Extending Partial Representations of Circle Graphs*

Steven Chaplick
LEHRSTUHL FÜR INFORMATIK I, UNIVERSITÄT WÜRZBURG, GERMANY.
E-MAIL: `STEVEN.CHAPLICK@UNI-WUERZBURG.DE`.

Radoslav Fulek
INSTITUTE OF SCIENCE AND TECHNOLOGY AUSTRIA, KLOSTERNEUBURG, AUSTRIA.
E-MAIL: `RADOSLAV.FULEK@GMAIL.COM`.

Pavel Klavík
COMPUTER SCIENCE INSTITUTE, CHARLES UNIVERSITY IN PRAGUE,
CZECH REPUBLIC. E-MAIL: `KLAVIK@IUUK.MFF.CUNI.CZ`.

ABSTRACT

The partial representation extension problem *is a recently introduced generalization of the recognition problem.* A circle graph *is an intersection graph of chords of a circle. We study the partial representation extension problem for circle graphs, where the input consists of a graph G and a partial representation $\mathcal{R}'$ giving some pre-drawn chords that represent an induced subgraph of G. The question is whether one can extend $\mathcal{R}'$ to a representation $\mathcal{R}$ of the entire graph G, i.e., whether one can draw the remaining chords into a partially pre-drawn representation to obtain a representation of G. Our main result is an $\mathcal{O}(n^3)$ time algorithm for partial representation extension of circle graphs, where n is the number of vertices. To show this, we describe the structure of all representations of a circle graph using split decomposition. This can be of independent interest.*

1 INTRODUCTION

Geometric graph representations are important topics of graph theory and computer science. A frequently studied type of representations are the so-called *intersection representations*. An intersection representation of a graph represents its vertices by some objects and encodes its edges by intersections of these objects, i.e., two vertices are adjacent if and only if the corresponding objects intersect. Classes of intersection graphs are obtained by restricting these objects; e.g., *interval graphs* are intersection graphs of intervals of the real line, *string graphs* are intersection graphs of curves in plane, and so on. These representations are well-studied; see e.g. [40].

For a fixed class $\mathcal{C}$ of intersection-defined graphs, a very natural computational problem is *recognition*. It asks whether an input graph G belongs to $\mathcal{C}$. In this paper, we study a recently introduced generalization of this problem called *partial representation extension* [29]. Its input gives with G a part of the representation and the problem asks whether this partial representation can be extended to a representation of the entire G; see Fig. 1 for an illustration. We show that this problem can be solved in polynomial time for the class of *circle graphs*.

Circle Graphs. Circle graphs are intersection graphs of chords of a circle. They were first considered by Even and Itai [19] in the early 1970s in study of stack sorting techniques. Other motivations are due to their

---

* The conference version of this paper appeared in Graph Drawing 2013 [10].

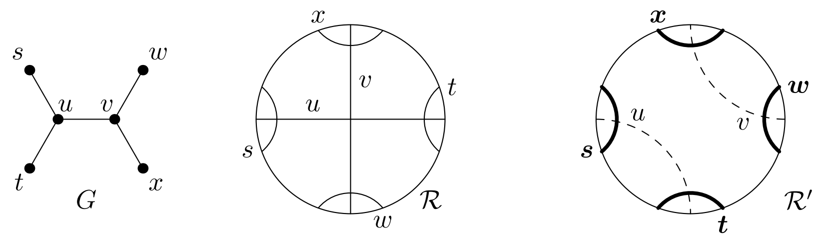

FIGURE 1. On the left, a circle graph G with a representation $\mathcal{R}$ is given. A partial representation $\mathcal{R}'$ given on the right with the pre-drawn chords $\boldsymbol{s}$, $\boldsymbol{t}$, $\boldsymbol{w}$, and $\boldsymbol{x}$ is not extendible. The chords are depicted as arcs to make the figure more readable.

relations to Gauss words [18] (see Fig. 2) and matroid representations [17, 8]. Circle graphs are also important regarding rank-width [35].

Let $\chi(G)$ denote the chromatic number of G, and let $\omega(G)$ denote the clique-number of G. Trivially we have $\omega(G) \le \chi(G)$ and the graphs for which every induced subgraph satisfies equality are the well-known *perfect graphs* [12]. In general, the difference between these two numbers can be arbitrarily high, e.g., there is a triangle-free graph with an arbitrary high chromatic number. Circle graphs are known to be *almost perfect* which means that $\chi(G) \le f(\omega(G))$ for some function f. The best known result for circle graphs [31] states that $f(k)$ is $\Omega(k \log k)$ and $\mathcal{O}(2^k)$.

Some NP-hard problems, such as maximum weighted clique and independent set [22], become tractable on circle graphs. On the other hand, problems such as vertex colorability [21] and Hamiltonicity [16] remain NP-complete even for circle graphs.

The complexity of recognition of circle graphs was a long standing open problem; see [40] for an overview. The first results, e.g., [19], gave existential characterizations which did not give polynomial-time algorithms. The mystery whether circle graphs can be recognized in polynomial time frustrated mathematicians for some years. It was resolved in the mid-1980s and several polynomial-time algorithms were discovered [7, 20, 33] (in time $\mathcal{O}(n^7)$ and similar). Later, a more efficient algorithm [39] based on *split decomposition* was given, and the current state-of-the-art recognition algorithm [23] runs in a quasi-linear time in the number of vertices and the number of edges of the graph.

The Partial Representation Extension Problem. It is quite surprising that this very natural generalization of the recognition problem was considered only recently. It is currently an active area of research which is inspiring a deeper investigation of many classical graph classes. For instance, a recent result of Angelini et al. [1] states that the problem is decidable in linear time for planar graphs. On the other hand, Fáry's Theorem claims that every planar graph has a straight-line embedding, but extension of such an embedding is NP-hard [36].

In the context of intersection-defined classes, this problem was first considered in [29] for interval graphs. Currently, the best known results are linear-time algorithms for interval graphs [5, 28] and proper interval graphs [26], a quadratic-time algorithm for unit interval graphs [26, 37, 38], and polynomial-time algorithms for permutation and function graphs [25], proper circular-arc graphs [3], and trapezoid graphs [32]. For chordal graphs (as subtree-in-a-tree graphs) several versions of the problems were considered [27] and all of them are NP-complete, and similarly for different contact representations of planar graphs [9]. In [30], minimal forbidden configurations making a partial interval representation non-extendible are characterized. Extending partial visibility representations is studied in [11].

The Structure of Representations. To solve the recognition problem for G, one just needs to build a single representation. However, to solve the partial representation extension problem, the structure of all representations of G must be well understood. A general approach used in the above papers is the following.

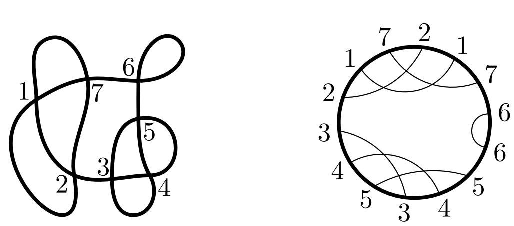

FIGURE 2. A self-intersecting closed curve with n intersections numbered $1, \ldots, n$ corresponds to a representation of circle graph with the vertices $1, \ldots, n$ where the endpoints of the chords are placed according to the order of the intersections along the curve.

We first derive necessary and sufficient constraints from the partial representation $\mathcal{R}'$. Then we efficiently test whether some representation $\mathcal{R}$ satisfies these constraints. If none satisfies them, then $\mathcal{R}'$ is not extendible. And if some $\mathcal{R}$ satisfies them, then it extends $\mathcal{R}'$.

It is well-known that the split decomposition [14, Theorem 3] captures the structure of all representations of circle graphs. The standard recognition algorithms produce a special type of representations using split decomposition as follows. We find a *split* in G, construct two smaller graphs, build their representation recursively, and then join these two representations to produce $\mathcal{R}$. In Section 3, we give a simple recursive description of all possible representations based on splits. Our result can be interpreted as "describing a structure like PQ-trees[1] for circle graphs." It is possible that the proof techniques from other papers on circle graphs such as [13, 23] would give a similar description. However, these techniques are more involved than our approach which turns out to be quite elementary and simple.

Restricted Representations. The partial representation extension problem belongs to a larger group of problems dealing with *restricted representations of graphs*. These problems ask whether there is some representation of an input graph G satisfying some additional constraints. We describe two examples of these problems.

An input of the *simultaneous representation problem*[2], shortly SIM, consists of graphs $G_1, \dots, G_k$ with some vertices common for all the graphs. The problem asks whether there exist representations $\mathcal{R}_1, \dots, \mathcal{R}_k$ representing the common vertices in the same way. This problem is polynomially solvable for permutation and comparability graphs [24]. They additionally show that for chordal graphs it is NP-complete when k is part of the input and polynomially solvable for $k = 2$. For interval graphs, a linear-time algorithm is known for $k = 2$ [5] and the complexity is open in general. For some classes, these problems are closely related to the partial representation extension problems. For example, there is an FPT algorithm for interval graphs with the number of common vertices as the parameter [29], and partial representations of interval graphs can be extended in linear time by reducing it to corresponding simultaneous representation problem [5].

The *bounded representation problem* [26] prescribes bounds for each vertex of the input graph and asks whether there is some representation satisfying these bounds. For circle graphs, the input specifies for each chord v a pair of arcs (A_v, A'_v) of the circle, and a solution is required to have one endpoint of v in A_v and the other one in A'_v. This problem is clearly a generalization of partial representation extension since one can describe a partial representation using singleton arcs. It is known to be polynomially solvable for interval and proper interval representations of interval graphs [2], and surprisingly it is NP-complete for unit interval representations [26, 37, 38]. The complexity for other classes is not known.

Our Results. We study the following problem (see Section 2 for definitions):

Problem:	Partial Representation Extension – REPEXT(CIRCLE)
Input:	A circle graph G and a partial representation $\mathcal{R}'$.
Output:	Is there a representation $\mathcal{R}$ of G extending $\mathcal{R}'$?

In Section 3, we describe a simple structure of all representations. This is used in Section 4 to obtain our main algorithmic result:

Theorem 1. The problem REPEXT(CIRCLE) can be solved in time $\mathcal{O}(n^3)$ where n is the number of vertices.

To spice up our results, we show in Section 5 the following for the simultaneous representation problem of circle graphs:

Theorem 2. If k is a part of the input, the problem SIM(CIRCLE) of k circle graphs is NP-complete.

Finally, we show that Theorem 1 implies the following.

Corollary. The problem SIM(CIRCLE) is FPT in the size of the common subgraph.

---

[1]See [6] for further information on PQ-trees.

[2]Here, we will focus on what is sometimes referred to as the *sunflower version* in the literature, see [4].

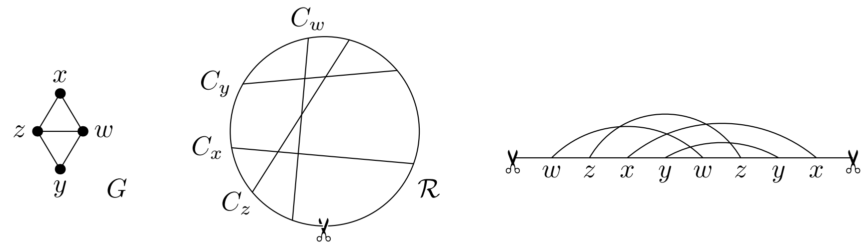

FIGURE 3. An example of a circle graph with a circle graph representation on the left; an interval overlap representation of the same graph on the right.

2 DEFINITIONS AND PRELIMINARIES

Circle Representations. A *circle representation* $\mathcal{R}$ of a graph G is a collection $\{C_u \mid u \in V(G)\}$ of chords of a circle such that C_u intersects C_v if and only if $uv \in E(G)$. A graph is a *circle graph* if it has a circle representation, and we denote the class of circle graphs by CIRCLE.

Notice that a representation of a circle graph is completely determined by the circular order of the endpoints of the chords in the representation, and two chords C_u and C_v cross if and only if their endpoints alternate in this order. For convenience we label both endpoints of the chord representing a vertex by the same label as the vertex.

Interval Overlap Graphs. Suppose that we cut the circle in a point which is not an endpoint of a chord and straighten it into a segment; see Fig. 3. From this straightening of the circle, each chord can now be seen as an arc above the resulting segment. Notice that two chords C_u and C_v cross if and only if their endpoints appear in the order $uvuv$ or $vuvu$ from left to right. Alternatively, circle graphs are called *interval overlap graphs.* Their vertices can be represented by intervals and two vertices are adjacent if and only if their intervals overlap which means they intersect and one is not a subset of the other.

Word representations. A sequence τ over an alphabet of symbols Σ is a *word.* A *circular word* represents the set of words which are cyclical shifts of one another. In the sequel, we represent a circular word by a word from its corresponding set of words. We denote words and circular words by small Greek letters.

For a word τ and a symbol u we write $u \in \tau$, if u appears at least once in τ. Thus, τ is also used to denote the set of symbols occurring in τ. A word τ is a *subword* of σ, if τ appears consecutively in σ. A word τ is a *subsequence* of σ, if the word τ can be obtained from σ by deleting some symbols. We say that u *alternates* with v in τ, if $uvuv$ or $vuvu$ is a subsequence of τ. The corresponding definitions also apply to circular words. If σ and τ are two words, we denote their concatenation by $\sigma\tau$.

The above interpretation of circle graphs as interval overlap graphs allows us to associate each representation $\mathcal{R}$ of G with a unique circular word τ over V. The word τ is obtained by the circular order of the endpoints of the chords in $\mathcal{R}$ as they appear along the circle when traversed clockwise. The occurrences of u and v alternate in τ if and only if $uv \in E(G)$. For example $\mathcal{R}$ in Fig. 1 corresponds to the circular word $\tau = susxvxtutwvw$. Notice that each vertex appears exactly twice in τ. A circular subsequence τ' of τ is *induced* by $V' \subseteq V(G)$ if τ' is obtained from τ by deleting symbols in $V(G) \setminus V'$.

Partial Representations. Partial representations are defined in [29] and other papers as representations of induced subgraphs. In this paper, we consider the following more general definition. A *partial representation* $\mathcal{R}'$ of a circle graph G is given by a circular word τ' consisting of symbols of $V(G)$ such that each $u \in V(G)$ appears at most twice in τ'. A representation $\mathcal{R}$ of G corresponding to a circular word τ *extends* $\mathcal{R}'$ if and only if τ' is a subsequence of τ. The endpoints in τ' and the corresponding vertices are called *pre-drawn.* If a pre-drawn vertex u has both occurrences in τ', the chord C_u is *pre-drawn.*

3 STRUCTURE OF REPRESENTATIONS OF MAXIMAL SPLITS

Let G be a connected graph. A *split* of G is a partition of the vertices of G into four parts A, B, $\mathfrak{s}(A)$ and $\mathfrak{s}(B)$, such that:

- We have $A \neq \emptyset$ and $B \neq \emptyset$, but possibly $\mathfrak{s}(A) = \emptyset$ or $\mathfrak{s}(B) = \emptyset$.
- For every $a \in A$ and $b \in B$, we have $ab \in E(G)$.
- There is no edge between $\mathfrak{s}(A)$ and $B \cup \mathfrak{s}(B)$, and between $\mathfrak{s}(B)$ and $A \cup \mathfrak{s}(A)$.

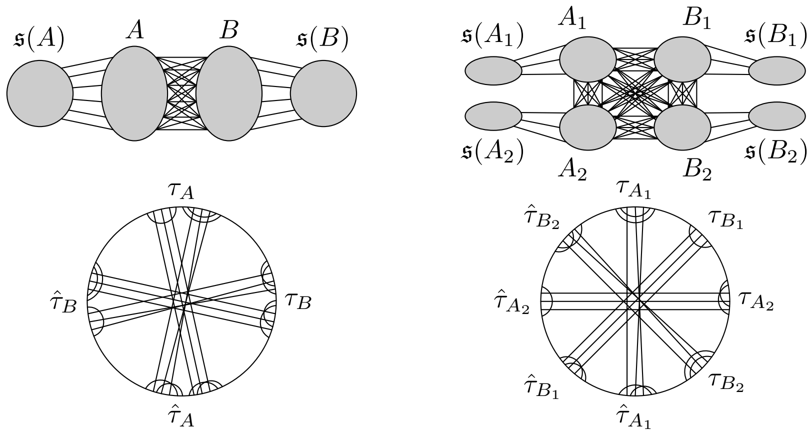

FIGURE 4. Two different representations of G with the split between A and B.

Fig. 4 shows two possible representations of a split. Notice that a split is uniquely determined just by the sets A and B, since $\mathfrak{s}(A)$ consists of connected components of $G \setminus (A \cup B)$ attached to A, and $\mathfrak{s}(B)$ of those attached to B. We refer to this split as the split *between* A and B. Alternatively, a split between A and B is a cut in G between A and B which is a complete bipartite graph.

The standard assumption is that a split is *non-trivial*, meaning that both sides of the split have at least two vertices: $|A \cup \mathfrak{s}(A)| \geq 2$ and $|B \cup \mathfrak{s}(B)| \geq 2$. The reason is that trivial splits are not very interesting: in every graph G, the choice $A = \{u\}$ and $B = N(u)$ for $u \in V(G)$ forms a trivial split. The goal of split decomposition is to divide a graph into smaller graphs and trivial splits are not helpful.

One of the novelties of this paper is that we study maximal splits. A split of G between A and B is *maximal* if there exists no split of G between A' and B' such that $A \subseteq A'$, $B \subseteq B'$ and $|A| < |A'|$ or $|B| < |B'|$. Both splits between A and B and between A' and B' are allowed to be trivial. Maximal splits satisfy the following property:

Lemma 1. A split between A and B is maximal, if and only if there exists no connected component C in $\mathfrak{s}(A)$ such that each vertex of C is either adjacent to all vertices of A, or to none of them, and similarly for $\mathfrak{s}(B)$ and B.

Proof. Suppose that such a component C in $\mathfrak{s}(A)$ exists, and let $C' \subseteq C$ consists of those vertices which are adjacent to all vertices of A. The split between A and B is not maximal since A and $B \cup C'$ forms a split, for which $C \setminus C' \subseteq \mathfrak{s}(B \cup C')$. Similarly, if such a component C in $\mathfrak{s}(B)$ exists, the split between A and B is not maximal.

On the other hand, suppose that a split between A and B is not maximal, so there exists a split between A' and B' such that, without loss of generality, $A \subseteq A'$ and $B \subsetneq B'$. Since every vertex of $B' \setminus B$ is adjacent to all vertices in A, we have $B' \setminus B \subseteq \mathfrak{s}(A)$. Choose an arbitrary $c \in B' \setminus B$ and let C be its connected component of $\mathfrak{s}(A)$. As argued, all vertices of $V(C) \cap B'$ are adjacent to all vertices of A. Since $V(C) \cap B' \neq$ and $V(C) \cap A = \emptyset$, the remaining vertices $V(C) \setminus B' \subseteq \mathfrak{s}(B')$. Therefore, they are non-adjacent to all vertices A, and C satisfies the properties from the statement of this lemma. ∎

We always start with a non-trivial split between A and B, and modify it using Lemma 1 into a maximal split which may become trivial. But such a trivial maximal split has a special structure, described below:

Lemma 2. Let A and B form a non-trivial split and let A' and B' form a trivial maximal split such that $A \subseteq A'$, $B \subseteq B'$, $A' = \{a\}$, and $\mathfrak{s}(A') = \emptyset$. Then a is an articulation in G, i.e., $G \setminus a$ is disconnected.

Proof. Since $A \neq \emptyset$, we have $A = \{a\}$. Since the split between A and B is non-trivial, we have $\mathfrak{s}(A) \neq \emptyset$. Therefore, a is an articulation in G which separates $\mathfrak{s}(A)$ from $B \cup \mathfrak{s}(B)$. ∎

In the rest of this section, we examine the recursive structure of every possible representation of G based on maximal splits. In Section 3.1, we analyze the structure of a representation of a maximal split. In Section 3.2, we use it to describe the structure of all circle representations. The described results still apply to trivial maximal splits, but are not very helpful. Therefore, in Section 3.3, we give a different description of all representations based on trivial maximal splits.

3.1 Structure of a Representation of a Maximal Split

Let $\mathcal{R}$ be a representation of a graph G with a maximal split between A and B. The representation $\mathcal{R}$ corresponds to a unique circular word τ. We consider the circular subsequence γ of τ induced by $A \cup B$. The maximal subwords of γ consisting of vertices of A alternate with the maximal subwords of γ consisting of vertices of B. We denote all these maximal subwords $\gamma_1, \ldots, \gamma_{2k}$ according to their circular order; so $\gamma = \gamma_1\gamma_2 \cdots \gamma_{2k}$. Without loss of generality, we assume that γ_1 consists of symbols from A. We call γ_i an *A-word* when i is odd, and a *B-word* when i is even.

We first investigate for each γ_i which symbols it contains.

Lemma 3. For the subwords $\gamma_1, \ldots, \gamma_k$ the following holds:

(a) Each γ_i contains each symbol at most once.
(b) The value of k is even and the opposite words γ_i and γ_{i+k} contain the same symbols.
(c) Let $i \neq j$. If $x \in \gamma_i$ and $y \in \gamma_j$, then $xy \in E(G)$.

Proof. (a) For every $a \in A$ and $b \in B$, the fact $ab \in E(G)$ implies that a and b alternate in the circular word γ. So if some γ_i contains both occurrences of, say, a, then a and b would not alternate in γ.

(b) Let γ_i be, say, an A-word. We first prove that all the other occurrences of the symbols from γ_i are contained in one word γ_j; so we get a matching between the words. Suppose that this is not true and there is $x \in \gamma_i, \gamma_j$ and $y \in \gamma_i, \gamma_{j'}$ for distinct i, j and j'. There is at least one B-word γ_ℓ placed in between γ_j and $\gamma_{j'}$ (in the part of the circle not containing γ_i). It is not possible for $z \in \gamma_\ell$ to alternate with both x and y, which contradicts $xz, yz \in E(G)$.

Now, let γ_i and γ_j be two matched A-words. Then every pair of matched B-words must occur on opposite sides of the circle with respect to γ_i and γ_j. Therefore the same number of B-words occur on both sides of γ_i and γ_j, and thus $j = i + k$.

(c) This is implied by (a) and (b) since the occurrences of x and y alternate in γ. ∎

Below, we prove that the structure of a maximal split between A and B greatly restricts possible representation of the vertices of $\mathfrak{s}(A) \cup \mathfrak{s}(B)$:

Lemma 4. Let τ and $\gamma_1, \ldots, \gamma_{2k}$ be defined as above. There exists a unique mapping $f : \mathfrak{s}(A) \cup \mathfrak{s}(B) \to \{1, \ldots, 2k\}$ satisfying the following properties:

(a) For $c \in \mathfrak{s}(A) \cup \mathfrak{s}(B)$, let $c\tau_c c\tau'_c$ be the subsequence of τ induced by $A \cup B \cup \{c\}$. Then either τ_c, or τ'_c is a subword of $\gamma_{f(c)}$. For $c \in \mathfrak{s}(A)$, the word $\gamma_{f(c)}$ is an A-word, while for $c \in \mathfrak{s}(B)$, it is a B-word.
(b) For each connected component C of $\mathfrak{s}(A) \cup \mathfrak{s}(B)$, the mapping $f|_C$ is constant, i.e., for all $c, c' \in C$, we have $f(c) = f(c')$, and we denote the image by $f(C)$.

Proof. Without loss of generality, we assume that $c \in \mathfrak{s}(A)$; a symmetric argument works for $c \in \mathfrak{s}(B)$. We first prove the existence and uniqueness of $\gamma_{f(c)}$ when c is adjacent to some vertex in $a \in A$. Since c alternates with a, both τ_c and τ'_c are non-empty. In (b), we prove that $f(c) = f(c')$ when $cc' \in E(G)$, so by induction the existence and uniqueness follows for all vertices of C.

(a) Since c alternates with $a \in A$, if such $\gamma_{f(c)}$ exists, then it is an A-word. Since A-words and B-words alternate in $\gamma = \gamma_1 \cdots \gamma_{2k}$, we get that τ_c is a subword of some A-word γ_i if and only if it contains no symbol from B. Since at most one of τ_c and τ'_c contains no symbol from B, it is easy to see that such γ_i is unique. It remains to prove that it always exists.

Let C be the connected component of $\mathfrak{s}(A)$ containing c. For contradiction, suppose that the property (a) fails for c. If property (a) fails for c, we have $b \in \tau_c$ and $b' \in \tau'_c$ such that $b, b' \in B$, $b \neq b'$. Since $bc, b'c \notin E(G)$, we also have $bb' \notin E(G)$; see Fig. 5(a).

For each $a \in A$, we have $ab, ab' \in E(G)$, so $A \cup \{b, b', c\}$ induces in τ the subsequence $cb\alpha bcb'\alpha' b'$ such that both α and α' consist only of symbols from A. For each $a \in A$, we have $a \in \alpha$ and $a \in \alpha'$, so c is adjacent to all vertices of A. Since every vertex $c' \in C$ is connected by a path to c, we have that $A \cup \{b, b'\} \cup C$ induces in τ the subsequence $\sigma b\alpha b\sigma' b'\alpha' b'$, where both σ and σ' consist of symbols of C. Therefore every $c' \in C$ is either adjacent to all vertices of A, or to none of them. By, Lemma 1, the split between A and B is not maximal.

(b) Let $c, c' \in \mathfrak{s}(A)$ such that $cc' \in E(C)$ and $f(c)$ is already determined. We want to prove that $f(c) = f(c')$. As depicted in Fig. 5(b), let $c\tau_c c'\hat{\tau}_c c\tau'_c c'\hat{\tau}'_c$ be the subsequence of τ induced by $A \cup B \cup \{c, c'\}$, and suppose that $\tau_c\hat{\tau}_c$ is a subword of $\gamma_{f(c)}$. Both τ'_c and $\hat{\tau}'_c$ cannot contain symbols from B, otherwise c' alternates with

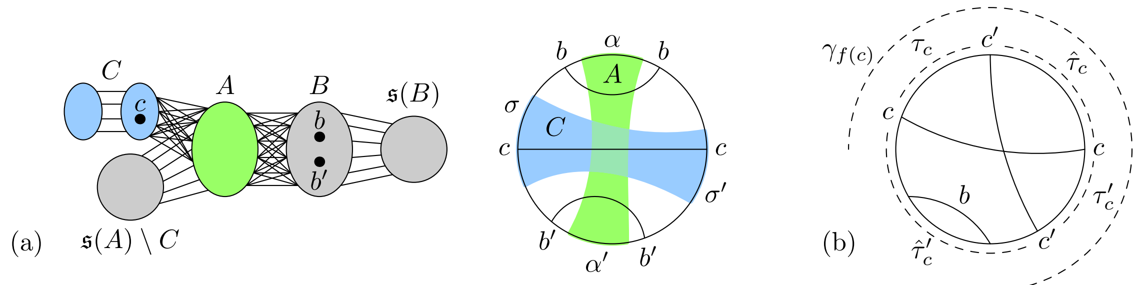

FIGURE 5. (a) On the left, a connected component C of $\mathfrak{s}(A)$ attached to A. On the right, the circular subsequences of τ induced by $A \cup \{b, b', c\}$ and $A \cup \{b, b'\} \cup C$. (b) The circular subsequence $c\tau_c c'\hat{\tau}_c c\tau'_c c'\hat{\tau}'_c$ induced by $A \cup B \cup \{c, c'\}$, with the word $\gamma_{f(c)}$ depicted. Exactly one of τ'_c and $\hat{\tau}'_c$ contains the symbols $b \in B$.

A and the argument in (a) applies. Therefore, either $\hat{\tau}_c\tau'_c$, or $\hat{\tau}'_c\tau_c$ is a subword of $\gamma_{f(c)}$, so $f(c) = f(c')$. We note that either $\hat{\tau}_c\tau'_c$, or $\hat{\tau}'_c\tau_c$ might be empty, so $f(c')$ could be chosen arbitrarily to satisfy (a). We then set $f(c') = f(c)$ to also satisfy (b). ∎

Let τ_i denote the subsequence of τ formed by γ_i, and of the symbols of $\bigcup_{C:f(C)=i} V(C)$ over all connected components C of $\mathfrak{s}(A) \cup \mathfrak{s}(B)$. By Lemma 4, the only difference between γ and τ is that each subword γ_i is replaced by the subword τ_i which additionally contains all occurrences of the vertices in some connected components of $\mathfrak{s}(A)$ or $\mathfrak{s}(B)$. Thus, $\tau = \tau_1\tau_2\cdots\tau_{2k}$.

Lemma 4 explains the following naming convention used for maximal splits between A and B in this paper; see Fig. 4. We call the vertices of A and B as *long vertices* with respect to the maximal split between A and B since each is represented by "long chords" between τ_i and τ_{i+k}. The vertices $\mathfrak{s}(A)$ and $\mathfrak{s}(B)$ are called *short vertices* with respect to the maximal split between A and B, because each is represented by "short chords" inside some τ_i. In the sequel, if the maximal split is clear from the context, we will just call some vertices long and some vertices short.

Lemma 5. If two long vertices $x, y \in A \cup B$ are connected by a path of length at least two having the internal vertices in $\mathfrak{s}(A) \cup \mathfrak{s}(B)$, then x and y belong to the same pair γ_i and γ_{i+k} in every representation.

Proof. Let C be the connected component of $\mathfrak{s}(A) \cup \mathfrak{s}(B)$ having the internal vertices of this path between x and y. By Lemma 4, all vertices of C have both symbols in $\tau_{f(C)}$. Therefore, $x, y \in \tau_{f(C)}$. So $x, y \in \gamma_{f(C)}$, and by Lemma 3(b) also $x, y \in \gamma_{f(C)+k}$. ∎

Also, we prove the following simple lemma:

Lemma 6. Let x, y, z, and w be distinct vertices inducing a clique in G, and let P be a path from x to y of length at least 2. If $xzywxzyw$ is a subsequence of the circular word τ of a circle representation of G, then some internal vertex of P is adjacent to z or w.

Proof. Let $v_1, \dots, v_k$ be the internal vertices of P such that $v_1x \in E(G)$. We prove by induction that no v_i having $v_iz \in E(G)$ or $v_iw \in E(G)$ implies that $v_ky \notin E(G)$. If v_1 is not such a vertex, then we get that $v_1xv_1zywxzyw$ is a subsequence of τ since $v_1x \in E(G)$. For the induction hypothesis, suppose that $v_iv_izywzyw$ is a subsequence of τ. Since $v_iv_{i+1} \in E(G)$, if v_{i+1} is not adjacent to z and w, we get that $v_{i+1}v_{i+1}zywzyw$ is a subsequence of τ. Therefore, v_k does not alternate with y, contradicting that $v_ky \in E(G)$. ∎

3.2 Conditions Forced by a Maximal Split

Now, we want to investigate the opposite relation. Namely, what can one say about a representation from the structure of a maximal split? Suppose that x and y are two long vertices. We want to know the properties of x and y which force every representation $\mathcal{R}$ to have a subword γ_i of γ containing both x and y.

Inspired by Naji [33, Section IV.4], we define a symmetric relation $\sim$ on $A \cup B$ where $x \sim y$ means that x and y must occur in the same subword γ_i of γ. This relation is given by two conditions:

(C1) Lemma 3(c) states that if $xy \notin E(G)$, then $x \sim y$, i.e., if x and y are placed in different subwords, then C_x intersects C_y. In particular, $x \sim x$.

(C2) Lemma 5 gives $x \sim y$ when x and y are connected by a non-trivial path with all the inner vertices in $\mathfrak{s}(A) \cup \mathfrak{s}(B)$.

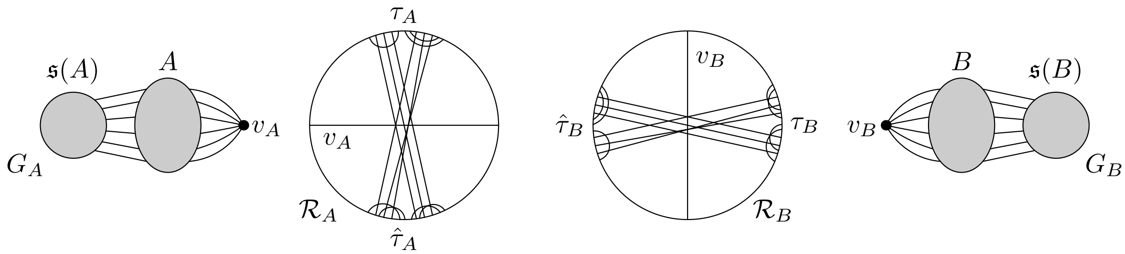

FIGURE 6. The graphs G_A and G_B together with some constructed representations $\mathcal{R}_A$ and $\mathcal{R}_B$. By joining these representations, we get the representation shown on the left in Fig. 4.

Let us take the transitive closure of $\sim$, which we denote by $\sim$ thereby slightly abusing the notation. Thus, we obtain an equivalence relation $\sim$ on $A \cup B$. Notice that every equivalence class of $\sim$ is either fully contained in A or in B. Figure 4 on right shows schematically a situation in which the relation $\sim$ has four equivalence classes A_1, A_2, B_1 and B_2.

Now, let Φ be an equivalence class of $\sim$. We denote by $\mathfrak{s}(\Phi)$ the set consisting of all the vertices in the connected components of $\mathfrak{s}(A) \cup \mathfrak{s}(B)$ which have a vertex adjacent to a vertex of Φ. Since $\sim$ satisfies (C2), we know that the sets $\mathfrak{s}(\Phi)$ of the equivalence classes of $\sim$ define a partition of $\mathfrak{s}(A) \cup \mathfrak{s}(B)$.

Recognition Algorithms Based on Splits. Split decompositions are used in the current state-of-the-art algorithms for recognizing circle graphs. If a circle graph contains no split, it is called a *prime graph*. The representation of a prime graph is uniquely determined (up to the orientation of the circle) and can be constructed efficiently. There is an algorithm which finds a split in a graph in linear time [15]. In fact, the entire *split decomposition tree* (i.e., the recursive decomposition tree obtained via splits) can be found in linear time. Usually the representation $\mathcal{R}$ is constructed as follows.

We define two graphs G_A and G_B where G_A is created from G by contracting the vertices of $B \cup \mathfrak{s}(B)$ into a new vertex v_A and G_B by contracting $A \cup \mathfrak{s}(A)$ into a new vertex v_B. So v_A is adjacent to all vertices in A and to no vertices in $\mathfrak{s}(A)$, and similarly for v_B. Then we apply the algorithm recursively on G_A and G_B and construct their representations $\mathcal{R}_A$ and $\mathcal{R}_B$; see Fig. 6. It remains to join the representations $\mathcal{R}_A$ and $\mathcal{R}_B$ in order to construct $\mathcal{R}$.

To this end we take $\mathcal{R}_A$ and replace C_{v_A} by the representation of $B \cup \mathfrak{s}(B)$ in $\mathcal{R}_B$. More precisely, let the circular ordering of the endpoints of chords defined by $\mathcal{R}_A$ be $v_A \tau_A v_A \hat{\tau}_A$ and let the circular ordering defined by $\mathcal{R}_B$ be $v_B \tau_B v_B \hat{\tau}_B$. The constructed $\mathcal{R}$ has the corresponding circular ordering $\tau_A \tau_B \hat{\tau}_A \hat{\tau}_B$. It is easy to see that $\mathcal{R}$ is a correct circle representation of G.

Structure of All Representations. The above algorithm constructs a very specific representation $\mathcal{R}$ of G, and a representation like the one in Fig. 4 on the right cannot be constructed in this way using the split between A and B. In what follows we describe the structure of all the representations of a circle graph G based on the different circular orderings of the equivalence classes of $\sim$. While the described structure of all the representations depends on the maximal split that we chose, the relation $\sim$ defined with respect to this maximal split can be used to generate all the representations of G.

We choose an arbitrary circular ordering $\Phi_1, \dots, \Phi_\ell$ of the classes of $\sim$. Let G_i be a graph constructed from G by contracting the vertices $V(G) \setminus \bigl(\Phi_i \cup \mathfrak{s}(\Phi_i)\bigr)$ into one vertex v_i; i.e., G_i is defined similarly to G_A and G_B above. Let $\mathcal{R}_1, \dots, \mathcal{R}_\ell$ be arbitrary representations of $G_1, \dots, G_\ell$. We join these representations as follows. Let $v_i \tau_i v_i \hat{\tau}_i$ be the circular ordering of $\mathcal{R}_i$. We construct $\mathcal{R}$ as the circular ordering

$$\tau_1 \tau_2 \dots \tau_{\ell-1} \tau_\ell \hat{\tau}_1 \hat{\tau}_2 \dots \hat{\tau}_{\ell-1} \hat{\tau}_\ell. \tag{1}$$

In Fig. 4, we obtain the representation on the left by the circular ordering $A_1 A_2 B_1 B_2$ of the classes of $\sim$ and the representation on the right by $A_1 B_1 A_2 B_2$.

First, we show that every representation obtained in this way is correct.

Lemma 7. Every circular ordering (1) constructed as above defines a circle representation of G.

Proof. Let $u, v \in V(G)$. We shall prove that u and v are adjacent in G if and only if they alternate in $\mathcal{R}$. Suppose that $u, v \in V(G_i) \setminus \{v_i\}$. Since the cyclic subsequence $\tau_i \hat{\tau}_i$ appears in both $\mathcal{R}_i$ and $\mathcal{R}$, two vertices in $V(G_i) \setminus \{v_i\}$ alternate in $\mathcal{R}$ if and only if they are adjacent in G_i, which is if and only if they are adjacent in G.

Otherwise, let $u \in V(G_i) \setminus \{v_i\}$ and $v \in V(G_j) \setminus \{v_j\}$ for $i \neq j$. Then $uv \in E(G)$ if and only if they are both long vertices. Each long vertex of Φ_t appears once in both τ_t and $\hat{\tau}_t$, but each short vertex $\mathfrak{s}(\Phi_t)$ has both its occurrences either in τ_t, or in $\hat{\tau}_t$. We conclude that u and v alternate in $\mathcal{R}$ if and only if they are both long vertices, i.e., if and only if they are adjacent in G since u and v do not satisfy (C1). ∎

Next, we analyze every representation $\mathcal{R}$ of G.

Lemma 8. Let τ be the circular word corresponding to a representation $\mathcal{R}$ of G. Then the symbols of $\Phi_i \cup \mathfrak{s}(\Phi_i)$ form exactly two subwords τ_i and $\hat{\tau}_i$ of τ such that for each $u \in \Phi_i$, we have $u \in \tau_i$ and $u \in \hat{\tau}_i$, while each $v \in \mathfrak{s}(\Phi_i)$ has both endpoint either in τ_i, or in $\hat{\tau}_i$.

Proof. Let $\mathcal{R}$ be a representation of G and consider how it represents $A \cup B$. We get the subwords $\gamma_1, \dots, \gamma_{2k}$ of the endpoints of $A \cup B$, as described in Section 3.1.

Let $x \in \Phi_i$ such that $x \in \gamma_j$. We claim that Φ_i is a subset of γ_j. Since Φ_i is an equivalence class of $\sim$, let $y \in \Phi_i$ such that one of the conditions (C1) or (C2) applies to x and y. Since $\sim$ is the transitive closure of conditions (C1) and (C2), to prove the claim, it is sufficient to show that $y \in \gamma_j$. If (C1) applies, then $y \in \gamma_j$ by Lemma 3(c). If (C2) applies, then $y \in \gamma_j$ by Lemma 5. By Lemma 3(a), each vertex of Φ_i appears exactly once in γ_j and once in γ_{j+k}.

Furthermore, we claim that the vertices of Φ_i form subwords of γ_j and γ_{j+k}. Let $z \in \gamma_j$ be placed between $x \in \Phi_i$ and $y \in \Phi_i$. First, we assume that (C1) or (C2) applies to x and y.

- If (C1) applies to x and y, then $xy \notin E(G)$. As x and y do not alternate, it is not possible for z to alternate with both x and y. Thus $z \sim x$ or $z \sim y$, which in turn implies that $z \in \Phi_i$.
- Suppose that (C2) applies to x and y. If $xz \notin E(G)$ or $yz \notin E(G)$, we get that $z \in \Phi_i$ by (C1). Otherwise, we claim that a path P from x to y having all the internal vertices in $\mathfrak{s}(\Phi_i)$ has at least one internal vertex adjacent to z. For every $w \in \gamma_{j+1}$ and $w \in \gamma_{j+1+k}$, we have $xw, yw, zw \in E(G)$, but none of the inner vertices of P are adjacent to w. Since $\{x, y, z, w\}$ induce the subsequence $xzywxzyw$ in τ, by Lemma 6 some inner vertex P has to alternate with z. Thus, $z \sim x$ and $z \sim y$ by (C2), so $z \in \Phi_i$.

If $x \sim y$ and neither of (C1) and (C2) applies, we easily proceed by an inductive argument on the number of applications of (C1) and (C2). If $x \sim y' \sim y$ and a vertex $z \in \gamma_j$ is placed between x and y in γ_j, then z is also placed in γ_j between x and y' or between y' and y.

By the above argument, each class Φ_i forms two subwords of γ. By adding the short vertices $\mathfrak{s}(\Phi_i)$ as in Lemma 4 applied on the maximal split between Φ_i and $A \cup B \setminus \Phi_i$, we obtain two subwords of τ for each class Φ_i. ∎

Now, we are ready to prove the main structural proposition.

Proposition 1. Let A and B form a maximal split of G and let $\sim$ be the equivalence relation defined by (C1) and (C2) on $A \cup B$. Then every representation $\mathcal{R}$ of G corresponds to some circular ordering $\Phi_1, \dots, \Phi_\ell$ and to some representations $\mathcal{R}_1, \dots, \mathcal{R}_\ell$ of $G_1, \dots, G_\ell$. More precisely, $\mathcal{R}$ can be constructed by arranging $\mathcal{R}_1, \dots, \mathcal{R}_\ell$ as in (1): $\tau_1 \dots \tau_\ell \hat{\tau}_1 \dots \hat{\tau}_\ell$.

Proof. By Lemma 7, every representation constructed by (1) is correct. On the other hand, let $\mathcal{R}$ be a representation of G with the corresponding circular word τ. According to Lemma 8, we know that $\Phi_i \cup \mathfrak{s}(\Phi_i)$ forms two subwords τ_i and $\hat{\tau}_i$ of τ. For $i \neq j$, the edges between Φ_i and Φ_j form a complete bipartite graph. The subwords τ_i, $\hat{\tau}_i$, τ_j and $\hat{\tau}_j$ alternate, i.e., appear as $\tau_i \tau_j \hat{\tau}_i \hat{\tau}_j$ or $\tau_j \tau_i \hat{\tau}_j \hat{\tau}_i$ in τ. Thus, if we start from some point along the circle, the order of τ_i's gives a circular ordering $\Phi_1, \dots, \Phi_\ell$ of the classes. The representation $\mathcal{R}_i$ has the circular word $v_i \tau_i v_i \hat{\tau}_i$. ∎

3.3 The Structure of All Representations of Trivial Maximal Splits

Let A and B form a trivial maximal split with $A = \{a\}$ and $\mathfrak{s}(A) = \emptyset$, created from a non-trivial split. The results described in Sections 3.1 and 3.2 still apply to this split, but they are not very helpful. By Lemma 2, a is an articulation in G. So, $G[B]$ consists of at least two connected components and $\sim$ has two equivalence classes $\Phi_1 = A$ and $\Phi_2 = B$. Since $G_2 \cong G$, Proposition 1 describes all representations of G in terms of all representations of G.

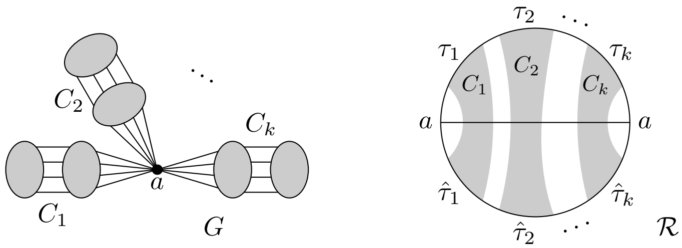

FIGURE 7. If a is an articulation, then every circle representation $\mathcal{R}$ of G consists of some ordering of connected components $C_1, \dots, C_\ell$ of $G \setminus a$ and it corresponds to the depicted circular word τ in which $a\tau_i a\hat{\tau}_i$ is some representation of the subgraph of G induced by $V(C_i) \cup \{a\}$.

In this section, we show that all possible representations can be easily described in a different way, based on all different representations of connected components of $G \setminus a$. The following lemma states that connected components do not alternate in any circle representation:

Lemma 9. Let C and C' be two distinct connected component of a circle graph G. No representation has a subword $uxvy$ where $u, v \in V(C)$ and $x, y \in V(C')$.

Proof. Let σ be the subsequence induced by $V(C) \cup V(C')$. We know that $\sigma = \sigma_1 \cdots \sigma_{2k}$ such that σ_i is the maximal subword consisting only of symbols from $V(C)$ if i is odd, and only of symbols from $V(C')$ if i is even. We want to prove that $k = 1$. For contradiction, suppose that $k > 1$. Since C is connected, there exists $u \in C$ such that $u \in \sigma_1$ and $u \in \sigma_i$ for $i > 1$. Since C' is connected, there exists $x \in C'$ such that $x \in \sigma_2\sigma_4 \cdots \sigma_{i-1}$ and $x \in \sigma_{i+1} \cdots \sigma_{2k}$. Since u and x alternate, we have $ux \in E(G)$ which is a contradiction. ∎

We choose an arbitrary ordering of the connected components of $G \setminus a$ as $C_1, \dots, C_\ell$. Let G_i be the subgraph of G induced by $\{a\} \cup V(C_i)$. Let $\mathcal{R}_i$ be an arbitrary representation of G_i having the circular word $a\tau_i a\hat{\tau}_i$. We construct the joined representation $\mathcal{R}$ of G by the circular word

$$a\tau_1\tau_2 \cdots \tau_{\ell-1}\tau_\ell a\hat{\tau}_\ell\hat{\tau}_{\ell-1} \cdots \hat{\tau}_2\hat{\tau}_1; \tag{2}$$

see Fig. 7. First, we prove that every such constructed representation of G is correct:

Lemma 10. Every circular ordering (2) constructed as above defines a circle representation of G.

Proof. Let τ be the circular ordering constructed using (2). Since $V(G_i)$ induces the subsequence $a\tau_i a\hat{\tau}_i$, each G_i is represented correctly in $\mathcal{R}_i$. For $i < j$, the vertices of $V(C_i) \cup V(C_j)$ induce in τ the subsequence $\tau_i\tau_j\hat{\tau}_j\hat{\tau}_i$, so no two vertices $u \in V(C_i)$ and $v \in V(C_j)$ alternate and the non-edges between C_i and C_j are represented correctly. ∎

Next, we analyze every representation $\mathcal{R}$ of G.

Lemma 11. Let τ be the circular word corresponding to a representation $\mathcal{R}$ of G. Then the symbols of $V(C_i)$ form exactly two subwords τ_i and $\hat{\tau}_i$ of τ such that $a\tau_i a\hat{\tau}_i$ is a subsequence of τ.

Proof. Since $V(C_i) \cap B \neq \emptyset$, there exists some $b \in V(C_i)$ which alternates with a, so the symbols of $V(C_i)$ form at least two subwords alternating with a. If $V(C_i)$ would form more than two subwords, then τ has a subsequence $auxva$, where $u, v \in V(C_i)$ and $x \in V(C_j)$ for $j \neq i$. Since some $y \in V(C_j)$ alternates with a, it follows that τ has the subsequence $auxvay$, so we get $uxvy$ which is not possible by Lemma 9. ∎

Now, we are ready to prove the following structural proposition.

Proposition 2. Let $A = \{a\}$ and B form a trivial maximal split of G created from a non-trivial split. Then every representation $\mathcal{R}$ of G corresponds to some ordering $C_1, \dots, C_\ell$ of connected components of $G \setminus a$ and to some representations $\mathcal{R}_1, \dots, \mathcal{R}_\ell$ of $G_1, \dots, G_\ell$ where G_i is the subgraph of G induced by $V(C_i) \cup \{a\}$. More precisely, $\mathcal{R}$ can be constructed by arranging $\mathcal{R}_1, \dots, \mathcal{R}_\ell$ as in (2): $a\tau_1 \dots \tau_\ell a\hat{\tau}_\ell \dots \hat{\tau}_1$.

Proof. By Lemma 10, every representation constructed by (2) is correct. On the other hand, let $\mathcal{R}$ be a representation of G corresponding to a circular word τ. Suppose that $G \setminus a$ has ℓ connected components. The circular word τ defines an ordering $C_1, \dots, C_\ell$ of the connected components of $G \setminus a$ in the following way. By Lemma 11, $\tau = a\tau_1\tau_2 \cdots \tau_\ell a\sigma_\ell\sigma_{\ell-1} \cdots \sigma_1$, where τ_i and precisely one σ_j are two maximal subwords of τ

containing all symbols from $V(C_i)$. Since the connected components $C_1, \dots, C_\ell$ cannot alternate by Lemma 9, we get that σ_i consists of symbols of C_i, i.e., $\sigma_i = \hat{\tau}_i$. Each $a\tau_i a\hat{\tau}_i$ gives some representation $\mathcal{R}_i$ of G_i. ∎

4 ALGORITHM

In this section, we describe an $\mathcal{O}(n^3)$ algorithm for the partial representation extension problem of circle graphs. It is based on the structure of all representations of Section 3. Recall that a partial representation $\mathcal{R}'$ gives a circular word τ' such that each vertex $u \in V(G)$ appears at most twice in τ'. We want to decide whether there exists a representation $\mathcal{R}$ corresponding to a circular word τ such that τ' is a subsequence of τ.

Dealing with Disconnected Graphs. To apply the structural properties of Section 3, we need to work with connected graphs. In general, the partial representation extension problems cannot be trivially restricted to connected inputs, as in the case of most graph problems. In particular, for some classes the problems are polynomial-time solvable for connected inputs and FPT in the number of components for disconnected inputs, but NP-complete in general; see e.g. [26, 27]. The reason is that the components are placed together in one representation and they restrict each other.

In the case of circle graphs, we can deal with disconnected inputs easily. By Lemma 9, we know that τ' cannot contain a subsequence $uxvy$ where u, v belong to one component and x, y to another one. If this happens, we immediately output "no". Otherwise the question of extendibility is equivalent to testing whether each component C is extendible where the partial representation of C is given by the subsequence of τ' containing all occurrences of the vertices of C. So from now on we assume that the input graph G is connected.

Overview. Our algorithm proceeds recursively via split decomposition. For each encountered graph G with a partial representation $\mathcal{R}'$ corresponding to the circular word τ', it proceeds with the following steps:

- If G is prime, we have two possible representations (one is reversal of the other) and we test whether one of them extends τ'. We return the result.
- Otherwise, we find a non-trivial split and modify it into a maximal split between A and B, using Lemma 1. Next, we proceed with one of the following steps.
- In Case I, the maximal split between A and B is non-trivial. We compute the relation $\sim$. We try to determine an ordering $\Phi_1, \dots, \Phi_\ell$ of the equivalence classes of $\sim$ along the circle as in (1) which is compatible with the partial representation $\mathcal{R}'$. This order is partially prescribed by pre-drawn endpoints of short and long vertices and we recurse on testing whether partial representations of different equivalence classes $\Phi \cup \mathfrak{s}(\Phi)$ can be extended. If no ordering is compatible, we stop and output "no".
- In Case II, the maximal split between A and B is trivial with $A = \{a\}$ and $\mathfrak{s}(A) = \emptyset$. We try to determine an ordering $C_1, \dots, C_k$ of the connected components of $G \setminus a$ along the circle as in (2) which is compatible with the partial representation $\mathcal{R}'$. This order is partially prescribed by pre-drawn endpoints of chords and we recurse on testing whether partial representations of different components C can be extended. If no ordering is compatible, we stop and output "no".

For a more detailed overview of the main steps, see Algorithm 1. Now we describe everything in detail.

Testing Correctness of $\mathcal{R}'$. In the beginning, the algorithm tests correctness of the input partial representation. If $u, v \in V(G)$ have both occurrences in τ', we check that these occurrences alternate if and only if $uv \in E(G)$, and if some pair is represented incorrectly, we stop the algorithm and output "no". If only a single endpoint of $u \in V(G)$ appears in τ', no checking is done. This checking can be done trivially in time $\mathcal{O}(n^2)$.

Prime Graphs. A graph is called *prime* if it contains no split. If G is a prime graph, then it has at most two different representations $\mathcal{R}$ and $\hat{\mathcal{R}}$ [15] where one is the reversal of the other. We just need to test whether one of them extends $\mathcal{R}'$. We can construct one of these representations in quasilinear time [23].

Finding a Maximal Split Between A and B. If the graph G is not prime, then we can find a non-trivial split between A' and B' in linear time [15]. Using Lemma 1, we modify it into a maximal split between A and B such that $A' \subseteq A$ and $B' \subseteq B$ in linear time.

Algorithm 1 The $\mathcal{O}(n^3)$ algorithm for RepExt(CIRCLE).

Input: A circle graph G and a partial representation $\mathcal{R}'$ corresponding to a circular word τ'.
Output: ACCEPT if $\mathcal{R}'$ is extendible, REJECT otherwise.

If $\mathcal{R}'$ is incorrect **then** REJECT.
If G is a prime graph **then**
  Construct the unique representations τ and (its reverse) τ_R of G.
  If τ' is a subsequence of τ or τ_R **then** ACCEPT **else** REJECT.
Else (G is not a prime graph)
  Find a non-trivial split between A' and B'.
  Modify it into a maximal split between A and B such that $A' \subseteq A$ and $B' \subseteq B$.
  Case I: If the maximal split between A and B is non-trivial **then**
    Compute the equivalence relation $\sim$.
    Let $\tau' = \tau'_1 \cdots \tau'_k$ be the maximal subwords of extended classes Ψ.
    Case I.1: If some extended class corresponds to two maximal subwords in τ' **then**
      Compute a circular ordering $\Psi_1, \dots, \Psi_\ell$ compatible with τ'.
      Construct the partial representations $\mathcal{R}'_i$ of G_i.
      If all $\mathcal{R}'_1, \dots, \mathcal{R}'_\ell$ are extendible **then** ACCEPT **else** REJECT.
    Case I.2: Else (each extended class corresponds to one maximal subword in τ')
      Construct the partial representations $\mathcal{R}'_i$ and $\widetilde{\mathcal{R}}'_i$ of G_i.
      Proceed as in the subroutine of Algorithm 2.
  Case II: Else (the maximal split between A and B is trivial with $A = \{a\}$ and $\mathfrak{s}(A) = \emptyset$)
    Compute the connected components of $G \setminus a$.
    Case II.1: If both endpoints of a appear in τ' **then**
      Compute a linear ordering $C_1, \dots, C_\ell$ compatible with τ'.
      Construct the partial representations $\mathcal{R}'_i$ of G_i.
      If all $\mathcal{R}'_1, \dots, \mathcal{R}'_\ell$ are extendible **then** ACCEPT **else** REJECT.
    Case II.2: Else if single endpoint of a appears in τ' **then**
      Decompose the problem into two subproblems.
      One is solved using Case II.1, the other as in Case I.2.
      If both succeed **then** ACCEPT **else** REJECT.
    Case II.3: Else (no endpoint of a appears in τ')
      Case II.3a: If some component has two maximal subwords in τ' **then**
        Decompose the problem into three subproblems.
        Two are solved using Case II.2, the last one using Case II.1.
        If all succeed **then** ACCEPT **else** REJECT.
      Else (no component has two maximal subwords in τ')
        Proceed as in the subroutine of Algorithm 3.

4.1 Case I: A Non-trivial Maximal Split Between A and B.

We start by computing the equivalence relation $\sim$ which can be done in time $\mathcal{O}(n^2)$. Next, we want to find an ordering of its equivalence classes. For a class Φ of $\sim$, we define the *extended class* Ψ of $\sim$ as $\Phi \cup \mathfrak{s}(\Phi)$. If some extended class has no vertex pre-drawn, we may choose an arbitrary representation and place it in an arbitrary order, so we can ignore such classes for the rest of Case I. Let $\sim$ have ℓ equivalence classes, all of them appearing in τ'.

The circular word τ' is composed of k *maximal subwords* $\tau' = \tau'_1\tau'_2 \cdots \tau'_k$ such that each τ'_i contains only symbols of one extended class Ψ. According to Proposition 1, each extended class Ψ corresponds to at most two different maximal subwords. Also, if two extended classes Ψ and $\hat{\Psi}$ each correspond to two different maximal subwords, then occurrences of these subwords alternate in τ'. Otherwise we reject the input.

Case I.1: An extended class corresponds to two maximal subwords.
We denote this class by Ψ_1 and put this class as first in the ordering. By renumbering, we may assume that Ψ_1 corresponds to τ'_1 and τ'_t. Then one circular order of the classes can be determined by the following linear ordering $<$ starting with Ψ_1. Let Ψ_i and Ψ_j be two distinct classes. If Ψ_i corresponds to τ'_a and Ψ_j

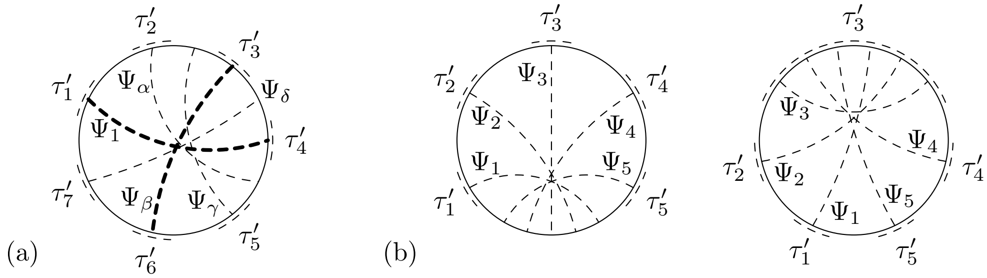

FIGURE 8. Each dashed line represents one extended class. (a) An example of Case I.1. We have $\tau' = \tau_1' \cdots \tau_7'$ and five extended classes Ψ_1 (corresponding to τ_1' and τ_4'), Ψ_α (to τ_2'), Ψ_β (to τ_3' and τ_6'), Ψ_γ (to τ_5'), and Ψ_δ (to τ_7'). We get that $\Psi_1 < \Psi_\alpha < \Psi_\beta$ and $\Psi_1 < \Psi_\gamma < \Psi_\beta < \Psi_\delta$, so one possible circular ordering is $\Psi_1, \Psi_\alpha, \Psi_\gamma, \Psi_\beta, \Psi_\delta$, and $\alpha = 2$, $\gamma = 3$, $\beta = 4$, and $\delta = 5$. By Lemma 12, $\mathcal{R}'$ is extendible if and only if $\mathcal{R}_1', \dots, \mathcal{R}_5'$ are extendible. (b) Examples of two possible extending representations in Case I.2. On the left, τ_i' is extended by τ_i in an extending representation $\mathcal{R}$, which is possible only when $\mathcal{R}_1', \dots, \mathcal{R}_5'$ are extendible. On the right, τ_3' is extended by both τ_3 and $\hat{\tau}_3$. By Lemma 13, $\mathcal{R}_1', \mathcal{R}_2', \mathcal{R}_4', \mathcal{R}_5'$ are extendible, but it is sufficient for $\widetilde{\mathcal{R}}_3'$ to be extendible.

corresponds to τ_b' such that either $a < b < t$ or $t < a < b$, we put $\Psi_i < \Psi_j$. We obtain the ordering of the classes as any linear extension of $<$. Since subwords of all extended classes with two subwords in τ' alternate, we get that $<$ is acyclic and a linear extension always exists. Figure 8(a) shows an example.

We have ordered the extended classes $\Psi_1, \dots, \Psi_\ell$ and the corresponding classes $\Phi_1, \dots, \Phi_\ell$. We construct each G_i with the vertices $\Psi_i \cup \{v_i\}$ as in Section 3.2, so v_i is adjacent to Φ_i and non-adjacent to $\mathfrak{s}(\Phi_i)$. The partial representation $\mathcal{R}_i'$ of G_i is either the word $v_i \tau_s' v_i$ (if Ψ_i corresponds to the single maximal subword τ_s' in τ') or the word $v_i \tau_s' v_i \tau_t'$ (if Ψ_i corresponds to two maximal subwords τ_s' and τ_t' in τ'). We test recursively, whether each representation $\mathcal{R}_i'$ of G_i is extendible to a representation of $\mathcal{R}_i$. If yes, we join $\mathcal{R}_1, \dots, \mathcal{R}_\ell$ as in Proposition 1. Otherwise, the algorithm outputs "no".

Lemma 12. In Case I.1, the representation $\mathcal{R}'$ is extendible if and only if the representations $\mathcal{R}_1', \dots, \mathcal{R}_\ell'$ of the graphs $G_1, \dots, G_\ell$ are extendible.

Proof. Suppose that $\mathcal{R}$ extends $\mathcal{R}'$. According to Proposition 1, the representations of $\Psi_1, \dots, \Psi_\ell$ are ordered along the circle, and so we obtain representations $\mathcal{R}_1, \dots, \mathcal{R}_\ell$ extending $\mathcal{R}_1', \dots, \mathcal{R}_\ell'$.

For the other implication, we just take $\mathcal{R}_1, \dots, \mathcal{R}_\ell$ and combine them to form $\mathcal{R}$ as in (1). This works since the ordering $<$ was constructed so that $\mathcal{R}$ extends $\mathcal{R}'$. ∎

Case I.2: No extended class corresponds to two maximal subwords.
We number the classes according to their appearance in τ', i.e., Ψ_i corresponds to the subword τ_i'. By Proposition 1, we know that in any representation $\mathcal{R}$ of G the class Ψ_i corresponds to two subwords τ_i and $\hat{\tau}_i$. The difficulty here arises from the potential for τ_i' to be a subsequence of $\tau_i \hat{\tau}_i$, but of neither τ_i, nor $\hat{\tau}_i$. Figure 8(b) shows two potential extending representations.

We solve this as follows. Instead of constructing just one partial representation $\mathcal{R}_i'$ of G_i corresponding to the circular word $\tau_i' v_i v_i$, we construct an additional partial representation $\widetilde{\mathcal{R}}_i'$ corresponding to the circular word $\tau_i' v_i$, i.e., v_i has only one endpoint pre-drawn. Figure 9 shows that $\widetilde{\mathcal{R}}_i'$ is less restrictive: if $\mathcal{R}_i'$ is extendible, then $\widetilde{\mathcal{R}}_i'$ is also extendible, but it might not be true the other way. For instance, every long chord in Φ_i alternates with v_i, so if some long chord has both endpoints pre-drawn in τ_i', $\mathcal{R}_i'$ is necessarily non-extendible, but $\widetilde{\mathcal{R}}_i'$ might be extendible.

The following lemma is the main trick of the algorithm and is essential to prove that it has cubic running time. It states that, if τ' is extendible, at most one class can be forced to use $\widetilde{\mathcal{R}}_i'$.

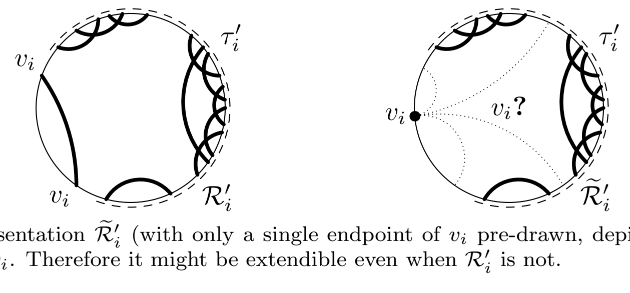

FIGURE 9. The partial representation $\widetilde{\mathcal{R}}_i'$ (with only a single endpoint of v_i pre-drawn, depicted by a dot) is less restrictive with respect to the position of v_i. Therefore it might be extendible even when $\mathcal{R}_i'$ is not.

Algorithm 2 The subroutine for Case I.2.

Test whether each of $\mathcal{R}'_2, \ldots, \mathcal{R}'_\ell$ is extendible.
If two of $\mathcal{R}'_2, \ldots, \mathcal{R}'_\ell$ are not extendible **then** REJECT.
If exactly one of $\mathcal{R}'_2, \ldots, \mathcal{R}'_\ell$, denoted by $\mathcal{R}'_i$, is not extendible **then**
    If $\widetilde{\mathcal{R}}'_i$ and $\mathcal{R}'_1$ are extendible **then** ACCEPT **else** REJECT.
Else (all of $\mathcal{R}'_2, \ldots, \mathcal{R}'_\ell$ are extendible)
    If $\widetilde{\mathcal{R}}'_1$ is extendible **then** ACCEPT **else** REJECT.

Lemma 13. In Case I.2, the representation $\mathcal{R}'$ is extendible if and only if $\widetilde{\mathcal{R}}'_i$ is extendible for some i and $\mathcal{R}'_j$ is extendible for all $j \neq i$.

Proof. When $\mathcal{R}_j$ corresponding to a word $v_j \tau_j v_j \hat{\tau}_j$ is an extension of $\mathcal{R}'_j$ for $j \neq i$, then τ'_j is a subsequence of, say, τ_j. On the other hand, when $\mathcal{R}_i$ corresponding to a word $v_i \tau_i v_i \hat{\tau}_i$ is an extension of $\widetilde{\mathcal{R}}'_i$, then τ'_i is a subsequence of $\tau_i \hat{\tau}_i$, but might not be of τ_i or $\hat{\tau}_i$. We use the circular ordering $\Psi_{i+1}, \ldots, \Psi_\ell, \Psi_1, \ldots, \Psi_i$ of the classes and we construct the representation $\mathcal{R}$ as in (1):

$$\boldsymbol{\tau_{i+1} \cdots \tau_\ell \tau_1 \cdots \tau_i} \hat{\tau}_{i+1} \cdots \hat{\tau}_\ell \hat{\tau}_1 \cdots \hat{\tau}_{i-1} \boldsymbol{\hat{\tau}_i},$$

where all pre-drawn endpoints of τ' appear in those words written in bold. It is easy to see that $\mathcal{R}$ extends $\mathcal{R}'$ since τ' has no pre-drawn endpoints in $\hat{\tau}_{i+1} \cdots \hat{\tau}_\ell \hat{\tau}_1 \cdots \hat{\tau}_{i-1}$.

For the other implication, suppose that $\mathcal{R}$ extends $\mathcal{R}'$. For contradiction, suppose that two distinct partial representations $\mathcal{R}'_i$ and $\mathcal{R}'_j$ are not extendible. According to Proposition 1, the representation $\mathcal{R}$ gives a representation $\mathcal{R}_i$ corresponding to $v_i \tau_i v_i \hat{\tau}_i$ of G_i and $\mathcal{R}_j$ corresponding to $v_j \tau_j v_j \hat{\tau}_j$ of G_j. Since $\mathcal{R}'_i$ and $\mathcal{R}'_j$ are non-extendible, we have that τ'_i is neither a subsequence of τ_i, nor $\hat{\tau}_i$, and similarly τ'_j is neither of τ_j, nor $\hat{\tau}_j$. Therefore, either $\tau_i \tau_j \hat{\tau}_i \hat{\tau}_j$, or $\tau_j \tau_i \hat{\tau}_j \hat{\tau}_i$ is a subsequence of τ, and we get that two maximal subwords in τ' correspond to both Ψ_i and Ψ_j which is a contradiction. ∎

Let $n = |V(G)|$ and let Ψ_1 be the largest class, so $|\Psi_i| \leq n/2$ for $i > 1$. If we want to recursively test for each Ψ_i whether both $\mathcal{R}'_i$ and $\widetilde{\mathcal{R}}'_i$ are extendible, the running time might be exponential since we might have $|\Psi_1| \approx n$. Fortunately, using Lemma 13, it is sufficient to test only one of $\mathcal{R}'_1$ and $\widetilde{\mathcal{R}}'_1$. We recursively test whether $\mathcal{R}'_2, \ldots, \mathcal{R}'_\ell$ are extendible; see the pseudocode of Algorithm 2:

- *Two or more of $\mathcal{R}'_2, \ldots, \mathcal{R}'_\ell$ are not extendible.* By Lemma 13, $\mathcal{R}'$ is non-extendible, the algorithm stops and outputs "no".
- *Exactly one of $\mathcal{R}'_2, \ldots, \mathcal{R}'_\ell$ is not extendible.* Let $\mathcal{R}'_i$ be the non-extendible representation. We test whether $\widetilde{\mathcal{R}}'_i$ and $\mathcal{R}'_1$ are extendible. If at least one is non-extendible, the algorithm stops and outputs "no". If both are extendible, we similarly join in $\mathcal{R}$ the representations $\mathcal{R}_1, \ldots, \mathcal{R}_\ell$ according to (1) as described in the proof of Lemma 13.
- *All representations $\mathcal{R}'_2, \ldots, \mathcal{R}'_\ell$ are extendible.* We have representations $\mathcal{R}_2, \ldots, \mathcal{R}_\ell$ where $\mathcal{R}_i$ extends $\mathcal{R}'_i$. We test whether the partial representation $\widetilde{\mathcal{R}}'_1$ is extendible. If not, the algorithm stops and outputs "no". If it extends, we get a representation $\mathcal{R}_1$ of $\widetilde{G}_1$. We construct the representation $\mathcal{R}$ using (1) as described in the proof of Lemma 13.

Lemma 14. In Case I.2, the representation $\mathcal{R}'$ is extendible if and only if the algorithm constructs it.

Proof. We know that $\widetilde{\mathcal{R}}'_i$ is extendible when $\mathcal{R}'_i$ is extendible. Lemma 13 states that $\mathcal{R}'$ is extendible if and only if at most one of $\mathcal{R}'_i$ is non-extendible while $\widetilde{\mathcal{R}}'_i$ is extendible. The algorithm tests this in Case I.2, while postponing Ψ_1 until it knows which of $\mathcal{R}'_1$ and $\widetilde{\mathcal{R}}'_1$ needs to be tested. ∎

4.2 Case II: A Trivial Maximal Split Between A and B

Let $A = \{a\}$ and $\mathfrak{s}(A) = \emptyset$. In Section 3.3 we characterized all possible representations $\mathcal{R}$ in terms of representations of connected components C of $G \setminus a$. We just need to test whether one of them is compatible with the partial representation $\mathcal{R}'$ corresponding to the circular word τ'. Similarly as in Section 4.1, we may

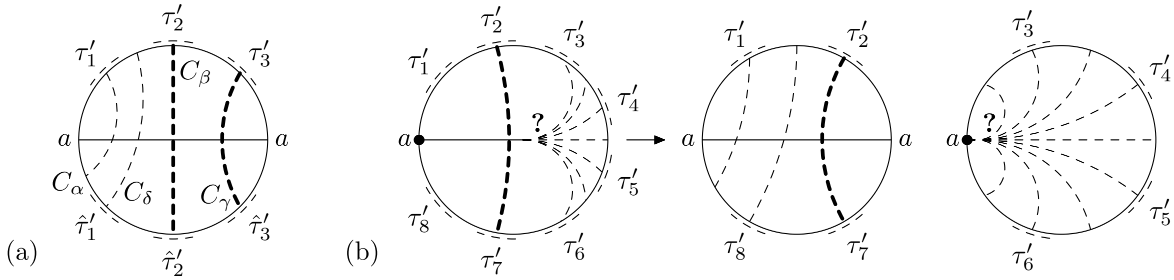

FIGURE 10. Each dashed line represents one connected component. (a) An example of Case II.1. We have $\tau' = a\tau'_1\tau'_2\tau'_3a\hat\tau'_3\hat\tau'_2\hat\tau'_1$ and four connected components C_α (corresponding to τ'_1), Ψ_β (to τ'_2 and $\hat\tau'_2$), Ψ_γ (to τ'_3 and $\hat\tau'_3$), and Ψ_δ (to $\hat\tau'_1$). We get that $C_\alpha < C_\beta < C_\gamma$ and $C_\delta < C_\beta < C_\gamma$, so one possible linear ordering is $C_\alpha, C_\delta, C_\beta, C_\gamma$, and $\alpha = 1$, $\delta = 2$, $\beta = 3$, and $\gamma = 4$. By Lemma 15, $\mathcal{R}'$ is extendible if and only if $\mathcal{R}'_1, \dots, \mathcal{R}'_4$ are extendible. (b) An example of Case II.2. On the left, we have a connected component corresponding to two maximal subwords τ'_2 and τ'_7. Therefore, every extending representation has the subsequence $a\tau'_2a\tau'_7$. We divide the problem into two depicted subproblems, one of Case II.1, the other of Case II.2 with each component C_i corresponding to exactly one maximal subword τ'_i.

assume that every connected component C has at least one endpoint in τ'; otherwise, we can deal with it trivially.

Case II.1: Both endpoints of a appear in τ'. The circular word τ' is composed of k and k' *maximal subwords* $\tau' = a\tau'_1\tau'_2\cdots\tau'_k a\hat\tau'_{k'}\hat\tau'_{k'-1}\cdots\hat\tau'_1$ such that each τ'_i contains only symbols of one connected component C and similarly for each $\hat\tau'_i$. According to Proposition 2, each connected component C corresponds to at most two different maximal subwords. If a connected component C corresponds to two subwords τ'_i and $\hat\tau'_j$, then $a\tau'_ia\hat\tau'_j$ is a subsequence of τ'. Also, if two components C and $\hat C$ each correspond to two different maximal subwords, then occurrences of these subwords do not alternate in τ'. Otherwise we reject the input.

Next, we find a linear ordering of ℓ connected components as follows. We order $C < C'$ if C corresponds to a subword τ'_s and C' to a subword τ'_t for $s < t$, or C to $\hat\tau'_s$ and C' to $\hat\tau'_t$ for $s < t$. We obtain a linear ordering $C_1, \dots, C_\ell$ as any linear extension. Since subwords of all connected components with two subwords in τ' do not alternate, we get that $<$ is acyclic and a linear extension always exists. Suppose that we renumber the maximal subwords of τ' in such a way that C_i corresponds to τ'_i and $\hat\tau'_i$ (one of them possibly empty). Let G_i be the subgraph of G induced by $V(C_i) \cup \{a\}$. Let $\mathcal{R}'_i$ be the partial representation of G_i corresponding to the circular word $a\tau'_ia\hat\tau'_i$, so $\tau' = a\tau'_1\cdots\tau'_\ell a\hat\tau'_1\cdots\hat\tau'_\ell$. Figure 10(a) shows an example.

Lemma 15. In Case II.1, the representation $\mathcal{R}'$ is extendible if and only if the representations $\mathcal{R}'_1, \dots, \mathcal{R}'_\ell$ of the graphs $G_1, \dots, G_\ell$ are extendible.

Proof. Suppose that $\mathcal{R}$ extends $\mathcal{R}'$. According to Proposition 2, the representations of $C_1, \dots, C_\ell$ are ordered along the circle, and so we obtain representations $\mathcal{R}_1, \dots, \mathcal{R}_\ell$ extending $\mathcal{R}'_1, \dots, \mathcal{R}'_\ell$.

For the other implication, we just take $\mathcal{R}_1, \dots, \mathcal{R}_\ell$ and combine them to form $\mathcal{R}$ as in (2). This works since the ordering $<$ was constructed so that $\mathcal{R}$ extends $\mathcal{R}'$. ∎

Case II.2: A single endpoint of a appears in τ'. The circular word τ' is composed of k *maximal subwords* $\tau' = a\tau'_1\tau'_2\cdots\tau'_k$ such that each τ'_i contains only symbols of one connected component C. According to Proposition 2, each connected component C corresponds to at most two different maximal subwords. Also, if two components C and $\hat C$ each correspond to two different maximal subwords, then occurrences of these subwords do not alternate in τ'. Otherwise we reject the input.

Suppose there is a component C corresponding to two maximal subwords τ'_s and τ'_t for $s < t$. Further let C be such a component that maximizes the value s. In every extending representation, we have the subsequence $a\tau'_sa\tau'_t$, so we can assume that the second endpoint of a is pre-drawn in between τ'_s and τ'_t. We divide testing whether $\mathcal{R}'$ is extendible into two subproblems. We deal with the connected components of the circular word $a\tau'_1\tau'_2\cdots\tau'_sa\tau'_t\tau'_{t+1}\cdots\tau'_k$ exactly as in Case II.1. It remains to decide whether $a\tau'_{s+1}\cdots\tau'_{t-1}$ is extendible where each connected component corresponds to precisely one maximal subword (note: when no such component C exists, we have precisely this situation). Figure 10(b) shows an example.

Suppose that we rename $\tau' = a\tau'_1\cdots\tau'_\ell$ such that τ'_i corresponds to the connected component C_i. Similarly to Case I.2, the difficulty comes from the fact that some τ'_i might be a subsequence of $\tau_i\hat\tau_i$ of (2) in an extending representation, but not of τ_i or $\hat\tau_i$. We consider two partial representations for each G_i: the partial representation $\mathcal{R}'_i$ corresponding to $a\tau'_ia$ and $\widetilde{\mathcal{R}}'_i$ corresponding to $a\tau'_i$. Again, if $\mathcal{R}'_i$ is extendible, then $\widetilde{\mathcal{R}}'_i$ is also extendible.

Lemma 16. In Case II.2 with no connected component correspond to two maximal subwords of τ', the representation $\mathcal{R}'$ is extendible if and only if $\widetilde{\mathcal{R}}'_i$ is extendible for some i and $\mathcal{R}'_j$ is extendible for all $j \neq i$.

Proof. When $\mathcal{R}_j$ corresponding to a word $a\tau_j a\hat{\tau}_j$ is an extension of $\mathcal{R}'_j$ for $j \neq i$, then τ'_j is a subsequence of, say, τ_j for $j < i$ and of $\hat{\tau}_j$ for $j > i$. On the other hand, when $\mathcal{R}_i$ corresponding to a word $a\tau_i a\hat{\tau}_i$ is an extension of $\widetilde{\mathcal{R}}'_i$, then τ'_i is a subsequence of $\tau_i\hat{\tau}_i$, but might not be of τ_i or $\hat{\tau}_i$. We use the linear ordering $C_1, \dots, C_{i-1}, C_\ell, C_{\ell-1}, \dots, C_i$ of the connected components and we construct the representation $\mathcal{R}$ as in (2):

$$\boldsymbol{a\tau_1\cdots\tau_{i-1}}\tau_\ell\cdots\tau_{i+1}\boldsymbol{\tau_i a\hat{\tau}_i\cdots\hat{\tau}_\ell}\hat{\tau}_{i-1}\cdots\hat{\tau}_1.$$

where all pre-drawn endpoints of τ' appear in those words written in bold. It is easy to see that $\mathcal{R}$ extends $\mathcal{R}'$ since there are no pre-drawn endpoints in $\tau_\ell\cdots\tau_{i+1}$ and in $\hat{\tau}_{i-1}\cdots\hat{\tau}_1$.

For the other implication, suppose that $\mathcal{R}$ corresponding to τ extends $\mathcal{R}'$, and we add into τ' the position of the other endpoint of a. It splits at most one maximal word τ'_i, so $a\tau'_j a$ is a subsequence of τ and $\mathcal{R}'_j$ is extendible. Since $a\tau'_i$ is a subsequence of τ, we get that $\widetilde{\mathcal{R}}'_i$ is extendible. ∎

The rest of this case proceeds exactly as Case I.2.

Lemma 17. In Case II.2, the representation $\mathcal{R}'$ is extendible if and only if the algorithm constructs it.

Proof. The proof is similar to Lemma 14. ∎

Case II.3: No endpoint of $\boldsymbol{a}$ appears in $\boldsymbol{\tau'}$. As in Case II.2, the circular word τ' is composed of k *maximal subwords* $\tau' = \tau'_1\tau'_2\cdots\tau'_k$. If two components C and $\hat{C}$ each correspond to two different maximal subwords, then occurrences of these subwords do not alternate in τ'. Otherwise we reject the input. Also, if some connected component C corresponds to two subwords τ'_i and τ'_j, then $a\tau'_i a\tau'_j$ is a subsequence of every extending representation. Therefore, existence of such a component restricts the possible positions of endpoints of a, so we divide this case into two subcases.

Case II.3a: Some component has two maximal subwords in $\boldsymbol{\tau'}$. By a suitable renaming of the subwords, let C be the connected component corresponding to τ'_p and τ'_q such that $p < q$, p is minimal, and $\tau'_{q+1}, \dots, \tau'_\ell, \tau'_1, \dots, \tau'_{p-1}$ correspond to connected components having only one maximal subword in τ'. Similarly, let C' be the connected component corresponding to τ'_s and τ'_t such that $s < t$ and all $\tau'_{s+1}, \dots, \tau'_{t-1}$ correspond to connected components having only one maximal subword in τ', and possibly $C = C'$. If $\mathcal{R}'$ is extendible, we get that every connected component corresponding to two maximal subwords τ'_x and τ'_y has $p \le x \le s < t \le y \le q$; otherwise we reject the input. Figure 11 shows an example.

It follows that every extending representation has $a\tau'_p\tau'_s a\tau'_t\tau'_q$ as a subsequence. Similarly as Case II.2, we can divide testing whether $\mathcal{R}'$ is extendible into three subproblems:

- Testing using Case II.2 whether the partial representation $\tau'_{q+1}\cdots\tau'_\ell\tau'_1\cdots\tau'_{p-1}a$ is extendible.
- Testing using Case II.1 whether the partial representation $a\tau'_p\cdots\tau'_s a\tau'_t\cdots\tau'_q$ is extendible.
- Testing using Case II.2 whether the partial representation $a\tau'_{s+1}\cdots\tau'_{t-1}$ is extendible.

Lemma 18. In Case II.3a, the representation $\mathcal{R}'$ is extendible if and only if the algorithm constructs it.

Proof. This is implied by Lemmas 15 and 17. ∎

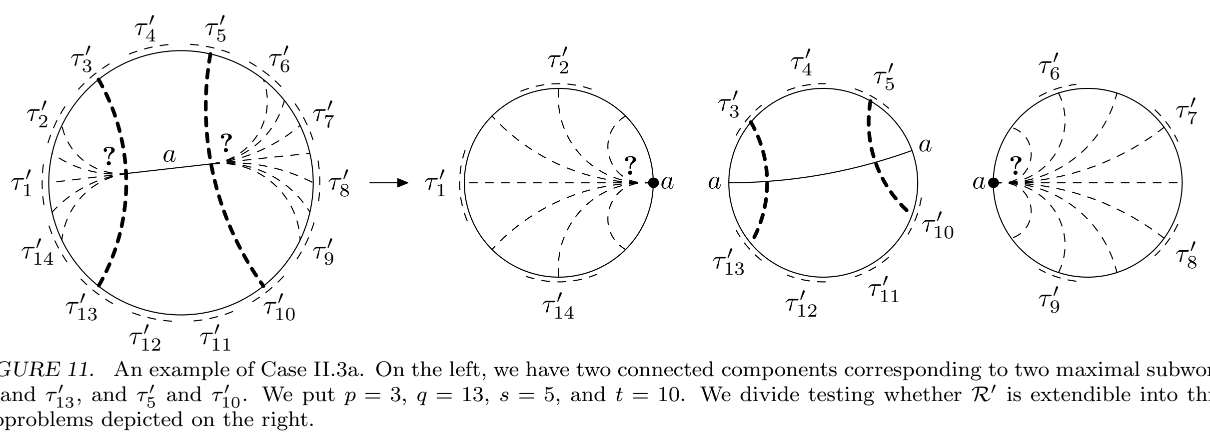

FIGURE 11. An example of Case II.3a. On the left, we have two connected components corresponding to two maximal subwords τ'_3 and τ'_{13}, and τ'_5 and τ'_{10}. We put $p = 3$, $q = 13$, $s = 5$, and $t = 10$. We divide testing whether $\mathcal{R}'$ is extendible into three subproblems depicted on the right.

Algorithm 3 The subroutine for Case II.3b.

Test whether each of $\mathcal{R}'_2, \dots, \mathcal{R}'_\ell$ is extendible.
If three of $\mathcal{R}'_2, \dots, \mathcal{R}'_\ell$ are not extendible **then** REJECT.
If exactly two of $\mathcal{R}'_2, \dots, \mathcal{R}'_\ell$, denoted $\mathcal{R}'_i$ and $\mathcal{R}'_j$, are not extendible **then**
    If $\widetilde{\mathcal{R}}'_i$, $\widetilde{\mathcal{R}}'_j$ and $\mathcal{R}'_1$ are extendible **then** ACCEPT **else** REJECT.
If exactly one of $\mathcal{R}'_2, \dots, \mathcal{R}'_\ell$, denoted $\mathcal{R}'_i$, is not extendible **then**
    If $\widetilde{\mathcal{R}}'_i$ and $\widetilde{\mathcal{R}}'_1$ are extendible **then** ACCEPT **else** REJECT.
Else (all of $\mathcal{R}'_2, \dots, \mathcal{R}'_\ell$ are extendible)
    If $\widetilde{\mathcal{R}}'_1$ is extendible **then** ACCEPT **else** REJECT.

Case II.3b: No component has two maximal subwords in $\boldsymbol{\tau'}$. Let τ'_i correspond to the connected component C_i, and define $\mathcal{R}'_i$ and $\widetilde{\mathcal{R}}'_i$ exactly as in Case II.2. Similarly to Case II.2, the difficulty comes from the fact that some τ'_i might correspond to both τ_i and $\hat{\tau}_i$ of (2) in an extending representation. Since we are placing two endpoints of a, we might have two such components C_i and C_j.

Lemma 19. The representation $\mathcal{R}'$ is extendible if and only if $\widetilde{\mathcal{R}}'_i$ and $\widetilde{\mathcal{R}}'_j$ are extendible for some i and j, and $\mathcal{R}'_k$ is extendible for all $k \neq i, j$.

Proof. Let $i < j$. When $\mathcal{R}_k$ corresponding to a word $a\tau_k a\hat{\tau}_k$ is an extension of $\mathcal{R}'_k$ for $k \neq i, j$, then τ'_k is a subsequence of, say, τ_k for $k < j$ and of $\hat{\tau}_k$ for $k > j$. On the other hand, when $\mathcal{R}_i$ corresponding to a word $a\tau_i a\hat{\tau}_i$ is an extension of $\widetilde{\mathcal{R}}'_i$, then τ'_i is a subsequence of $\tau_i\hat{\tau}_i$, but might not be of τ_i or $\hat{\tau}_i$, and similarly for $\mathcal{R}_j$. We use the linear ordering $C_i, C_{i-1}, \dots, C_1, C_{i+1}, \dots, C_{j-1}, C_\ell, C_{\ell-1}, \dots, C_{j+1}, C_j$ of the connected components and we construct the representation $\mathcal{R}$ as in (2):

$$a\boldsymbol{\tau_i}\hat{\tau}_{i-1}\cdots\hat{\tau}_1\boldsymbol{\tau_{i+1}\cdots\tau_{j-1}}\tau_\ell\cdots\tau_{j+1}\boldsymbol{\tau_j}a\boldsymbol{\hat{\tau}_j\cdots\hat{\tau}_\ell}\hat{\tau}_{j-1}\cdots\hat{\tau}_{i+1}\boldsymbol{\tau_1\cdots\tau_{i-1}\hat{\tau}_i},$$

where all pre-drawn endpoints of τ' appear in those words written in bold. It is easy to see that $\mathcal{R}$ extends $\mathcal{R}'$ since there are no pre-drawn endpoints in $\hat{\tau}_{i-1}\cdots\hat{\tau}_1$, in $\tau_\ell\cdots\tau_{j+1}$, and in $\hat{\tau}_{j-1}\cdots\hat{\tau}_{i+1}$.

For the other implication, suppose that $\mathcal{R}$ corresponding to τ extends $\mathcal{R}'$, and we add into τ' the positions of the endpoints of a. It is not possible that both endpoints split the same maximal word τ'_i, otherwise the remaining components C_k would alternate with C_i. It is additionally not possible that two maximal words are split by the same endpoint. So at most two maximal words τ'_i and τ'_j are split by the endpoints of a. Therefore, for every $k \neq i, j$, we have $a\tau'_k a$ as a subsequence of τ, so $\mathcal{R}'_k$ is extendible. Since $a\tau'_i$ and $a\tau'_j$ are subsequences of τ, we get that $\widetilde{\mathcal{R}}'_i$ and $\widetilde{\mathcal{R}}'_j$ are also extendible. ∎

Let $n = |V(G)|$ and let C_1 be the largest component, so $|V(C_i)| \leq n/2$ for $i > 1$. The algorithm works similarly to Case I.2; see Algorithm 3 for a pseudocode. So we test the extendibility of only one of $\mathcal{R}'_1$ and $\widetilde{\mathcal{R}}'_1$ while testing both types of representations for at most two other graphs G_i and G_j.

Lemma 20. In Case II.3b, the representation $\mathcal{R}'$ is extendible if and only if the algorithm constructs it.

Proof. We use Lemma 19 similarly as in the proof of Lemmas 14 and 17. ∎

4.3 Analysis of the Algorithm

By using the established results, we show that the partial representation extension problem of circle graphs can be solved in cubic time.

Lemma 21. The described algorithm correctly decides whether the partial representation $\mathcal{R}'$ of G is extendible.

Proof. If the input graph G is prime, we just test both representations whether they extend τ'. If the input graph G contains a non-trivial split, we modify it into a maximal split between A and B using Lemma 1. Next, we proceed by Case I or Case II, depending whether the maximal split is trivial or not. For Case I, the algorithm is correct by Lemmas 12 and 14. For Case II, the algorithm is correct by Lemmas 15, 17, 18, and 20. ∎

Lemma 22. The running time of the algorithm is $\mathcal{O}(n^3)$ where n is the number of vertices.

Proof. Let $T(n)$ denote the time complexity of the algorithm for at most n vertices in the worst case. We want to show that $T(n) = \mathcal{O}(n^3)$.

As described, we can test whether the graph G is prime and construct a unique representation τ in quasi-linear time using [23], but for the purpose of our analysis $\mathcal{O}(n^2)$ is sufficient. Since each symbol appears twice in τ, we can easily test in linear time whether τ' is a subsequence of τ or its reversal. If G is not prime, then we can find a non-trivial split between A' and B' using [15] and modify it using Lemma 1 into a maximal split between A and B such that $A' \subseteq A$ and $B' \subseteq B$. Both can be achieved in linear time.

Case I. We compute the $\sim$ relation in time $\mathcal{O}(n^2)$.

- In Case I.1, we divide the problem into ℓ smaller disjoint subproblems of total size n, each of size $n_i + 1$ solvable by induction hypothesis in time $\mathcal{O}(n_i^3)$, so the total running time is $\mathcal{O}(n^3)$.
- In Case I.2, we test both representations $\mathcal{R}'_i$ and $\widetilde{\mathcal{R}}'_i$ for at most one extended class of size $|\Psi_i| \le \frac{n}{2}$, while we test exactly one of these representations of all remaining extended classes. We get the following recursion:

$$T(n) \le T(n/2+1) + \sum_j T(|\Psi_j|+1) + \mathcal{O}(n^2) \le T(n/2+1) + \mathcal{O}(n^3).$$

  By the Master Theorem, we get that $T(n) \le \mathcal{O}(n^3)$. Since the depth of the recursion is at most linear, each level of the recursion adds to at most $\mathcal{O}(n^2)$ and we get $\mathcal{O}(n^3)$ in total over all levels.

Case II. We find connected components of $G \setminus a$ in linear time.

- In Case II.1, the analysis is similar as in Case I.1.
- In Case II.2, we divide the input into two disjoint subproblems, one is solved as in Case II.1, the other as in Case I.2. Therefore, the total running time is $\mathcal{O}(n^3)$.
- In Case II.3a, we divide the input into three disjoint subproblems solved using Case II.1 and Case II.2, so the total running time is $\mathcal{O}(n^3)$.
- In Case II.3b, we test both representations $\mathcal{R}'_i$ and $\widetilde{\mathcal{R}}'_i$ for at most two extended class of size $|\Psi_i| \le \frac{n}{2}$, while we test exactly one of these representations of all remaining extended classes. We get the following recursion:

$$T(n) \le 2T(n/2+1) + \sum_j T(|\Psi_j|+1) + \mathcal{O}(n^2) \le 2T(n/2+1) + \mathcal{O}(n^3).$$

  By the Master Theorem, we again get that $T(n) \le \mathcal{O}(n^3)$.

Therefore, the total running time is $\mathcal{O}(n^3)$. ∎

The proof of the main result in this paper now follows easily.

Proof of Theorem 1. The result is implied by Lemma 21 and Lemma 22. ∎

5 SIMULTANEOUS REPRESENTATIONS OF CIRCLE GRAPHS

In this section, we give two results concerning the simultaneous representation problem for circle graphs: We show that this problem is NP-complete and FPT in the size of the common intersection. Formally, we deal with the following decision problem:

Problem:	Simultaneous Representation for Circle Graphs – SIM(CIRCLE)
Input:	Graphs $G_1, \dots, G_k$ such that $G_i \cap G_j = I$ for all $i \neq j$.
Output:	Do there exist representations $\mathcal{R}_1, \dots, \mathcal{R}_k$ of $G_1, \dots, G_k$ which use the same representation of the vertices of I?

Proof of Theorem 2. To show that SIM(CIRCLE) is NP-complete, we reduce it from the *total ordering problem*:

Problem: The total ordering problem - TotalOrdering
Input: A finite set S and a finite set T of triples from S.
Output: Does there exist a total ordering $<$ of S such that for all $(x, y, z) \in T$ either $x < y < z$, or $z < y < x$?

Opatrny [34] proved this problem is NP-complete.

Given an instance (S, T) of TotalOrdering and let $s = |S|$ and $t = |T|$. We construct a set of $t+1$ graphs $G_0, G_1, \dots, G_t$ as follows, so the number k from Sim(CIRCLE) is equal $t+1$. The intersection of $G_0, G_1, \dots, G_t$ is an independent set $I = S \cup \{w\}$ where w is a special vertex. The graph G_0 consists of a clique K_{s+1}, and to each vertex of this clique we attach exactly one vertex of I as a leaf. The graph G_i corresponds to the i-th constraint $(x_i, y_i, z_i) \in T$. In addition to I, each G_i contains two vertices u_i and v_i of degree three, such that u_i is adjacent to v_i, x_i and z_i, and v_i is further adjacent to y_i and the special vertex w. See Fig. 12 for an example of this construction.

The clique in G_0 defines a split where each class of $\sim$ is a singleton. According to Proposition 1, every representation $\mathcal{R}_0$ of G_0 places the elements of I in some circular ordering $wws_1s_1s_2s_2 \cdots s_ss_s$ which corresponds to the total ordering $s_1 < s_2 < \cdots < s_s$. Now the representations $\mathcal{R}_1, \dots, \mathcal{R}_t$ of $G_1, \dots, G_t$ can be constructed if and only if all the total ordering constraints are satisfied. This implies that there exists a solution $\mathcal{R}_0, \dots, \mathcal{R}_t$ of $G_0, \dots, G_t$ if and only if the instance (S, T) of TotalOrdering is solvable. ∎

Further, we show that the problem is FPT in size of the common subgraph I.

Proof of Corollary. We just consider all possible representations of the common subgraph I which are all words of length $2|V(I)|$. Each word gives some partial representation $\mathcal{R}'$. We just solve k instance of RepExt(CIRCLE) for each G_i and the partial representation $\mathcal{R}'$ of I, which can be done in polynomial time according to Theorem 1. ∎

6 CONCLUSIONS

The structural results described in Section 3, namely Propositions 1 and 2, are the main new tools developed in this paper. Using it, one can easily work with the structure of all representations which is a key component of the algorithm of Section 4 that solves the partial representation extension problem for circle graphs. The algorithm works with the recursive structure of all representations and matches the partial representation on it. Proposition 1 also seems to be useful in attacking the following open problems:

Question 1. What is the complexity of Sim(CIRCLE) for a fixed number k of graphs? In particular, what is it for $k = 2$?

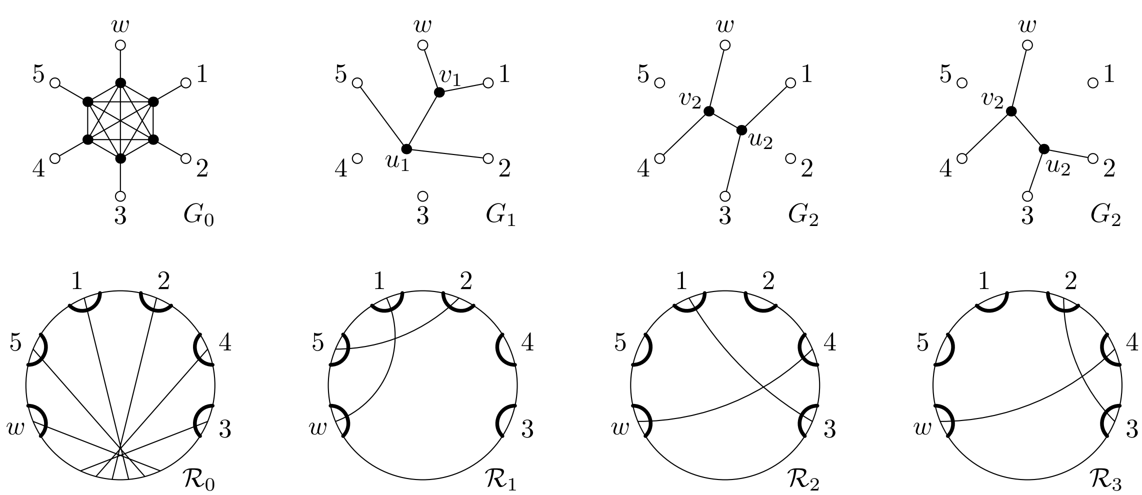

FIGURE 12. Let $S = \{1, 2, 3, 4, 5\}$ and T consisting of three triples $(5, 1, 2)$, $(1, 4, 3)$ and $(2, 4, 3)$ be the instance of TotalOrdering. We construct graphs $G_0, \dots, G_3$ depicted in the top, with the common vertices I depicted in white. Possible simultaneous representations are depicted in the bottom, giving the total ordering $5 < 1 < 2 < 4 < 3$.

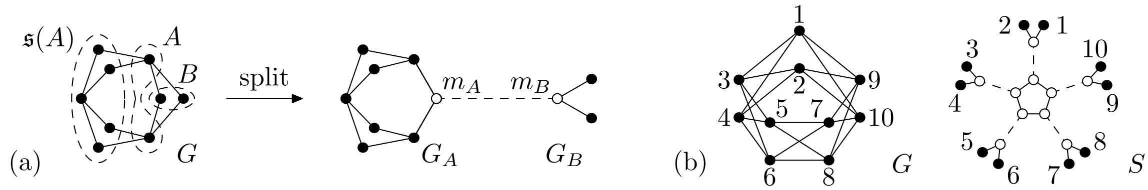

FIGURE 13. (a) An example of a split of the graph G. The marker vertices are depicted in white. The tree edge is depicted by a dashed line. (b) The split tree S of the graph G.

Recall that in the bounded representation problem, we give for some chords two circular arcs and we want to construct a representation which places endpoints into these circular arcs.

Question 2. What is the complexity of the bounded representation problem for circle graphs? This question is also open for interval graphs and proper interval graphs.

Permutation Graphs. Permutation graphs are intersection graphs of segments between two parallel lines. So every permutation representation of G consists of two words τ and $\hat{\tau}$, each containing each vertex $V(G)$ exactly once, and $uv \in E(G)$ if and only if their order in τ and $\hat{\tau}$ differs. We denote the class by PERM.

Let $\hat{\tau}_R$ be the reversal of $\hat{\tau}$. Since $\hat{\tau}$ is a circle representation of G, it follows that every permutation graph is a circle graph. More strongly, a graph G is a permutation graph if and only if $\widetilde{G}$ constructed from G by adding a universal vertex u is a circle graph, since $u\tau u\hat{\tau}_R$ is a circle representation of $\widetilde{G}$.

The partial representation problem for permutation graphs is studied in [25] and solved in time $\mathcal{O}(n^3)$. The following results gives an alternative algorithm running in time $\mathcal{O}(n^3)$ as well.

Proposition 3. The problem $\textsc{RepExt}(\mathsf{PERM})$ reduces in time $\mathcal{O}(n+m)$ to $\textsc{RepExt}(\mathsf{CIRCLE})$.

Proof. Let G be a permutation graph with a partial representation $\mathcal{R}'$ corresponding to two words τ' and $\hat{\tau}'$. The problem $\textsc{RepExt}(\mathsf{PERM})$ asks whether there exists words τ and $\hat{\tau}$ representing $\mathcal{R}$ such that τ' and $\hat{\tau}'$ are subsequences of τ and $\hat{\tau}$, respectively. The reduction constructs the circle graph $\widetilde{G}$ by adding a universal vertex u to G and the partial representation $\widetilde{\mathcal{R}}'$ given by the circular word $u\tau'u\hat{\tau}'_R$. The reduction clearly works in linear time. It is correct since the partial representation $\mathcal{R}'$ of G is extendible if and only if $\widetilde{\mathcal{R}}'$ of $\widetilde{G}$ is extendible. ∎

Minimal Split Decomposition and Split Trees. A split decomposition of G works as follows. Consider a split between A and B. We replace G by the graphs G_A and G_B defined in Section 3.2. Then we apply the decomposition recursively on G_A and G_B, and we stop on prime graphs containing no splits. We note that by different orders of splits, different decompositions of G may be constructed. A split decomposition can be computed in linear time [15].

A split decomposition is called *minimal* if it is constructed by the least number of splits. Suppose that we also stop on *degenerate graphs* which are complete graphs K_n and stars $S_n = K_{1,n}$. Cunningham [14, Theorem 3] proved that the minimal split decomposition of a connected graph stopping on prime and degenerate graphs is unique.

The unique *split tree* S representing a graph G encodes the minimal split decomposition [23]. A split tree is a graph with two types of vertices (normal and marker vertices) and two types of edges (normal and tree edges). We initially put $S = G$ and modify it according to the minimal split decomposition. If the minimal decomposition contains a split between A and B in G, then we replace G in S by the graphs G_A and G_B, and connect the marker vertices m_A and m_B by a *tree edge* (see Fig. 13a). We repeat this recursively on G_A and G_B; see Fig. 13b. Each prime and degenerate graph is a *node* of the split tree. A node that is incident with exactly one tree edge is called a *leaf node*.

The minimal split decompositions and the split trees can be computed in quasi-linear time [23]. Similarly as in Propositions 1 and 2, it should be possible to derive every circle representation of a connected graph G from the split tree S, but the precise statement is unclear. It is a natural question whether split trees can be used to solve the partial representation extension problem:

Question 3. Is it possible to use split trees S to solve $\textsc{RepExt}(\mathsf{CIRCLE})$? Can it be done faster that in time $\mathcal{O}(n^3)$?

ACKNOWLEDGMENTS

We want to thank to an anonymous reviewer for pointing out than the results described in Section 3 work only for maximal splits.

The second author is supported by the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme (FP7/2007-2013) under REA grant agreement no [291734].

The third author is supported by CE-ITI (P202/12/G061 of GAČR) and Charles University as GAUK 1334217.

References

[1] P. Angelini, G. D. Battista, F. Frati, V. Jelínek, J. Kratochvíl, M. Patrignani, and I. Rutter. Testing planarity of partially embedded graphs. *ACM Trans. Algorithms*, 11(4):32:1–32:42, 2015.

[2] M. Balko, P. Klavík, and Y. Otachi. Bounded representations of interval and proper interval graphs. In *ISAAC*, volume 8283 of *Lecture Notes in Computer Science*, pages 535–546, 2013.

[3] J. Bang-Jensen, J. Huang, and X. Zhu. Completing orientations of partially oriented graphs. *CoRR*, abs/1509.01301, 2015.

[4] T. Bläsius, S. G. Kobourov, and I. Rutter. Simultaneous embedding of planar graphs. In Roberto Tamassia, editor, *Handbook of Graph Drawing and Visualization*, pages 349–383. CRC Press, 2013.

[5] T. Bläsius and I. Rutter. Simultaneous PQ-ordering with applications to constrained embedding problems. *ACM Trans. Algorithms*, 12(2):16:1–16:46, 2015.

[6] K. S. Booth and G. S. Lueker. Testing for the consecutive ones property, interval graphs, and planarity using PQ-tree algorithms. *Journal of Computational Systems Science*, 13:335–379, 1976.

[7] A. Bouchet. Reducing prime graphs and recognizing circle graphs. *Combinatorica*, 7(3):243–254, 1987.

[8] A. Bouchet. Unimodularity and circle graphs. *Discrete Mathematics*, 66(1-2):203–208, 1987.

[9] S. Chaplick, P. Dorbec, J. Kratochvíl, M. Montassier, and J. Stacho. Contact representations of planar graphs: Extending a partial representation is hard. In *WG'14*, volume 8747 of *Lecture Notes in Computer Science*, pages 139–151. 2014.

[10] S. Chaplick, R. Fulek, and P. Klavík. Extending partial representations of circle graphs. In *Graph Drawing*, volume 8242 of *Lecture Notes in Computer Science*, pages 131–142. 2013.

[11] S. Chaplick, G. Guśpiel, G. Gutowski, T. Krawczyk, and G. Liotta. The partial visibility representation extension problem. In *Graph Drawing and Network Visualization: 24th International Symposium, GD 2016*, volume 9801 of *Lecture Notes in Computer Science*, pages 266–279, 2016.

[12] M. Chudnovsky, N. Robertson, P. Seymour, and R. Thomas. The strong perfect graph theorem. *Annals of Mathematics*, 164:51–229, 2006.

[13] B. Courcelle. Circle graphs and monadic second-order logic. *J. Applied Logic*, 6(3):416–442, 2008.

[14] W. Cunningham. Decomposition of directed graphs. *SIAM J. Alg. and Disc. Methods*, 3:214–228, 1982.

[15] E. Dahlhaus. Parallel algorithms for hierarchical clustering and applications to split decomposition and parity graph recognition. *Journal of Algorithms*, 36(2):205–240, 1998.

[16] P. Damaschke. The hamiltonian circuit problem for circle graphs is NP-complete. *Information Processing Letters*, 32(1):1–2, 1989.

[17] H. de Fraysseix. Local complementation and interlacement graphs. *Discrete Mathematics*, 33(1):29–35, 1981.

[18] H. de Fraysseix and P. O. de Mendez. On a characterization of gauss codes. *Discrete & Computational Geometry*, 22(2):287–295, 1999.

[19] S. Even and A. Itai. Queues, stacks, and graphs. *Theory of Machines and Computation (Z. Kohavi and A. Paz, Eds.)*, pages 71–76, 1971.

[20] C. P. Gabor, K. J. Supowit, and W. Hsu. Recognizing circle graphs in polynomial time. *J. ACM*, 36(3):435–473, 1989.

[21] M. Garey, D. Johnson, G. Miller, and C. Papadimitriou. The complexity of coloring circular arcs and chords. *SIAM Journal on Algebraic Discrete Methods*, 1(2):216–227, 1980.

[22] F. Gavril. Maximum weight independent sets and cliques in intersection graphs of filaments. *Information Processing Letters*, 73(5-6):181–188, 2000.

[23] E. Gioan, C. Paul, M. Tedder, and D. Corneil. Practical and efficient circle graph recognition. *Algorithmica*, 69(4):759–788, 2014.

[24] K. R. Jampani and A. Lubiw. The simultaneous representation problem for chordal, comparability and permutation graphs. *Journal of Graph Algortihms and Applications*, 16(2):283–315, 2012.

[25] P. Klavík, J. Kratochvíl, T. Krawczyk, and B. Walczak. Extending partial representations of function graphs and permutation graphs. In Leah Epstein and Paolo Ferragina, editors, *Algorithms ESA 2012*, volume 7501 of *LNCS*, pages 671–682. 2012.

[26] P. Klavík, J. Kratochvíl, Y. Otachi, I. Rutter, T. Saitoh, M. Saumell, and T. Vyskočil. Extending partial representations of proper and unit interval graphs. *Algorithmica*, 77(4):1071–1104, 2017.

[27] P. Klavík, J. Kratochvíl, Y. Otachi, and T. Saitoh. Extending partial representations of subclasses of chordal graphs. *Theoretical Computer Science*, 576:85–101, 2015.

[28] P. Klavík, J. Kratochvíl, Y. Otachi, T. Saitoh, and T. Vyskočil. Extending partial representations of interval graphs. *Algorithmica*, 78(3):945–967, 2017.

[29] P. Klavík, J. Kratochvíl, and T. Vyskočil. Extending partial representations of interval graphs. In Mitsunori Ogihara and Jun Tarui, editors, *Theory and Applications of Models of Computation*, volume 6648 of *LNCS*, pages 276–285. 2011.

[30] P. Klavík and M. Saumell. Minimal obstructions for partial representations of interval graphs. In *ISAAC*, volume 8889 of *Lecture Notes in Computer Science*, pages 401–413, 2014.

[31] A. Kostochka and J. Kratochvíl. Covering and coloring polygon-circle graphs. *Discrete Mathematics*, 163(1-3):299–305, 1997.

[32] T. Krawczyk and B. Walczak. Extending partial representations of trapezoid graphs. In *WG 2017*, Lecture Notes in Computer Science, 2017.

[33] W. Naji. *Graphes de Cordes: Une Caracterisation et ses Applications*. PhD thesis, l'Université Scientifique et Médicale de Grenoble, 1985.

[34] J. Opatrny. Total ordering problem. *SIAM J. on Computing*, 8(1):111–114, 1979.

[35] S. Oum. Rank-width and vertex-minors. *J. Comb. Theory, Ser. B*, 95(1):79–100, 2005.

[36] M. Patrignani. On extending a partial straight-line drawing. In Patrick Healy and Nikola S. Nikolov, editors, *Graph Drawing*, volume 3843 of *LNCS*, pages 380–385. 2006.

[37] F. J. Soulignac. Bounded, minimal, and short representations of unit interval and unit circular-arc graphs. Chapter I: theory. *J. Graph Algorithms Appl*, 21(4):455–489, 2017.

[38] F. J. Soulignac. Bounded, minimal, and short representations of unit interval and unit circular-arc graphs. Chapter II: algorithms. *J. Graph Algorithms Appl*, 21(4):491–525, 2017.

[39] J. P. Spinrad. Recognition of circle graphs. *J. of Algorithms*, 16(2):264–282, 1994.

[40] J. P. Spinrad. *Efficient Graph Representations*. Field Institute Monographs, 2003.