v_3$, $e' = v_4v_5$, and $e'' = v_6v_1$. This AP_6 has the non-edge v_1v_2 as its *base* and the edge v_3v_6 as its *front*, cf. Definition 3.2. Note that either $\alpha = (\ell_e \vee \ell_{e'} \vee \ell_{e''})$ and $\overline{\alpha} = (\overline{\ell_e} \vee \overline{\ell_{e'}} \vee \overline{\ell_{e''}})$, or $\alpha = (\overline{\ell_e} \vee \overline{\ell_{e'}} \vee \overline{\ell_{e''}})$ and $\overline{\alpha} = (\ell_e \vee \ell_{e'} \vee \ell_{e''})$. Assume without loss of generality that $\alpha = (\ell_e \vee \ell_{e'} \vee \ell_{e''})$ and $\overline{\alpha} = (\overline{\ell_e} \vee \overline{\ell_{e'}} \vee \overline{\ell_{e''}})$. Recall by our assumption that α shares at least one literal with at least one of the dual clauses $\{\beta, \overline{\beta}\}$. Assume without loss of generality that α shares at least one literal with β (the case where α shares at least one literal with $\overline{\beta}$ can be handled in exactly the same way). Furthermore, let without loss of generality $\ell_{e'}$ be the common literal of α and β, i.e. let $\beta = (\ell_{e'} \vee \ell_p \vee \ell_q)$.

Since α is a clause of ϕ_1, it follows by the construction of ϕ_1 that no two literals among $\{\ell_e, \ell_{e'}, \ell_{e''}\}$ are one the negation of the other (cf. lines 3-5 of Algorithm 2). Similarly no two literals among $\{\ell_{e'}, \ell_p, \ell_q\}$ are one the negation of the other, since β is a clause of ϕ_1. Consider now an *arbitrary* truth assignment τ of the variables $x_1, x_2, \ldots, x_k$, such that $\alpha = 0$ in τ, i.e. $\ell_e = \ell_{e'} = \ell_{e''} = 0$ in τ. Note that such an assignment exists, since no two literals among $\{\ell_e, \ell_{e'}, \ell_{e''}\}$ are one the negation of the other. Let χ be the 2-coloring of the vertices of H^* (i.e. of the edges of H) that corresponds to the truth assignment τ, cf. Observation 4. Since $\alpha = 0$ in the truth assignment τ, Observation 5 implies that the AP_6 on the vertices $v_1, \ldots, v_6$ is monochromatic in the edge-coloring χ of H. Then, due to the results of [21], if we flip in χ the colors of all edges of $C^*(v_3v_6)$, in the new edge coloring χ' of H no edge of $C^*(v_3v_6)$ participates in a monochromatic AP_6.

Let τ' be the truth assignment that corresponds to this new coloring χ' (cf. Observation 4). Then τ and τ' coincide on all variables except the variable of the component $C^*(v_3v_6)$ of H^*. Note that the color of $e' = v_4v_5$ has been flipped in the transition from χ' to χ, since $e' \in C^*(v_3v_6)$, and thus $\ell_{e'} = 1$ in χ'. Furthermore, since no edge of $C^*(v_3v_6)$ participates in a monochromatic AP_6 in χ', it follows that both clauses $\beta = (\ell_{e'} \vee \ell_p \vee \ell_q)$ and $\overline{\beta} = (\overline{\ell_{e'}} \vee \overline{\ell_p} \vee \overline{\ell_q})$ are satisfied in τ', i.e. $\beta = 1$ and $\overline{\beta} = 1$ in τ', since both β and $\overline{\beta}$ include one of the literals $\{\ell_{e'}, \overline{\ell_{e'}}\}$. We will now prove that $\{\ell_p, \ell_q\} \cap \{\ell_e, \ell_{e''}\} \neq \emptyset$. Assume otherwise that $\{\ell_p, \ell_q\} \cap \{\ell_e, \ell_{e''}\} = \emptyset$. We distinguish the following three cases.

Case 1. $\ell_p \neq \ell_{e'}$ and $\ell_q \neq \ell_{e'}$. Then, since no two literals among $\{\ell_{e'}, \ell_p, \ell_q\}$ are one the negation of the other, it follows that $\ell_p, \ell_q \notin \{\ell_{e'}, \overline{\ell_{e'}}\}$. Therefore the values of ℓ_p and ℓ_q remain the same in both assignments τ and τ'. Since τ has been assumed to be an arbitrary assignment such that $\ell_e = \ell_{e'} = \ell_{e''} = 0$ in τ, we can choose the assignment τ to be such that $\ell_p = \ell_q = 1$ in τ. Since the value of $\ell_{e'}$ changes to 1 in τ', while the values of ℓ_p and ℓ_q are the same in both τ and τ', it follows that $\ell_{e'} = \ell_p = \ell_q = 1$ in τ', and thus $\overline{\beta} = 0$ in τ', which is a contradiction.

Case 2. Exactly one of $\{\ell_p, \ell_q\}$ is equal to $\ell_{e'}$. Let without loss of generality $\ell_p = \ell_{e'}$ and $\ell_q \neq \ell_{e'}$, i.e. $\ell_q \notin \{\ell_{e'}, \overline{\ell_{e'}}\}$. Therefore the value of ℓ_q remains the same in both assignments τ and τ'. Since τ has been assumed to be an arbitrary assignment such that $\ell_e = \ell_{e'} = \ell_{e''} = 0$ in τ, we can choose the assignment τ to be such that $\ell_q = 1$ in τ. Since the value of $\ell_p = \ell_{e'}$ changes to 1 in τ', while the value of ℓ_q is the same in both τ and τ', it follows that $\ell_{e'} = \ell_p = \ell_q = 1$ in τ', and thus $\overline{\beta} = 0$ in τ', which is a contradiction.

Case 3. $\ell_p = \ell_q = \ell_{e'}$. Then $\beta = (\ell_{e'} \vee \ell_p \vee \ell_q) = (\ell_{e'})$ and $\overline{\beta} = (\overline{\ell_{e'}} \vee \overline{\ell_p} \vee \overline{\ell_q}) = (\overline{\ell_{e'}})$, and thus it is not possible that both $\beta = 1$ and $\overline{\beta} = 1$ in τ', which is again a contradiction.

Therefore $\{\ell_p, \ell_q\} \cap \{\ell_e, \ell_{e''}\} \neq \emptyset$. Thus, since the clauses α and β share also the

literal $\ell_{e'}$, it follows that α and β share at least two literals. Therefore $\alpha = \beta$ by Lemma 6.2. This is a contradiction, since we assumed that $\beta \notin \{\alpha, \overline{\alpha}\}$. Therefore $\beta \in \{\alpha, \overline{\alpha}\}$, and thus $\{\alpha, \overline{\alpha}\} = \{\beta, \overline{\beta}\}$. This completes the proof of the lemma.
□

DEFINITION 6.4. *The clauses of ϕ_2 are partitioned into the* sub-formulas ϕ_2', ϕ_2'', *such that ϕ_2' contains all tautologies of ϕ_2 and all clauses of ϕ_2 in which at least one literal corresponds to an uncommitted edge, while ϕ_2'' contains all the remaining clauses of ϕ_2.*

LEMMA 6.5. *Let $\{e_1, e_2, e_3\}$ be the three edges of an AC_6 in H, which has clauses in ϕ_1. Let e be an edge of H such that $(\ell_e \vee \ell_{e_1})$ is a clause in ϕ_2''. Then ϕ_2'' contains also at least one of the clauses $\{(\ell_e \vee \overline{\ell_{e_2}}), (\ell_e \vee \overline{\ell_{e_3}})\}$.*

Proof. Recall that H is the associated split graph of $\widetilde{G}$, where $\widetilde{G}$ is the bipartite complement $\widehat{C}(P)$ of the domination bipartite graph $C(P)$ of the partial order P, cf. Definitions 3.3 and 4.2. For the purposes of the proof denote $C(P) = (U, V, E)$, where $U = \{u_1, u_2, \ldots, u_n\}$, $V = \{v_1, v_2, \ldots, v_n\}$; then $u_i v_j \in E$ if and only if $u_i <_P u_j$ (cf. Definition 4.2). Furthermore denote $\widetilde{G} = (U, V, \widetilde{E})$ for the bipartite complement $\widetilde{G} = \widehat{C}(P)$ of $C(P)$. Then $H = (U \cup V, E_H)$, where $E_H = \widetilde{E} \cup (V \times V)$ (cf. Definition 3.3). Moreover let $E_0 = \{u_i v_i \mid 1 \leq i \leq n\}$ and observe that $E_0 \subseteq \widetilde{E} \subseteq E_H$. Since edges of E correspond to non-edges of $\widetilde{E}$, it follows by the definition of E that $u_i v_j \notin \widetilde{E}$ if and only if $u_i <_P u_j$. That is, the non-edges of $\widetilde{E}$ between vertices of U and vertices of V follow the transitivity of the partial order P.

Since H is a split graph, Lemma 3.15 implies that the AC_6 of H is an AP_6, i.e. an alternating path of length 6 (cf. Figure 1(b)). Furthermore, since V induces a clique and U induces an independent set in H, Lemma 3.14 implies that the vertices of the AP_6 in H belong alternately to U and to V. Thus let $u_i, v_j, u_p, v_q, u_r, v_s$ be the vertices of the AP_6 (where $u_i v_j \notin E_H$ according to our notation, cf. Definition 3.1). Without loss of generality let $e_1 = u_p v_j$, $e_2 = u_r v_q$, and $e_3 = u_i v_s$. Since the AP_6 has clauses in ϕ_1 by assumption, note by the construction of ϕ_1 (cf. Section 5) that no two literals among $\{\ell_{e_1}, \ell_{e_2}, \ell_{e_3}\}$ are one the negation of the other. Therefore no pair among the edges $\{e_1, e_2, e_3\}$ is in conflict, and thus Lemma 3.7 implies that the edges $u_p v_s, u_i v_q, u_r v_j$ exist in H and $e_2 = u_r v_q || u_p v_s$, $e_1 = u_p v_j || u_i v_q$, and $e_3 = u_i v_s || u_r v_j$. Therefore $\ell_{u_i v_q} = \overline{\ell_{e_1}}$, $\ell_{u_p v_s} = \overline{\ell_{e_2}}$, and $\ell_{u_r v_j} = \overline{\ell_{e_3}}$.

Since $e_1 = u_p v_j$ and $(\ell_e \vee \ell_{e_1})$ is a clause of ϕ_2'' (and thus also of ϕ_2), it follows by the construction of ϕ_2 (cf. Section 5) that either $e = u_a v_p$ or $e = u_j v_a$ for some index $a \in \{1, 2, \ldots, n\}$.

Case 1. $e = u_a v_p$. Denote $E_H' = E_H \setminus E_0$. Then it follows by the construction of ϕ_2 that $u_a v_j \notin E_H'$, and thus either $u_a v_j \notin E_H$ or $u_a v_j \in E_0$. Furthermore, since $(\ell_e \vee \ell_{e_1})$ is a clause of ϕ_2'' by assumption, it follows by Definition 6.4 that e is a committed edge in H. That is, there exists an edge $e' = u_b v_c$ such that $e' || e$, and thus $\ell_{e'} = \overline{\ell_e}$. Since $e' || e$, it follows that $u_a v_c, u_b v_p \notin E_H$. Furthermore, since $u_b v_p, u_p v_q \notin E_H$, it follows that $u_b <_P u_p$ and $u_p <_P u_q$. Therefore $u_b <_P u_q$, since P is a partial order, and thus also $u_b v_q \notin E_H$.

Note that either $a = j$ or $a \neq j$ (cf. Figures 3(a) and 3(b), respectively. We distinguish now these two cases, which are illustrated in Figures 4(a) and 4(b), respectively. In these figures, the edges e_1, e_2, e_3 of the AP_6, as well as the edges e and e', are drawn by thick lines and all other edges are drawn by thin lines, while non-edges are illustrated with dashed lines.

Case 1.1. $a = j$ (cf. Figure 4(a)). Suppose that $u_i v_c \in E_H$. Then $u_i v_c || u_a v_j =$

u_jv_j, since $u_iv_j, u_av_c \notin E_H$. Thus the edge $u_jv_j \in E_0$ is committed, which is a contradiction by Lemma 5.2. Therefore $u_iv_c \notin E_H$. Suppose now that $u_pv_c \notin E_H$. Then $u_bv_c||u_pv_p$, since $u_bv_p, u_pv_c \notin E_H$. Thus the edge $u_pv_p \in E_0$ is committed, which is a contradiction by Lemma 5.2. Therefore $u_pv_c \in E_H$. Furthermore $u_pv_c||u_iv_q$, since $u_pv_q, u_iv_c \notin E_H$, and thus $\ell_{u_pv_c} = \overline{\ell_{u_iv_q}}$. Therefore, since $\ell_{u_iv_q} = \overline{\ell_{e_1}}$, it follows that $\ell_{u_pv_c} = \ell_{e_1}$.

Suppose that $u_av_q \notin E_H$, and thus $u_a <_P u_q$. Then, since $u_iv_j \notin E_H$, it follows that $u_i <_P u_j$. Therefore, since $a = j$ and P is a partial order, it follows that $u_i <_P u_q$, and thus $u_iv_q \notin E_H$, which is a contradiction. Therefore $u_av_q \in E_H$. Furthermore $u_av_q||u_pv_c$, since $u_av_c, u_pv_q \notin E_H$, and thus $\ell_{u_av_q} = \overline{\ell_{u_pv_c}}$. Therefore, since $\ell_{u_pv_c} = \ell_{e_1}$, it follows that $\ell_{u_av_q} = \overline{\ell_{e_1}}$.

Since $u_bv_q, u_av_c \notin E_H$, it follows that $e' = u_bv_c||u_av_q$, and thus $\ell_{e'} = \overline{\ell_{u_av_q}}$. Therefore, since $\ell_{u_av_q} = \overline{\ell_{e_1}}$, it follows that $\ell_{e'} = \ell_{e_1}$. Finally, since $e'||e$, it follows that $\ell_e = \overline{\ell_{e'}}$, and thus $\ell_e = \overline{\ell_{e_1}}$. Therefore the clause $(\ell_e \vee \ell_{e_1})$ of ϕ''_2 is a tautology, which is a contradiction by Definition 6.4.

Case 1.2. $a \neq j$ (cf. Figure 4(b)). Then $u_av_j \notin E_0$. Thus, since $u_av_j \notin E'_H$, it follows that $u_av_j \notin E_H$. Suppose that $u_av_s \in E_H$ (cf. Figure 4(b)). Then $u_av_s||u_rv_j$, since $u_av_j, u_rv_s \notin E_H$, and thus $\ell_{u_av_s} = \overline{\ell_{u_rv_j}}$. Therefore, since $\ell_{u_rv_j} = \overline{\ell_{e_3}}$, it follows that $\ell_{u_av_s} = \ell_{e_3}$. Suppose that $u_av_q \notin E_H$. Then $u_rv_q||u_av_s$, since $u_rv_s, u_av_q \notin E_H$. Therefore $\ell_{u_av_s} = \overline{\ell_{e_2}}$, since $\ell_{u_rv_q} = \ell_{e_2}$. Thus, since $\ell_{u_av_s} = \ell_{e_3}$, it follows that $\ell_{e_3} = \overline{\ell_{e_2}}$. This is a contradiction, since no two literals among $\{\ell_{e_1}, \ell_{e_2}, \ell_{e_3}\}$ are one the negation of the other. Therefore $u_av_q \in E_H$. Moreover, since $u_av_j, u_pv_q \notin E_H$, it follows that $u_av_q||u_pv_j = e_1$, and thus $\ell_{u_av_q} = \overline{\ell_{e_1}}$. Furthermore $u_av_q||u_bv_c = e'$, since $u_bv_q, u_av_c \notin E_H$. Therefore $\ell_{u_av_q} = \overline{\ell_{e'}}$. Thus $\ell_{u_av_q} = \ell_e$, since $\ell_{e'} = \overline{\ell_e}$. Therefore, since $\ell_{u_av_q} = \overline{\ell_{e_1}}$ and $\ell_{u_av_q} = \ell_e$, it follows that $\ell_e = \overline{\ell_{e_1}}$. Therefore the clause $(\ell_e \vee \ell_{e_1})$ of ϕ''_2 is a tautology, which is a contradiction by Definition 6.4.

Therefore $u_av_s \notin E_H$. Then also $u_av_s \notin E'_H$, and thus ϕ_2 has the clause $(\ell_{u_av_p} \vee \ell_{u_pv_s}) = (\ell_e \vee \overline{\ell_{e_2}})$, since $e = u_av_p$ and $\ell_{u_pv_s} = \overline{\ell_{e_2}}$. Furthermore, since both e and u_pv_s are committed in H (as $e'||e$ and $u_rv_q||u_pv_s$), the clause $(\ell_e \vee \overline{\ell_{e_2}})$ belongs to ϕ''_2 by Definition 6.4.

Case 2. $e = u_jv_a$. This case is exactly symmetric to Case 1. To see this, imagine exchanging the roles of U and V, i.e. U induces now a clique (instead of an independent set) and V induces an independent set (instead of a clique) in H. Imagine also flipping the lines L_1 and L_2 in Figure 4 (i.e. L_2 comes now above L_1), such that the vertices of U and V still lie on the lines L_1 and L_2, respectively. Similarly to Cases 1.1 and 1.2, we distinguish the cases $a = p$ (Case 2.1) and $a \neq p$ (Case 2.2), respectively. Then, Case 2.1 leads to a contradiction (similarly to Case 1.1), and Case 2.2 implies that the clause $(\ell_e \vee \overline{\ell_{e_3}})$ belongs to ϕ''_2 (instead of the clause $(\ell_e \vee \overline{\ell_{e_2}})$ in Case 1.2).

Summarizing, if $e = u_av_p$ then ϕ''_2 includes the clause $(\ell_e \vee \overline{\ell_{e_2}})$, while if $e = u_jv_a$ then ϕ''_2 includes the clause $(\ell_e \vee \overline{\ell_{e_3}})$. This completes the proof of the lemma. $\square$

Lemma 6.6. *Let $\{e_1, e_2, e_3\}$ be the three edges of an AC_6 in H, which has clauses in ϕ_1. Let e be an edge of H such that $(\ell_e \vee \overline{\ell_{e_1}})$ is a clause in ϕ''_2. Then ϕ''_2 contains also at least one of the clauses $\{(\ell_e \vee \ell_{e_2}), (\ell_e \vee \ell_{e_3})\}$.*

Proof. Since H is a split graph, Lemma 3.15 implies that the AC_6 of H is an AP_6, i.e. an alternating path of length 6 (cf. Figure 1(b)). Using the notation of Lemma 6.5, denote by V and U the clique and the independent set of H, respectively. Then the vertices of the AP_6 in H belong alternately to U and to V by Lemma 3.14.

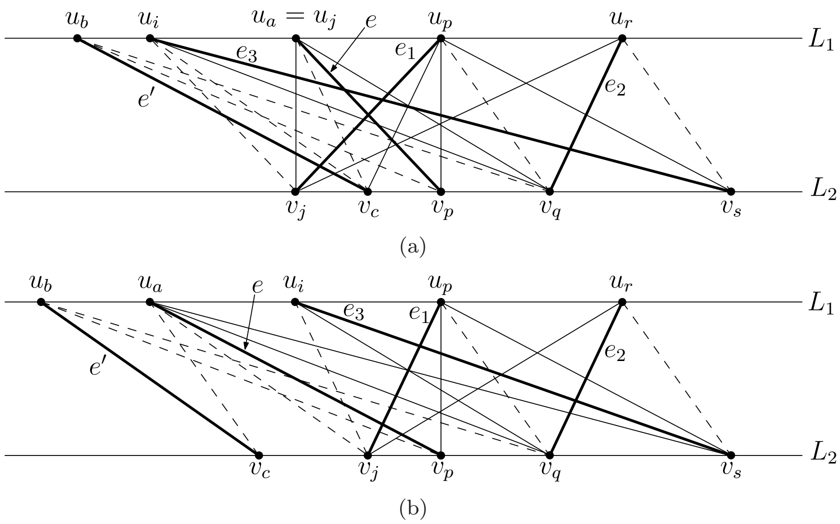

FIGURE 4. *(a) The Case 1.1 and (b) the Case 1.2 in the proof of Lemma 6.5.*

That is, $u_i, v_j, u_p, v_q, u_r, v_s$ are the vertices of the AP_6 in this order, for some vertices $u_i, u_p, u_r \in U$ and $v_j, v_q, v_s \in V$ (where $u_iv_j, u_pv_q, u_rv_s \notin E_H$ according to our notation, cf. Definition 3.1). Without loss of generality let $e_1 = u_pv_j$, $e_2 = u_rv_q$, and $e_3 = u_iv_s$. Then, similarly to the preamble of the proof of Lemma 6.5, it follows that the edges $e_1' = u_iv_q$, $e_2' = u_pv_s$, and $e_3' = u_rv_j$ exist in H and $e_1 = u_pv_j || u_iv_q = e_1'$, $e_2 = u_rv_q || u_pv_s = e_2'$, and $e_3 = u_iv_s || u_rv_j = e_3'$. Therefore $\ell_{e_1'} = \overline{\ell_{e_1}}$, $\ell_{e_2'} = \overline{\ell_{e_2}}$, and $\ell_{e_3'} = \overline{\ell_{e_3}}$.

Since $u_iv_j, u_pv_q, u_rv_s \notin E_H$, it follows that the vertices $u_i, v_q, u_p, v_s, u_r, v_j$ (in this order) build an AC_6 in H, where $\{e_1', e_2', e_3'\}$ are its three edges. Therefore, by applying Lemma 3.15 on this new AC_6, it follows that if $(\ell_e \vee \ell_{e_1'})$ is a clause in ϕ_2'', then ϕ_2'' contains also at least one of the clauses $\{(\ell_e \vee \overline{\ell_{e_2'}}), (\ell_e \vee \overline{\ell_{e_3'}})\}$. This completes the proof of the lemma, since $\ell_{e_1'} = \overline{\ell_{e_1}}$, $\ell_{e_2'} = \overline{\ell_{e_2}}$, and $\ell_{e_3'} = \overline{\ell_{e_3}}$. ☐

The next corollary, which follows easily by Definition 2.2 and by Lemmas 6.3-6.6, allows us to use the linear time algorithm for gradually mixed formulas (cf. Theorem 2.3) in order to solve the SAT problem on $\phi_1 \wedge \phi_2''$.

COROLLARY 6.7. *$\phi_1 \wedge \phi_2''$ is a gradually mixed formula.*

Proof. First note that, by construction, every clause of ϕ_1 has 3 literals and every clause of ϕ_2 has 2 literals. Furthermore, the first condition of Definition 2.2 is satisfied due to Lemma 6.3. Regarding the second condition of Definition 2.2, consider an arbitrary AC_6 in H that has clauses in ϕ_1. Denote by $\{e_1, e_2, e_3\}$ the three edges of this AC_6. Then this AC_6 contributes to the formula ϕ_1 by the two (dual) clauses $\alpha = (\ell_{e_1} \vee \ell_{e_2} \vee \ell_{e_3})$ and $\overline{\alpha} = (\overline{\ell_{e_1}} \vee \overline{\ell_{e_2}} \vee \overline{\ell_{e_3}})$, cf. the construction of ϕ_1 in Section 5. If $(\ell_e \vee \ell_{e_1})$ is a clause of ϕ_2'', then Lemma 6.5 implies that ϕ_2'' includes also at least one of the clauses $\{(\ell_e \vee \overline{\ell_{e_2}}), (\ell_e \vee \overline{\ell_{e_3}})\}$. Similarly, if $(\ell_e \vee \overline{\ell_{e_1}})$ is a clause of ϕ_2'', Lemma 6.6 implies that ϕ_2'' includes also at least one of the clauses $\{(\ell_e \vee \ell_{e_2}), (\ell_e \vee \ell_{e_3})\}$. Therefore the second condition of Definition 2.2 is also satisfied for the formula $\phi_1 \wedge \phi_2''$, i.e. $\phi_1 \wedge \phi_2''$ is a gradually mixed formula. ☐

6.2. The recognition algorithm. In this section we use Corollary 6.7 to design an algorithm that decides satisfiability on $\phi_1 \wedge \phi_2$ in time linear to its size (cf. Theorem 6.8). This will enable us to combine the results of Sections 4 and 5 to recognize efficiently whether a given graph is a PI graph, or equivalently, due to Theorem 4.1, whether a given partial order P is the intersection of a linear order P_1 and an interval order P_2.

THEOREM 6.8. *$\phi_1 \wedge \phi_2$ is satisfiable if and only if $\phi_1 \wedge \phi_2''$ is satisfiable. Given a satisfying truth assignment of $\phi_1 \wedge \phi_2''$ we can compute a satisfying truth assignment of $\phi_1 \wedge \phi_2$ in linear time.*

Proof. If $\phi_1 \wedge \phi_2$ is satisfiable then $\phi_1 \wedge \phi_2''$ is also satisfiable as a sub-formula of $\phi_1 \wedge \phi_2$. Conversely, suppose that $\phi_1 \wedge \phi_2''$ is satisfiable and let τ be a satisfying assignment. Consider an arbitrary clause $\gamma = (\ell_{e_1} \vee \ell_{e_2})$ of the sub-formula ϕ_2' of ϕ_2, cf. Definition 6.4. If γ is a tautology then γ is satisfied by any truth assignment of ϕ, and thus also by τ. Assume now that γ is not a tautology. Then at least one of its literals $\{\ell_{e_1}, \ell_{e_2}\}$ corresponds to an uncommitted edge by Definition 6.4. Recall now by the construction of ϕ_1 (cf. Section 5) that in every clause of ϕ_1, no literal is the negation of another literal. Thus, for every clause of ϕ_1, no pair among the three edges in the corresponding AC_6 is in conflict. Therefore Lemma 3.7 implies that all three edges of such an AC_6 are committed. Thus, for every literal ℓ_e of ϕ_2', which corresponds to an uncommitted edge e, neither ℓ_e nor $\overline{\ell_e}$ appears in ϕ_1. Furthermore recall that ϕ_2'' does not include any literal ℓ_e of any uncommitted edge e of H by Definition 6.4.

Summarizing, for every literal ℓ_e of ϕ_2', which corresponds to an uncommitted edge e, neither ℓ_e nor $\overline{\ell_e}$ appears in $\phi_1 \wedge \phi_2''$. That is, the truth assignment τ of $\phi_1 \wedge \phi_2$ does not assign any value to the literal ℓ_e. Furthermore, since e is uncommitted, no edge of H is assigned the literal $\overline{\ell_e}$. Therefore we can extend (in linear time) the truth assignment τ to a truth assignment τ' that satisfies both $\phi_1 \wedge \phi_2''$ and ϕ_2', by setting $\ell_e = 1$ for all uncommitted edges e of H. That is, τ' satisfies the formula $\phi_1 \wedge \phi_2$. Therefore $\phi_1 \wedge \phi_2$ is satisfiable if and only if $\phi_1 \wedge \phi_2''$ is satisfiable. This completes the proof of the theorem. □

Now we are ready to present our recognition algorithm for PI graphs (Algorithm 5). Its correctness and timing analysis is established in Theorem 6.9.

THEOREM 6.9. *Let $G = (V, E)$ be a graph and $\overline{G} = (V, \overline{E})$ be its complement, where $|V| = n$ and $|\overline{E}| = m$. Then Algorithm 5 constructs in $O(n^2 m)$ time a PI representation of G, or it announces that G is not a PI graph.*

Proof. If the given graph G is a trapezoid graph, then Algorithm 5 computes in line 2 a partial order P of its complement $\overline{G}$. Otherwise, if G is not a trapezoid graph, then clearly it is also not a PI graph, and thus the algorithm correctly announces in line 3 that G is not a PI graph.

Let $C(P)$ be the domination bipartite graph of the partial order P (cf. Definition 4.2), and let $\widetilde{G} = \widehat{C}(P)$ be the bipartite complement of $C(P)$, which are computed in lines 4 and 5 of Algorithm 5, respectively. Furthermore let H be the associated split graph of $\widetilde{G}$ (cf. Definition 3.3) and H^* be the conflict graph of H (cf. Definition 3.6), which are computed in lines 6 and 7 of Algorithm 5, respectively. If H^* is not bipartite, i.e. if $\chi(H^*) > 2$, then $\widetilde{G}$ is not linear-interval coverable by Lemma 5.1, and thus G is not a PI graph by Corollary 4.9. Therefore Algorithm 5 correctly announces in line 18 that G is not a PI graph if H^* is not bipartite.

Suppose now that H^* is bipartite, i.e. $\chi(H^*) \leq 2$. Let χ_0 be a 2-coloring of the vertices of H^*, which is computed in line 9 of Algorithm 5. Furthermore let ϕ_1 and ϕ_2 be the Boolean formulas that can be computed by Algorithms 2 and 3, respectively (cf. line 10 of Algorithm 5). If the formula $\phi_1 \wedge \phi_2$ is not satisfiable, then $\widetilde{G}$ is not linear-interval coverable by Theorem 5.4, and thus G is not a PI graph by

Algorithm 5 Recognition of PI graphs

Input: A graph $G = (V, E)$

Output: A PI representation R of G, or the announcement that G is not a PI graph

```
1: if G is a trapezoid graph then
2:     Compute a partial order P of the complement \overline{G}
3: else return "G is not a PI graph"
4: Compute the domination bipartite graph C(P) from P
5: \widetilde{G} ← \widehat{C}(P)
6: Compute the associated split graph H of \widetilde{G}
7: Compute the conflict graph H* of H
8: if H* is bipartite then
9:     Compute a 2-coloring χ_0 of the vertices of H*
10:    Compute the formulas φ_1 and φ_2
11:    if φ_1 ∧ φ_2 is satisfiable then
12:        Compute a satisfying truth assignment τ of φ_1 ∧ φ_2 by Theorem 6.8
13:        Compute from τ a linear-order cover of \widetilde{G} by Algorithm 4
14:        Compute a PI representation R of G by Algorithm 1
15:    else
16:        return "G is not a PI graph"
17: else
18:    return "G is not a PI graph"
19: return R
```

Corollary 4.9. Therefore Algorithm 5 correctly announces in line 16 that G is not a PI graph if $\phi_1 \wedge \phi_2$ is not satisfiable.

Suppose now that $\phi_1 \wedge \phi_2$ is satisfiable, and let τ be a satisfying truth assignment of $\phi_1 \wedge \phi_2$, as it is computed in line 12 of Algorithm 5. Then $\widetilde{G}$ is linear-interval coverable by Theorem 5.4, and thus G is a PI graph by Corollary 4.9. Furthermore, given τ, we can compute a linear-interval cover of $\widetilde{G}$ using Algorithm 4 (cf. line 13 of Algorithm 5). Finally, given this linear-interval cover of $\widetilde{G}$, we can compute a PI representation R of G using Algorithm 1 (cf. line 14 of Algorithm 5). Thus, if $\phi_1 \wedge \phi_2$ is satisfiable, Algorithm 5 correctly returns R in line 19.

Time complexity. First note that the complement $\overline{G}$ of G can be computed in $O(n^2)$ time, since both G and $\overline{G}$ have n vertices. Furthermore, using the algorithm of [15] we can decide in $O(n^2)$ time whether G is a trapezoid graph, and within the same time bound we can compute a trapezoid representation of G, if it exists. Suppose in the following that G is a trapezoid graph. Then we can then compute in $O(n^2)$ time a partial order P of the complement $\overline{G}$ of G as follows: $u <_P v$ if and only if the trapezoid for vertex u lies entirely to the left of the trapezoid for vertex v in this trapezoid representation of G. Therefore, lines 1-3 of Algorithm 5 can be executed in $O(n^2)$ time in total. Note that we choose to compute the partial order P using the trapezoid graph recognition algorithm of [15], in order to achieve the $O(n^2)$ time bound. Alternatively we could solve the transitive orientation problem on $\overline{G}$ using the standard forcing algorithm with $O(nm)$ running time (note that m is the number of edges of $\overline{G}$).

Denote $\widetilde{G} = (U, V, \widetilde{E})$, where $U = \{u_1, u_2, \ldots, u_n\}$ and $V = \{v_1, v_2, \ldots, v_n\}$.

Furthermore denote $E_0 = \{u_i v_i \mid 1 \leq i \leq n\}$. Then $H = (U, V, E_H)$, where $E_H = \widetilde{E} \cup (V \times V)$ by Definition 3.3. Since $C(P)$ and H have $2n$ vertices each, each of the lines 4-6 of Algorithm 5 can be computed by a straightforward implementation in $O(n^2)$ time. Note that the partial order P has m pairs of comparable elements, since the complement $\overline{G}$ of G has m edges. Therefore the domination bipartite graph $C(P)$ of P has m edges (cf. Definition 4.2), and thus its bipartite complement $\widetilde{G} = \widehat{C}(P)$ has $|\widetilde{E}| = n^2 - m$ edges.

Consider a pair $\{e, e'\}$ of edges of H that are in conflict, i.e. $e||e'$ in H. Then $e, e' \notin V \times V$ by Observation 3, since H is a split graph and V induces a clique in H. Therefore both e and e' are edges of $\widetilde{G}$, i.e. $e, e' \in \widetilde{E}$, and thus $e = u_i v_j$ and $e' = u_p v_q$ for some indices $i, j, p, q \in \{1, 2, \ldots, n\}$. Furthermore, since e and e' are in conflict, it follows that $u_i v_q, u_p v_j \notin \widetilde{E}$. That is, every pair of conflicting edges in H corresponds to exactly one pair $\{u_i v_q, u_p v_j\}$ of non-edges of $\widetilde{G} = \widehat{C}(P)$. Equivalently, every edge in the conflict graph H^* of H corresponds to exactly one pair of edges of $C(P)$. Since $C(P)$ has m edges, it follows that the conflict graph H^* has at most $O(m^2)$ edges. Furthermore note that the conflict graph H^* has $\binom{n}{2} + |\widetilde{E}| = O(n^2)$ vertices, since H has $\binom{n}{2} + |\widetilde{E}|$ edges. Therefore the conflict graph H^* can be computed in $O(n^2 + m^2)$ time (cf. line 7 of Algorithm 5).

Note now that in time linear to the size of H^*, we can check whether H^* is bipartite, and we can compute a 2-coloring χ_0 of the vertices of H^*, if one exists. Therefore lines 8-9 of Algorithm 5 can be executed in $O(n^2 + m^2)$ time. Furthermore, in time linear to the size of H^*, i.e. in $O(n^2 + m^2)$ time, we can compute the connected components $C_1, C_2, \ldots, C_k$ of H^*. Then, having already computed the 2-coloring χ_0 and the connected components $C_1, C_2, \ldots, C_k$ of H^*, we can assign to every edge e of H the literal $\ell_e \in \{x_i, \overline{x_i} \mid 1 \leq i \leq k\}$ (cf. Section 5). This can be done in $O(n^2)$ time, since H has $\binom{n}{2} + |\widetilde{E}| = O(n^2)$ edges.

Now we bound the size of the formulas ϕ_1 and ϕ_2 that are computed by Algorithms 2 and 3, respectively. Regarding the size of ϕ_2, note that, by the construction of ϕ_2, if $(\ell_e \vee \ell_{e'})$ is a clause of ϕ_2, then $e = u_i v_t$, $e' = u_t v_j$, and $u_i v_j \notin E_H \setminus E_0$, for some indices $i, j, t \in \{1, 2, \ldots, n\}$. That is, for every index $t \in \{1, 2, \ldots, n\}$ and for every pair (i, j) of indices in the set $\{(i, j) \mid i = j \text{ or } u_i v_j \notin E_H\}$, the formula ϕ_2 has at most one clause. Note that every pair (i, j) of the set $\{(i, j) \mid u_i v_j \notin E_H\}$ corresponds to exactly one edge $u_i v_j$ of the bipartite graph $C(P)$. Thus, since $C(P)$ has m edges, it follows that $|\{(i, j) \mid i = j \text{ or } u_i v_j \notin E_H\}| \leq n + m$. Therefore ϕ_2 has at most $n(n+m)$ clauses, and thus ϕ_2 can be computed in $O(n(n+m))$ time by Algorithm 3.

Regarding the size of ϕ_1, recall first that every connected component C_i of the conflict graph H^* has been assigned exactly one Boolean variable x_i, where $i \in \{1, 2, \ldots, k\}$. Furthermore recall that every edge e of H has been assigned a literal $\ell_e \in \{x_i, \overline{x_i} \mid 1 \leq i \leq k\}$. Therefore, since every clause of ϕ_1 appears only once in ϕ_1 (cf. lines 4-5 of Algorithm 2), it follows by the construction of ϕ_1 and by Lemma 6.3 that ϕ_1 has at most $2\frac{k}{3}$ clauses. Furthermore note that $k = O(n^2)$, since H^* has $O(n^2)$ vertices. Thus ϕ_1 has at most $O(n^2)$ clauses.

Claim 2. *The following two statements are equivalent:*

(a) the formula ϕ_1 contains the clauses $\alpha = (\ell_e \vee \ell_{e'} \vee \ell_{e''})$ and $\alpha' = (\overline{\ell_e} \vee \overline{\ell_{e'}} \vee \overline{\ell_{e''}})$,

(b) there exist four distinct vertices a, b, c, d in H, such that:

- *$ab \notin E_H$ and $bc, cd, da \in E_H$,*
- *either $a, c \in U$ and $b, d \in V$, or $a, c \in V$ and $b, d \in U$,*

- *the edges bc, cd, da are committed in H,*
- $\ell_{bc} = \ell_e$, $\ell_{cd} = \overline{\ell_{e'}}$, $\ell_{da} = \ell_{e''}$, *and*
- $\ell_e \neq \overline{\ell_{e'}}$, $\ell_{e'} \neq \overline{\ell_{e''}}$, $\ell_e \neq \overline{\ell_{e''}}$.

Proof of Claim 2. ((a) $\Rightarrow$ (b)) Consider first a pair of clauses $\alpha = (\ell_e \vee \ell_{e'} \vee \ell_{e''})$ and $\alpha' = (\overline{\ell_e} \vee \overline{\ell_{e'}} \vee \overline{\ell_{e''}})$ in ϕ_1. These clauses correspond to an AC_6 on the edges $\{e, e', e''\}$ of H by the construction of ϕ_1. Furthermore, since H is a split graph, Lemma 3.15 implies that this AC_6 of H is an AP_6, i.e. an alternating path of length 6 (cf. Figure 1(b)). Let $w_1, w_2, w_3, w_4, w_5, w_6$ be the vertices of this AP_6, such that $e = w_2w_3$, $e' = w_4w_5$, and $e'' = w_6w_1$ (note that there always exists an enumeration of the vertices of the AP_6 such that the edges e, e', e'' are met in this order on the AP_6). Then, since V induces a clique in H and U induces an independent set in H, Lemma 3.14 implies that either $w_1, w_3, w_5 \in U$ and $w_2, w_4, w_6 \in V$, or $w_1, w_3, w_5 \in V$ and $w_2, w_4, w_6 \in U$. Since $\ell_e \neq \overline{\ell_{e'}}$, $\ell_{e'} \neq \overline{\ell_{e''}}$, and $\ell_e \neq \overline{\ell_{e''}}$ (cf. line 3 of Algorithm 2), it follows that no pair among the edges $\{e, e', e''\}$ is in conflict in H. Therefore Lemma 3.7 implies that the edges w_3w_6, w_4w_1, w_5w_2 exist in H and $e' || w_3w_6$, $e || w_4w_1$, and $e'' || w_5w_2$. Thus all six edges $\{e, e', e'', w_3w_6, w_4w_1, w_5w_2\}$ are committed. Furthermore $\ell_{w_4w_1} = \overline{\ell_e}$, $\ell_{w_3w_6} = \overline{\ell_{e'}}$, and $\ell_{w_5w_2} = \overline{\ell_{e''}}$. Thus the vertices $a = w_1$, $b = w_2$, $c = w_3$, and $d = w_6$ of H satisfy the conditions of the part (b) of the claim.

((b) $\Rightarrow$ (a)) Conversely, consider four vertices a, b, c, d in H, as specified in the part (b) of the claim. Then, since the edge cd is committed, there exists an edge $pq \in E_H$ such that $pc, qd \notin E_H$, and thus $cd || pq$. Then $\ell_{pq} = \overline{\ell_{cd}}$. Therefore, since $\ell_{cd} = \overline{\ell_{e'}}$, it follows that $\ell_{pq} = \ell_{e'}$. Thus there exists an AC_6 in H on the vertices a, b, c, p, q, d, where $\ell_{bc} = \ell_e$, $\ell_{pq} = \ell_{e'}$, and $\ell_{da} = \ell_{e''}$. Furthermore, since $\ell_e \neq \overline{\ell_{e'}}$, $\ell_{e'} \neq \overline{\ell_{e''}}$, and $\ell_e \neq \overline{\ell_{e''}}$ by assumption, it follows by the construction of ϕ_1 (cf. Algorithm 2) that ϕ_1 contains the clauses $\alpha = (\ell_e \vee \ell_{e'} \vee \ell_{e''})$ and $\alpha' = (\overline{\ell_e} \vee \overline{\ell_{e'}} \vee \overline{\ell_{e''}})$. ■

Now, due to Claim 2, we can implement Algorithm 2 for the computation of ϕ_1 in time $O(n^2m + m^2)$ as follows. Recall first that $C(P)$ has m edges. We iterate for every edge u_iv_j of $C(P)$, i.e. for every *non-edge* $u_iv_j \notin E_H$ of H. For every such u_iv_j, we mark all vertices in the sets A and B, where $A = \{v \in V \mid u_iv \in E_H \text{ and } u_iv \text{ is committed in } H\}$ and $B = \{u \in U \mid uv_j \in E_H \text{ and } uv_j \text{ is committed in } H\}$. Then we scan through the adjacency lists of all vertices in A to discover a pair of vertices $v \in A$ and $u \in B$ such that uv is a committed edge of H, and $\ell_{v_ju} \neq \ell_{uv}$, $\ell_{uv} \neq \ell_{vu_i}$, and $\ell_{v_ju} \neq \overline{\ell_{vu_i}}$. Since H has $O(n^2)$ edges, this scan through the adjacency lists of the vertices of A can be done in $O(n^2)$ time. If we discover such an edge uv, then we add to ϕ_1 the clauses $\alpha = (\ell_{v_ju} \vee \overline{\ell_{uv}} \vee \ell_{vu_i})$ and $\alpha' = (\overline{\ell_{v_ju}} \vee \ell_{uv} \vee \overline{\ell_{vu_i}})$. Due to Claim 2, Algorithm 2 would add the same two clauses to ϕ_1.

Due to Lemma 6.3, no other clause of ϕ_1 has one of the literals $\{\ell_{v_ju}, \overline{\ell_{v_ju}}, \ell_{uv}, \overline{\ell_{uv}}, \ell_{vu_i}, \overline{\ell_{vu_i}}\}$. After we add the two clauses α and α' to ϕ_1, we visit all edges e of H which correspond to the same connected component in H^* with one of the edges $\{v_ju, uv, vu_j\}$. Note that exactly these edges e of H have a literal $\ell_e \in \{\ell_{v_ju}, \overline{\ell_{v_ju}}, \ell_{uv}, \overline{\ell_{uv}}, \ell_{vu_i}, \overline{\ell_{vu_i}}\}$. We then mark all these edges e such that we avoid visiting them again in any subsequent iteration during the construction of ϕ_1. Thus we ensure that each clause appears at most once ϕ_1 (cf. lines 4-5 of Algorithm 2). Note that we can perform all such markings of edges e (for all iterations during the construction of ϕ_1) in time linear to the size of H^*, i.e. in $O(n^2 + m^2)$ time. Summarizing, we need in total $O(n^2m + m^2)$ time to compute the formula ϕ_1. Thus, since the formula ϕ_2 can be computed in $O(n(n + m))$ time, it follows that line 10 of

Algorithm 5 can be executed in $O(n^2m + m^2)$ time.

Now, we can test whether the formula $\phi_1 \wedge \phi_2$ is satisfiable in time linear to its size by Theorem 6.8; moreover, within the same time bound we can compute a satisfying truth assignment τ of $\phi_1 \wedge \phi_2$, if one exists. Thus, since ϕ_1 has $O(n^2)$ clauses and ϕ_2 has $O(n(n+m))$ clauses, lines 11-12 of Algorithm 5 can be executed in $O(n(n+m))$ time. Furthermore, line 13 of Algorithm 5 can be executed in $O(n^2)$ time by Theorem 5.4, calling Algorithm 4 as a subroutine. Finally, line 14 of Algorithm 5 can be executed in $O(n^2)$ time by Theorem 4.10, calling Algorithm 1 as a subroutine. Summarizing, since $m = O(n^2)$, the total running time of Algorithm 5 is $O(n^2m)$. This completes the proof of the theorem. □

Due to characterization of PI graphs in Theorem 4.1 using partial orders, the next theorem follows now by Theorem 6.9.

THEOREM 6.10. *Let $P = (U, R)$ be a partial order, where $|U| = n$ and $|R| = m$. Then we can decide in $O(n^2m)$ time whether P is a linear-interval order, and in this case we can compute a linear order P_1 and an interval order P_2 such that $P = P_1 \cap P_2$.*

7. Concluding remarks. In this article we provided the first polynomial algorithm for the recognition of simple-triangle graphs, or equivalently for the recognition of linear-interval orders, solving thus a longstanding open problem. For a graph G with n vertices, where its complement $\overline{G}$ has m edges, our $O(n^2m)$-time algorithm either computes a simple-triangle representation of G, or it announces that such one does not exist. The main tool for our recognition algorithm was a new hybrid tractable subclass of 3SAT, called the class of *gradually mixed* formulas. In addition, we introduced the notion of a *linear-interval cover* of bipartite graphs, which naturally extends the well-known notion of the chain-cover of bipartite graphs. There are two main lines for further research. The first one is to identify more "islands of tractability" for hybrid classes of SAT (and more generally of CSP), while the ultimate goal is to find a complete characterization of the hybrid classes of CSP that are tractable. The second line for further research is to resolve the complexity of the recognition for the related classes with simple-triangle graphs, such as the classes of *unit* and *proper tolerance* graphs [11] (these are subclasses of parallelogram graphs, and thus also subclasses of trapezoid graphs), *proper bitolerance* graphs [2, 11] (they coincide with *unit bitolerance* graphs [2]), and *multitolerance* graphs [18] (they naturally generalize trapezoid graphs [18, 20]). On the contrary, the recognition problems for the related classes of *triangle* graphs [17], *tolerance* and *bounded tolerance* (i.e. *parallelogram*) graphs [19], and *max-tolerance* graphs [14] have been already proved to be NP-complete.

Acknowledgment. The author wishes to thank Jayme Luiz Szwarcfiter for initial discussions on the simple-triangle recognition problem.

REFERENCES

[1] S. Benzer. On the topology of the genetic fine structure. *Proc. of the National Academy of Sciences (PNAS)*, 45:1607–1620, 1959.

[2] K. P. Bogart and G. Isaak. Proper and unit bitolerance orders and graphs. *Discrete Mathematics*, 181(1-3):37–51, 1998.

[3] A. Brandstädt, V. B. Le, and J. P. Spinrad. *Graph classes: a survey.* Society for Industrial and Applied Mathematics (SIAM), 1999.

[4] A. V. Carrano. Establishing the order to human chromosome-specific DNA fragments. *Biotechnology and the Human Genome*, Plenum Press:37–50, 1988.

[5] M. R. Cerioli, F. de S. Oliveira, and J. L. Szwarcfiter. Linear-interval dimension and PI orders. *Electronic Notes in Discrete Mathematics*, 30:111–116, 2008.

[6] D. G. Corneil and P. A. Kamula. Extensions of permutation and interval graphs. In *Proceedings of the 18th Southeastern Conference on Combinatorics, Graph Theory and Computing*, pages 267–275, 1987.

[7] I. Dagan, M. C. Golumbic, and R. Y. Pinter. Trapezoid graphs and their coloring. *Discrete Applied Mathematics*, 21(1):35–46, 1988.

[8] S. Even, A. Itai, and A. Shamir. On the complexity of timetable and multicommodity flow problems. *SIAM Journal on Computing*, 5(4):691–703, 1976.

[9] P. W. Goldberg, M. C. Golumbic, H. Kaplan, and R. Shamir. Four strikes against physical mapping of DNA. *Journal of Computational Biology*, 2(1):139–152, 1995.

[10] M. C. Golumbic. *Algorithmic graph theory and perfect graphs (Annals of Discrete Mathematics, Vol. 57)*. North-Holland Publishing Co., 2 edition, 2004.

[11] M. C. Golumbic and A. N. Trenk. *Tolerance Graphs*. Cambridge Studies in Advanced Mathematics, 2004.

[12] P. L. Hammer, T. Ibaraki, and U. N. Peled. Threshold numbers and threshold completions. *Annals of Discrete Mathematics*, 11:125–145, 1981.

[13] T. Ibaraki and U. N. Peled. Sufficient conditions for graphs to have threshold number 2. *Annals of Discrete Mathematics*, 11:241–268, 1981.

[14] M. Kaufmann, J. Kratochvil, K. A. Lehmann, and A. R. Subramanian. Max-tolerance graphs as intersection graphs: cliques, cycles, and recognition. In *Proceedings of the 17th annual ACM-SIAM symposium on Discrete Algorithms (SODA)*, pages 832–841, 2006.

[15] T.-H. Ma and J. P. Spinrad. On the 2-chain subgraph cover and related problems. *Journal of Algorithms*, 17:251–268, 1994.

[16] N. Mahadev and U. N. Peled. *Threshold Graphs and Related Topics (Annals of Discrete Mathematics, Vol. 56)*. North-Holland Publishing Co., 1995.

[17] G. B. Mertzios. The recognition of triangle graphs. *Theoretical Computer Science*, 438:34–47, 2012.

[18] G. B. Mertzios. An intersection model for multitolerance graphs: Efficient algorithms and hierarchy. *Algorithmica*, 69(3):540–581, 2014.

[19] G. B. Mertzios, I. Sau, and S. Zaks. The recognition of tolerance and bounded tolerance graphs. *SIAM Journal on Computing*, 40(5):1234–1257, 2011.

[20] A. Parra. Triangulating multitolerance graphs. *Discrete Applied Mathematics*, 84(1-3):183–197, 1998.

[21] T. Raschle and K. Simon. Recognition of graphs with threshold dimension two. In *Proceedings of the 27th ACM symposium on Theory of computing (STOC)*, pages 650–661, 1995.

[22] J. P. Spinrad. *Efficient graph representations*, volume 19 of *Fields Institute Monographs*. American Mathematical Society, 2003.

[23] M. Yannakakis. The complexity of the partial order dimension problem. *SIAM Journal on Algebraic and Discrete Methods*, 3:351–358, 1982.