Solving the Canonical Representation and Star System Problems for Proper Circular-Arc Graphs in Logspace

Johannes Köbler  Sebastian Kuhnert*  Oleg Verbitsky†

Humboldt-Universität zu Berlin, Institut für Informatik
Unter den Linden 6, D-10099 Berlin
15 March 2012

Abstract

We present a logspace algorithm that constructs a canonical intersection model for a given proper circular-arc graph, where canonical means that models of isomorphic graphs are equal. This implies that the recognition and the isomorphism problems for this class of graphs are solvable in logspace. For a broader class of concave-round graphs, that still possess (not necessary proper) circular-arc models, we show that the latter can also be constructed canonically in logspace. Finally, we consider the search version of the Star System Problem that consists in reconstruction of a graph from its closed neighborhood hypergraph. We solve it in logspace for the classes of proper circular-arc, concave-round, and co-convex graphs.

Contents

1 Introduction

2 Basic definitions
   2.1 Graphs and hypergraphs
   2.2 Arc and interval systems
   2.3 Arc and interval representations of hypergraphs

3 Canonical arc representations of hypergraphs

*Supported by DFG grant KO 1053/7–1.
†Supported by DFG grant VE 652/1–1. This work was initiated under support by the Alexander von Humboldt Fellowship. On leave from the Institute for Applied Problems of Mechanics and Mathematics, Lviv, Ukraine.
1 Introduction

With a family of sets $\mathcal{H}$ we associate the intersection graph $\mathbb{I}(\mathcal{H})$ on vertex set $\mathcal{H}$ where $A \in \mathcal{H}$ and $B \in \mathcal{H}$ are adjacent if and only if they have a non-empty intersection. We call $\mathcal{H}$ an intersection model of a graph $G$, if $G$ is isomorphic to $\mathbb{I}(\mathcal{H})$. Any isomorphism from $G$ to $\mathbb{I}(\mathcal{H})$ is called a representation of $G$ by an intersection model. If $\mathcal{H}$ consists of real intervals (resp. arcs of a circle), it is also referred to as an interval model (resp. an arc model). An intersection model $\mathcal{H}$ is proper if the sets in $\mathcal{H}$ are pairwise incomparable by inclusion. $G$ is called a (proper) interval graph if it has a (proper) interval model. The classes of circular-arc and proper circular-arc graphs are defined similarly. Throughout the paper we will use the shorthands CA and PCA respectively.

We design a canonical representation algorithm that for each PCA graph produces its representation by a proper arc model so that the models of isomorphic graphs are equal. Our algorithm is quite efficient: it takes logarithmic space. Note that this result provides a simultaneous solution in logspace of both the recognition and the isomorphism problems for the class of PCA graphs.

Earlier we have already solved in logspace the canonical representation problem for proper interval graphs [26]. Though circular-arc graphs may at first glance appear close relatives of interval graphs, the extension of the result of [26] achieved here is far from being straightforward. Differences between the two classes of graphs are well known. For example, while every proper interval graph is known to be representable by an intersection model consisting of unit intervals, the analogous statement for PCA graphs is not true. Another difference, very important in our
context, lies in relationship to interval and circular-arc hypergraphs that we will explain shortly.

An interval hypergraph is a hypergraph isomorphic to a system of intervals of integers. A circular-arc (CA) hypergraph is definable similarly if, instead of integer intervals, we consider arcs in a discrete cycle. With any graph \( G \), we associate its closed neighborhood hypergraph \( \mathcal{N}[G] = \{ \mathcal{N}[v] \}_{v \in V(G)} \) on the vertex set of \( G \), where for each vertex \( v \) we have hyperedge \( \mathcal{N}[v] \) consisting of \( v \) and all vertices adjacent to it. Roberts [36] (see also [16]) discovered that \( G \) is a proper interval graph iff \( \mathcal{N}[G] \) is an interval hypergraph. The circular-arc world is more complex. While \( \mathcal{N}[G] \) is a CA hypergraph if \( G \) is a PCA graph, the converse is not always true. PCA graphs are properly contained in the class of those graphs whose neighborhood hypergraphs are CA. Graphs with this property are called concave-round by Bang-Jensen, Huang, and Yeo [4] and Tucker graphs by Chen [9]. The latter name is justified by Tucker's result [39] saying that all these graphs are CA (even though not necessary proper CA).

By the Tucker result, it makes sense to consider the problem of constructing arc representations for the concave-round graphs. We solve it also in logspace and also in a canonical way.

Our working tool is a logspace algorithm for computing a canonical representation of CA hypergraphs. We derive it from our earlier result for interval hypergraphs [26] using the known complementation trick invented by Tucker [39]. This algorithm also gives us a logspace test of the so-called circular ones property of a Boolean matrix, which is an important problem arising in computational biology.

Our techniques are also applicable to the Star System Problem where, given a hypergraph \( \mathcal{H} \), we have to decide whether or not \( \mathcal{H} = \mathcal{N}[G] \) for some graph \( G \). In the stronger search version we have to find such a \( G \) if it exists. In the restriction of the Star System Problem to a class of graphs \( C \), we seek for \( G \) only in the class \( C \). We give logspace algorithms solving the search version of the problem for the classes of PCA and concave-round graphs.

**Comparison with previous work**

**Recognition, model construction, and isomorphism testing.** The recognition problem for PCA graphs, along with model construction, was solved in linear time by Deng, Hell, and Huang [15] and by Kaplan and Nussbaum [23]; and in \( \text{AC}^2 \) by Chen [8, 10]. Note that linear-time and logspace results are in general incomparable, while the existence of a logspace algorithm for a problem implies that it is solvable in \( \text{AC}^1 \). The isomorphism problem for PCA graphs was solved in linear time by Lin, Soulignac, and Szwarcfiter [29].

The isomorphism problem for the broader class of concave-round graphs was solved in \( \text{AC}^2 \) by Chen [9], who also designed a sequential \( \mathcal{O}(n^2) \) isomorphism test for this class in [12]. Circular-arc models for concave-round graphs were known to be constructible also in \( \text{AC}^2 \) (Chen [8]).

Improving the complexity upper bounds for the class of all CA graphs remains a challenging problem. While this class can be recognized in linear time by Mc-
### Table 1: Algorithms for classes of circular arc graphs.

| Class of graphs | Recognition and model construction | Isomorphism problem |
|-----------------|------------------------------------|---------------------|
| Circular arc    | $O(n + m)$                          | $O(nm)$             |
| Concave round   | $O(n + m)$                          | $O(n^2)$            |
| Proper CA       | $O(n + m)$                          | $O(n + m)$          |
| Interval        | $O(n + m)$                          | $O(n + m)$          |
| Proper interval | $O(n + m)$                          | $O(n + m)$          |

Connell’s algorithm [33] (along with constructing an intersection model), the best known isomorphism test for CA graphs due to Hsu [19] runs in time $O(mn)$, where $m$ and $n$ denote the number of edges and vertices in the input graph. This is further evidence showing that transition from interval to CA graphs is rather nontrivial: For the former class we have linear-time recognition and isomorphism algorithms due to the seminal work by Booth and Lueker [6, 32], and a canonical representation algorithm taking logarithmic space is designed in [26]. Available algorithms for the CA graphs and their subclasses are summarized in Table 1.

Deciding if a hypergraph is CA is equivalent to deciding if its incidence matrix has the circular ones property, that is, if the columns can be permuted so that the 1-entries in each row form a segment up to a cyclic shift. By the aforementioned result of Tucker [39], the latter problem easily reduces to deciding the related consecutive ones property (where no cyclic shift is allowed) or, equivalently, to the recognition problem of interval hypergraphs, for which linear-time [6, 20] and parallel $AC^2$ [13, 2] algorithms are known. Algorithms for the CA and the interval hypergraphs are summarized in Table 2.

A simple consequence of our canonical representation of CA hypergraphs in logspace is logspace recognition and isomorphism (even canonical labeling) algorithms for the class of circular convex bipartite graphs, studied by Liang and Blum [28]. Earlier an $AC^2$ isomorphism test for this class was known (Chen [9], who also presents a sequential $O(mn)$ algorithm in [11]).

**Star System Problem.** The general *Star System Problem* is known to be NP-complete (Lalonde [27]). It stays NP-complete if restricted to non-co-bipartite
graphs (Aigner and Triesch [1]) or to \(H\)-free graphs for each \(H\) in a family of graphs including cycles ans paths on at least 5 vertices (Fomin et al. [17]). The restriction to co-bipartite graphs has the same complexity as the general graph isomorphism problem (Aigner and Triesch [1]). Polynomial-time algorithms are known for \(H\)-free graphs for \(H\) being a cycle or a path on at most 4 vertices (Fomin et al. [17]) and for bipartite graphs (Boros et al. [7]). In these cases the search problem is tractable as well. An analysis of the algorithms in [17] for \(C_3\) - and \(C_4\)-free graphs shows that the Star System Problem for these classes is solvable even in logspace, and the same holds true for the class of bipartite graphs; see Section 9.

Solving in logspace the search version of the Star System Problem for PCA graphs and their relatives, we also show that any concave-round graph \(G\) is reconstructible from its closed neighborhood hypergraph up to isomorphism, that is,

\[
\text{if } \mathcal{N}[G] \cong \mathcal{N}[H], \text{ then } G \cong H. \tag{1.1}
\]

Previously such a reconstructibility result was known for complete graphs and complements of forests (Aigner and Triesch [1]). Moreover, the implication (1.1) is known to be true whenever both \(G\) and \(H\) are bipartite (Boros et al. [7]), co-bipartite [1,7], chordal (Harary and McKee [18]) or, more generally, \(C_4\)-free (Fomin et al. [17]). Note that the classes of chordal, interval, and proper interval graphs are recognizable in logspace by Reif [34] in combination with Reingold [35] or by methods of [26]. Therefore, the results of Fomin et al. [17] about \(C_4\)-free graphs imply that the Star System Problem for each of the classes of chordal, interval, and proper interval graphs is solvable in logspace.

**Roadmap**

We begin with basic definitions in Section 2. The canonical representation algorithm for CA hypergraphs is designed in Section 3. In Section 4 we state some conditions under which a CA hypergraph has an essentially unique arc representation. This result plays an important role in solving the canonical representation problem for PCA graphs and their relatives. In Section 5 we give an account of auxiliary results allowing us to make a reduction of the canonical representation problem from graphs
to hypergraphs. Our analysis of the problem for graphs is split into the non-co-bipartite case treated in Section 6 and the co-bipartite case in Section 7. The Star System Problem is studied in Section 8. We conclude with some open question in Section 9.

2 Basic definitions

2.1 Graphs and hypergraphs

We write \( G \cong H \) to say that \( G \) and \( H \) are isomorphic graphs. Referring to a class of graphs, we always assume that it is closed under isomorphism. The vertex set of a graph \( G \) is denoted by \( V(G) \). The complement of a graph \( G \) is the graph \( \overline{G} \) with the same vertex set such that two vertices are adjacent in \( \overline{G} \) iff they are not adjacent in \( G \). For a class of graphs \( C \), its co-class consists of the complements of graphs in \( C \). In this way there appear co-bipartite graphs, co-convex graphs etc.

The set of all vertices at distance at most (resp. exactly) 1 from a vertex \( v \in V(G) \) is called the closed (resp. open) neighborhood of \( v \) and denoted by \( N[v] \) (resp. \( N(v) \)). Note that \( N[v] = N(v) \cup \{v\} \).

If \( N[u] = N[v] \), we call the vertices \( u \) and \( v \) twins. According to our terminology, only adjacent vertices can be twins. If \( N(u) = N(v) \), we call \( u \) and \( v \) fraternal vertices. A vertex \( u \) is called universal if \( N[u] = V(G) \).

Since twins have the same adjacency to any other vertex, we can define the quotient-graph of a graph \( G \) whose vertices are the twin-classes of \( G \), where two classes are adjacent if their representatives are adjacent. Note that the quotient-graph is always twin-free.

The canonical labeling problem for a class of graphs \( C \) is, given a graph \( G \in C \) with \( n \) vertices, to compute a map \( \lambda_G : V(G) \to [1, \ldots, n] \) so that the graph \( G^* = \lambda_G(G) \), the image of \( G \) under \( \lambda_G \) on the vertex set \([1, \ldots, n]\), is the same for isomorphic input graphs. Equivalently, for all \( G \in C \) we have to compute an isomorphism from \( G \) to another graph \( G^* \) so that

\[ G^* = H^* \text{ whenever } G \cong H. \]

In fact, the condition \( V(G^*) = [1, \ldots, n] \) requires no special care as the vertices of \( G^* \) can be sorted and renamed. We say that \( \lambda_G \) is a canonical labeling and \( G^* \) is a canonical form of \( G \).

Recall that a hypergraph is a pair \((X, \mathcal{H})\), where \( X \) is a set of vertices and \( \mathcal{H} \) is a family of subsets of \( X \), called hyperedges. We will use the same notation \( \mathcal{H} \) to denote a hypergraph and its hyperedge set, and similarly to graphs we will write \( V(\mathcal{H}) \) referring to the vertex set \( X \) of a hypergraph \( \mathcal{H} \). A vertex is called isolated if it is contained in no hyperedge. If \( \mathcal{H} \) has no isolated vertex, then the hypergraph is determined by its hyperedge set and indication of the vertex set is actually not needed. Twins in a hypergraph are two vertices such that every hyperedge contains either both or none of them.
We will sometimes consider edge-colored hypergraphs. Specifically, by an edge-coloring of a hypergraph \( \mathcal{H} \) we understand a mapping \( c: \mathcal{H} \to C \) of the hyperedge set to a set of colors \( C \). An isomorphism from a hypergraph \( \mathcal{H} \) to a hypergraph \( \mathcal{K} \) is a bijection \( \phi: V(\mathcal{H}) \to V(\mathcal{K}) \) such that

- \( H \in \mathcal{H} \) iff \( \phi(H) \in \mathcal{K} \) for every \( H \subseteq V(\mathcal{H}) \), and
- \( c(H) = c(\phi(H)) \) for every \( H \in \mathcal{H} \).

Without loss of generality we can suppose that colors are integer numbers. In particular, this allows us to define a hypergraph with multiple hyperedges as an edge-colored hypergraph where colors are positive integers; then the color \( c(H) \geq 1 \) of a hyperedge \( H \in \mathcal{H} \) is called the multiplicity of \( H \).

The complement of a hypergraph \( \mathcal{H} \) is the hypergraph \( \overline{\mathcal{H}} = \{V(\mathcal{H}) \setminus H\}_{H \in \mathcal{H}} \) on the same vertex set. It is supposed that the hyperedge \( V(\mathcal{H}) \setminus H \) of \( \overline{\mathcal{H}} \) inherits the multiplicity of \( H \) in \( \mathcal{H} \).

We will use notation \( A \bowtie B \) for \( A, B \in \mathcal{H} \) to say that these sets have a non-empty intersection. Besides intersection, it will be useful to consider also other relations on \( \mathcal{H} \). We say that two hyperedges \( A \) and \( B \) overlap and write \( A \bowtie B \) if \( A \bowtie B \) but neither of the two sets includes the other. If additionally \( A \cup B \neq V(\mathcal{H}) \), we say that \( A \) and \( B \) strictly overlap and write \( A \bowtie^* B \). Note that for nonempty \( A \) and \( B \)

\[
A \bowtie B \text{ iff } A \bowtie B \text{ or } A \subseteq B \text{ or } A \supseteq B.
\]

Similarly, we define

\[
A \bowtie^* B \text{ iff } A \bowtie^* B \text{ or } A \subseteq B \text{ or } A \supseteq B.
\]

and say that such two sets strictly intersect. Note that \( \bowtie, \bowtie, \bowtie^* \), and \( \bowtie^* \) are symmetric relations. We call a hypergraph \( \mathcal{H} \) connected if it has no isolated vertex and the graph \( (\mathcal{H}, \bowtie) = \mathbb{I}(\mathcal{H}) \) is connected. We call \( \mathcal{H} \) strictly connected if it has no isolated vertex, the graph \( (\mathcal{H}, \bowtie^*) \) is connected, and neither \( \emptyset \) nor \( V(\mathcal{H}) \) are hyperedges of \( \mathcal{H} \).

Given a hypergraph \( \mathcal{H} \), we now define its dual hypergraph \( \mathcal{H}^* \). Suppose first that \( \mathcal{H} \) has neither twins nor multiple hyperedges. Given a vertex \( x \in V(\mathcal{H}) \), let \( x^* = \{ H \in \mathcal{H} : x \in H \} \). Then \( V(\mathcal{H}^*) = \mathcal{H} \) and \( \mathcal{H}^* = \{ x^* : x \in V(\mathcal{H}) \} \), that is, the vertices of \( \mathcal{H}^* \) are the hyperedges of \( \mathcal{H} \), and each vertex \( x \) of \( \mathcal{H} \) gives rise to the hyperedge \( x^* \) of \( \mathcal{H}^* \). The map \( x \mapsto x^* \) is an isomorphism from \( \mathcal{H} \) to \( (\mathcal{H}^*)^* \), that is called canonical. In the case that a hyperedge \( H \) has multiplicity \( m \) in \( \mathcal{H} \), the vertex set \( V(\mathcal{H}^*) \) contains exactly \( m \) distinct clones of \( H \), that are twins in \( \mathcal{H}^* \). If a vertex \( x \) has \( k \) twins in \( \mathcal{H} \), then the hyperedge \( x^* = \{ H \in V(\mathcal{H}) : x \in H \} \), being \( k \) times contributed, has multiplicity \( k \) in \( \mathcal{H}^* \). Therefore, this hyperedge of \( \mathcal{H}^* \) contributes \( k \) twin-vertices in \( (\mathcal{H}^*)^* \). It is natural to give them names \( x_1^*, \ldots, x_k^* \), where \( x_1 = x, \ldots, x_k \) are the twins of \( x \). Then we can again consider the one-to-one map \( v \mapsto v^* \), that is a (canonical) isomorphism from \( \mathcal{H} \) to \( (\mathcal{H}^*)^* \).

The incidence graph of a hypergraph \( \mathcal{H} \) is the bipartite graph with vertex classes \( V(\mathcal{H}) \) and \( \mathcal{H} \) where \( x \in V(\mathcal{H}) \) and \( H \in \mathcal{H} \) are adjacent if \( x \in H \). If a hyperedge
Given a graph $G$, we associate with it two hypergraphs defined on the vertex set $V(G)$. The *closed neighborhood hypergraph* of a graph $G$ is defined by $\mathcal{N}[G] = \{N[v]\}_{v \in V(G)}$. Note that two vertices are twins in the hypergraph $\mathcal{N}[G]$ iff they are twins in the graph $G$. Furthermore, $\mathcal{N}(G) = \{N(v)\}_{v \in V(G)}$ is the *open neighborhood hypergraph* of $G$. Given a bipartite graph $G$ and a bipartition $V(G) = U \cup W$ of its vertices into two independent sets, by $\mathcal{N}_U(G)$ we denote the hypergraph $\{N(w)\}_{w \in W}$ on the vertex set $U$. Note that the incidence graphs of both $\mathcal{N}_U(G)$ and $\mathcal{N}_W(G)$ are isomorphic to $G$ and that $(\mathcal{N}_U(G))^* \cong \mathcal{N}_W(G)$.

### 2.2 Arc and interval systems

A *circle of size* $n$, denoted by $C_n$, is a directed cycle on $n$ vertices. The vertices of $C_n$ will be called points. An arc $[a, b]$ with extreme points $a$ and $b$ consists of the points in the circle appearing in the directed path from $a$ to $b$. The successor relation on $C_n$ will be referred to as a *circular order* of the points of $C_n$; it induces a linear order on any arc. In order to refer to the extreme points of an arc $A$, we will often use notation $A = [a^-, a^+]$ and call $a^-$ the *start point* and $a^+$ the *end point* of $A$. Note that the extreme points $a^-$ and $a^+$ are uniquely determined by the set $A$, unless $A = C_n$. Given a circle, its orientation will be referred to as *clockwise*. We will very oft model the circle $C_n$ by the cyclic group $\mathbb{Z}_n$, where the successor of an element $x$ is $(x + 1) \mod n$.

We also use the standard notation $[a, b] = \{x : a \leq x \leq b\}$ for *intervals* of positive integers. Similarly to arcs, the intervals within the segment $[1, n]$ could be introduced graph-theoretically using the successor relation on $[1, n]$.

We will allow the empty interval and the empty arc.

An *arc system* is a hypergraph whose vertex set is a circle and hyperedges are arcs. Note that the complement $\overline{A}$ of an arc system $A$ is an arc system as well. An *interval system* is a hypergraph whose vertex set is a segment of integers $[1, n]$ and hyperedges are intervals. Identifying the segment $[1, n]$ with a directed path on $n$ points (or embedding it in $\mathbb{Z}_{n+1}$), we see that the notion of an arc system extends the notion of an interval system.

An arc or an interval system is *tight* if its arcs or intervals have the following property: whenever $A = [a^-, a^+]$ includes a nonempty $B = [b^-, b^+]$, it holds either $a^- = b^-$ or $a^+ = b^+$; this condition is not required if $A = C_n$ in the case of an arc system on a circle $C_n$. Note that, if an arc system $A$ is tight, its complement $\overline{A}$ is also tight.

We define the *lexicographic order* $\leq^*$ between intervals of integers by setting $A \leq^* B$ if $a^- < b^-$ or if $a^- = b^-$ and $a^+ \leq b^+$. This is a linear order. Given a circle $C_n$ with successor relation $\prec$, we also define the lexicographic order between arcs on $C_n$ excepting the arc $C_n$. Now this is a circular order, that is, a successor relation $\prec^*$ arranging the arcs into a directed cycle. Given two arcs $A, B \neq C_n$, we call $B$
the successor of $A$ and write $A \prec^* B$ if

$$a^- = b^- \text{ and } a^+ \prec b^+$$

or if

$$a^- \prec b^- \text{ while } |A| = n - 1 \text{ and } |B| = 1.$$ 

The last two conditions say that $A$ is the longest among all arcs with start point $a^-$ and $B$ is the shortest among all arcs with start point $b^-$. 

The lexicographic order $\prec^*$ induces the circular order $\prec^*_A$ on any arc system $A$ on $C_n$. For $A, B \in A$ we define $A \prec^*_A B$ if $A \prec^* B$ or if in the chain $A \prec^* X_1 \prec^* \ldots \prec^* X_k \prec^* B$ none of the arcs $X_i$ belongs to $A$.

### 2.3 Arc and interval representations of hypergraphs

An arc representation of a hypergraph $\mathcal{H}$ is an isomorphism $\lambda$ from $\mathcal{H}$ to an arc system $A$. The arc system $A$ is referred to as an arc model of $\mathcal{H}$. The notions of an interval representation and an interval model of a hypergraph are introduced similarly. Hypergraphs having arc representations are called circular-arc hypergraphs, and those having interval representations are called interval hypergraphs.

Recall that by a circular order on a finite set we mean a successor relation $\prec$ on this set that determines a directed cycle on it. An arc representation of a hypergraph $\mathcal{H}$ can then be defined as a circular order $\prec$ on $V(\mathcal{H})$ such that the hyperedges of $\mathcal{H}$ are arcs w.r.t. $\prec$. We will call such a relation $\prec$ an arc ordering of $\mathcal{H}$. Thus, a hypergraph is CA if it has an arc ordering. This terminology has the advantage that if two arc representations are obtainable from one another by rotation of the circle, they are not considered different anymore. Similarly, we will use the notion of an interval ordering of $\mathcal{H}$.

An arc or an interval representation (or ordering) of a hypergraph is tight if the corresponding arc or interval model is tight. We call an interval (resp. circular-arc) hypergraph tight, if it admits a tight interval (resp. arc) representation. Recognition of tight interval or CA hypergraphs reduces to recognition of interval or CA hypergraphs, respectively. To see this, given a hypergraph $\mathcal{H}$, define its tightened hypergraph $\mathcal{H}^\oplus$ by

$$\mathcal{H}^\oplus = \mathcal{H} \cup \{A \setminus B : A, B \in \mathcal{H} \text{ and } B \subset A\}.$$ 

**Lemma 2.1.** $\lambda$ is a tight interval (resp. arc) representation of a hypergraph $\mathcal{H}$ iff it is an interval (resp. arc) representation of the hypergraph $\mathcal{H}^\oplus$. ■

A bipartite graph $G$ is called convex (resp. circular convex) if its vertex set admits splitting into two independent sets, $U$ and $W$, such that $N_U(G)$ is an interval (resp. circular-arc) hypergraph. If both $N_U(G)$ and $N_W(G)$ are interval hypergraphs, $G$ is called biconvex.

---

1The equivalent notions of arc/interval representations and arc/interval orderings for hypergraphs will be used in parallel, though the latter will become preferable starting from Section 5 in order to avoid confusing with arc/interval representations of graphs that appear there.
3 Canonical arc representations of hypergraphs

In the canonical representation problem for interval (CA) hypergraphs we have, for each input hypergraph, to compute its interval (resp. arc) representation such that the resulting interval (resp. arc) models are equal for isomorphic input hypergraphs. Representations and models produced by an algorithm solving the problem will be called canonical.

Though in other sections we deal with uncolored hypergraphs, note that the notions of a model, a representation, and the canonical representation problem are quite meaningful for edge-colored hypergraphs.

Theorem 3.1. ([26]) The canonical representation problem for edge-colored interval hypergraphs is solvable in logspace.

In [26], we stated this result for interval hypergraphs with multiple hyperedges. Extension to the edge-colored hypergraphs takes no extra work because any set of colors can be modeled as a set of positive integers (just as we defined edge-colored hypergraphs in Section 2.1).

Now we generalize Theorem 3.1 to CA hypergraphs. Given a hyperedge $H$ of a hypergraph $H$, we use notation $H = V(H) \setminus H^c$.

Theorem 3.2. The canonical representation problem for edge-colored CA hypergraphs is solvable in logspace.

Proof. We will describe a logspace reduction to the canonical representation problem for edge-colored interval hypergraphs. To facilitate the exposition, we first treat the case of CA hypergraphs with no coloring. Let $H$ be such a hypergraph. For each vertex $x \in V(H)$, we construct a hypergraph $H_x = \{H_x\} \cup H$ on the same vertex set and its edge-coloring $c_x : H_x \to \{0, 1, 2\}$ as follows. If $x \notin H$, then $H_x = H$ and $H_x = H^c$ otherwise. We set $c_x(H_x)$ to 0 if $x \notin H$ and $H^c \notin H$, to 1 if $x \in H$ and $H^c \notin H$, and to 2 if both $H$ and $H^c$ are in $H$.

For another hypergraph $H'$ and a vertex $y \in V(H')$, let the hypergraph $H'_y$ and its edge-coloring $c'_y$ be defined in the same way. If $\phi$ is an isomorphism from $H$ to $H'$, then $\phi$ is also an isomorphism from the edge-colored hypergraph $(H_x, c_x)$ to the edge-colored hypergraph $(H'_x, c'_x)$ for every $x \in V(H)$. Thus, the multiset of isomorphism types of all colored hypergraphs $(H_x, c_x)$ is an isomorphism invariant of the hypergraph $H$, that is, it is the same for any isomorphic copy of $H$.

If $A$ is an arc system on the cycle $\mathbb{Z}_n = \{0, 1, \ldots, n-1\}$, then $A_0$ is an interval system on the interval $[0, n-1]$. Using symmetry of the cycle, we conclude from this observation that if $H$ is a CA hypergraph, then every $H_x$ is an interval hypergraph. This fact is stated by Tucker in [39] Theorem 1] in terms of incidence matrices.

In the next step, we apply the canonical representation algorithm from Theorem 3.1 to each edge-colored interval hypergraph $(H_x, c_x)$. It produces an interval model of $(H_x, c_x)$, consisting of an interval system $I_x$ along with edge-coloring $d_x$, and an interval representation $\lambda_x$ (recall that the latter is a map from $V(H_x)$ onto $V(I_x)$ providing an isomorphism from $(H_x, c_x)$ to $(I_x, d_x)$). Note that the set
\{(I_x, d_x)\}_{x \in V(H)} is an isomorphism invariant of the hypergraph \(H\). Let \((I_x, d_x)\) be the lexicographically smallest element of this set (or any one of such elements); this is an isomorphism invariant of \(H\) as well. Finally, we take \(\lambda_x\) as the output. In the rest of the proof we argue that this is really a canonical arc representation of \(H\).

For an edge-colored hypergraph \((K, c)\) with color set \(\{0, 1, 2\}\) we define \(K^c\) to be the uncolored hypergraph obtained from \((K, c)\) by keeping a hyperedge \(K \in K\) unchanged if \(c(K) = 0\), by replacing \(K\) with \(\overline{K}\) if \(c(K) = 1\), and by adding \(\overline{K}\) as a new hyperedge while keeping \(K\) if \(c(K) = 2\). Let us make a few observations about this operation. If \(\mathcal{A}\) is an arc system with any \(\{0, 1, 2\}\)-coloring \(c\), then \(\mathcal{A}^c\) is an arc system too. In particular, if \(\mathcal{A}\) is an interval system on the interval of length \(n\), then \(\mathcal{A}^c\) becomes an arc system after closing the interval into the cycle \(\mathbb{Z}_n\). If \(\phi\) is an isomorphism from \((K, c)\) to another edge-colored hypergraph \((\mathcal{M}, d)\), then \(\phi\) is an isomorphism from \(K^c\) to \(\mathcal{M}^d\). Finally, this is a kind of the inverse transformation for our construction of \((H_x, c_x)\) as we have \(H_x^c = H\) for any \(x\).

Consider now the arc system \(T_x^{d_x}\). Since \(\lambda_x\) is an isomorphism from \(H_x^c = H\) to \(T_x^{d_x}\), this map is an arc representation of \(H\). The arc model \(T_x^{d_x}\) is canonical because it is deterministically obtained from an isomorphism invariant of \(H\).

In the general case, when an input hypergraph \(H\) is endowed with an edge-coloring \(h: \mathcal{H} \rightarrow \mathbb{Z}\), our algorithm again constructs the hypergraphs \(H_x\), now with edge-coloring \(h_x(H_x) = 3h(H) + c_x(H_x)\), where \(c_x\) is as defined above. The next step is, as above, to compute the canonical interval representation \(\lambda_x\) for each \((H_x, h_x)\) and to choose the \(\lambda_x\) producing the lexicographically smallest interval model.

In [26] Corollary 5.2 we used Theorem 3.1 to solve the canonical labeling problem for the class of convex graphs in logspace. Similarly, we can now use Theorem 3.2 to extend this result to circular convex graphs.

**Corollary 3.3.** The canonical labeling problem for the class of circular convex graphs is solvable in logspace.

**Proof.** We can focus on connected graphs because any disconnected circular convex graph is actually convex, and the canonical labeling algorithm of [26] does the job in this case. Note also that the algorithm we design below for connected circular convex graphs applies, in particular, to connected convex graphs. Furthermore, the general case reduces to the connected case by determining connected components and merging their canonical labelings in the lexicographic order of the canonical forms.

By a *(colored) bigraph* we mean a graph \(G\) whose vertices are colored in red and blue so that both color classes are independent sets. In other words, this is a bipartite graph with specified and distinguished vertex classes. With \(G\) we associate a hypergraph \(R_G\) whose vertices are the blue vertices of \(G\) and where each red vertex \(u\) of \(G\) contributes hyperedge \(N(u)\) (thus, \(R_G\) has multiple hyperedges if \(G\) has fraternal red vertices). Extending our terminology, we call a bigraph \(G\) *circular convex* if \(R_G\) is a CA hypergraph. For a given bigraph, the property of being circular convex is decidable in logspace because the class of CA hypergraphs is decidable in logspace by Theorem 3.2.
Note also that the canonical labeling problem can be considered as well for vertex-colored graphs. We now claim that it suffices to solve the canonical labeling problem for connected circular convex bigraphs. To show this, note that if $G$ is a connected bipartite graph, it has a unique bipartition and, hence, exactly two bigraph versions, $G'$ and $G''$. If only one of them, say $G'$, is circular convex and we can find its canonical labeling $\lambda_{G'}$, then this map can serve as a canonical labeling of $G$. If both, then we have two candidates, $\lambda_{G'}$ and $\lambda_{G''}$, for a canonical labeling of $G$ and we choose one that produces a canonical form with lexicographically least adjacency matrix (if $\lambda_{G'}$ and $\lambda_{G''}$ produce the same canonical form, they are equally suitable).

Thus, suppose that as input we have a connected circular convex bigraph $G$. Postponing some small technical complications, we first consider the case when $G$ has no fraternal red vertices. We construct the CA hypergraph $R = R_G$ and, using the algorithm of Theorem 3.2, compute its canonical representation $\lambda_R$ and canonical model $R^*$. Given $R^*$, we define a bigraph $G^*$ with the sets of blue vertices $V(R^*)$ and red vertices $R^*$, connecting a blue vertex $b \in V(R^*)$ and a red vertex $R \in R^*$ by an edge if and only if $b \in R$. Note that the bigraph $G^*$ is the same for isomorphic input bigraphs because, if $G \cong H$, then $R_G \cong R_H$, which gives the same hypergraph $R^*$. Furthermore, we construct a map $\lambda_G$ by $\lambda_G(v) = \lambda_R(v)$ for a blue vertex $v$ of $G$ and $\lambda_G(u) = \lambda_R(N(u))$ for a red vertex $u$ of $G$. Since $\lambda_R$ is an isomorphism from $R$ to $R^*$, we have $\lambda_R(N(u)) \in R^*$. Since $v \in N(u)$ iff $\lambda_G(v) \in \lambda_G(u)$, the vertices $v$ and $u$ are adjacent in $G$ iff $\lambda_G(v)$ and $\lambda_G(u)$ are adjacent in $G^*$. Therefore, the map $\lambda_G$ is an isomorphism from $G$ to $G^*$ and, hence, can be output as the desired canonical labeling of $G$.

If $G$ has fraternal red vertices, we need to fix a map $\mu : V(G) \to \mathbb{N}$ that takes any class of $k$ fraternal vertices onto the set $\{1, \ldots, k\}$ (therewith assigning an identifier to each fraternal vertex in the class). We modify the definition of $G^*$ as follows. If a hyperedge $R$ has multiplicity $k$ in $R^*$, it contributes a class of $k$ red fraternal vertices in $G^*$, whose members are arranged in a sequence $R_1, \ldots, R_k$. Finally, $\lambda_G$ is constructed so that if $R = \lambda_R(N(u))$, then $\lambda_G(u) = R_{\mu(u)}$.

The canonical labeling problem for a class of hypergraphs $\mathcal{C}$ is defined exactly as for graphs. Notice a similarity between the pairs of notions canonical labeling/canonical form and canonical representation/canonical model for CA and interval hypergraphs. Canonical representation algorithms of Theorems 3.1 and 3.2 readily imply that the canonical labeling problem is solvable in logarithmic space for the classes of interval and CA hypergraphs respectively. In Section 4 we will need an extension of this result to the dual classes.

**Lemma 3.4.** Suppose that the canonical labeling problem for a class of hypergraphs $\mathcal{C}$ is solvable in logarithmic space. Then the same holds true for the class of dual hypergraphs $\mathcal{C}^* = \{ K^* : K \in \mathcal{C} \}$.

**Proof.** We begin with a general observation. Suppose that $\phi : V(\mathcal{H}) \to V(\mathcal{K})$ is an isomorphism from a hypergraph $\mathcal{H}$ to a hypergraph $\mathcal{K}$. Let $\bar{\phi} : \mathcal{H} \to \mathcal{K}$ denote the action of $\phi$ on sets, that is, $\bar{\phi}(H) = \{ \phi(x) : x \in H \}$. This is a bijection from $\mathcal{H}$
to \( \mathcal{K} \) respecting the multiplicities of hyperedges. Therefore, \( \bar{\phi} \) can be considered a bijection from \( V(\mathcal{H}^*) \) to \( V(\mathcal{K}^*) \). Note that \( \bar{\phi} \) takes a set \( x^* = \{ H \in V(\mathcal{H}) : x \in H \} \) to a set \( \phi(x)^* \). It follows that \( \bar{\phi} \) is an isomorphism from the dual hypergraph \( \mathcal{H}^* \) to the dual hypergraph \( \mathcal{K}^* \).

Given a hypergraph \( \mathcal{H} \in \mathcal{C}^* \), we have to construct a one-to-one map \( \lambda \) with domain \( V(\mathcal{H}) \) so that the isomorphic image \( \lambda(\mathcal{H}) \) is the same for all isomorphic copies of \( \mathcal{H} \). Since \( \mathcal{H} \) is isomorphic to the dual of a hypergraph in \( \mathcal{C} \), the dual \( \mathcal{H}^* \) belongs to \( \mathcal{C} \). Running the canonical labeling problem for this class, we obtain a canonical labeling \( \mu \) transforming \( \mathcal{H}^* \) to its canonical form \( \mu(\mathcal{H}^*) \). According to the preceding discussion, the map \( \bar{\mu} \) is an isomorphism from \( (\mathcal{H}^*)^* \) to \( (\mu(\mathcal{H}^*))^* \). Denote the canonical isomorphism from \( \mathcal{H} \) to \( (\mathcal{H}^*)^* \) by \( \kappa \) and define \( \lambda = \bar{\mu} \circ \kappa \). Being the composition of isomorphisms, \( \lambda \) is an isomorphism from \( \mathcal{H} \) to \( (\mu(\mathcal{H}^*))^* \). The latter hypergraph does not change if we replace \( \mathcal{H} \) with any isomorphic copy because this will not change the isomorphism type of \( \mathcal{H}^* \) and, hence, will not change the canonical form \( \mu(\mathcal{H}^*) \).

Corollary 3.5. The duals of CA hypergraphs admit canonical labeling in logspace.

4 A uniqueness property for tight representations of hypergraphs

An important role in our further analysis will be played by the fact that if an interval or CA hypergraph has sufficiently strong connectedness properties, then its interval/arc representation is essentially unique. First of all, we have to explain what “essentially” means here.

Let \( \mathcal{H} \) be a hypergraph. A transposition of twins is a one-to-one map of \( V(\mathcal{H}) \) onto itself that fixes each vertex except, possibly, a pair of twins of \( \mathcal{H} \). Any composition \( \pi \) of transpositions of twins is called a permutation of twins. Note that, if \( \lambda \) is an interval or arc representation of \( \mathcal{H} \), then the composition \( \lambda \circ \pi \) is, respectively, an interval or arc representation of \( \mathcal{H} \) as well. We call two representations \( \lambda \) and \( \lambda' \) equivalent up to permutation of twins if \( \lambda' = \lambda \circ \pi \) for some permutation of twins \( \pi \). For twin-free hypergraphs this relation just coincides with equality of representations.

Lemma 4.1. Arc representations \( \lambda \) and \( \lambda' \) are equivalent up to permutation of twins iff \( \lambda(H) = \lambda'(H) \) for every hyperedge \( H \in \mathcal{H} \).

Proof. Suppose that \( \lambda' = \lambda \circ \pi \) for a permutation of twins \( \pi \). Let \( H \) be an arbitrary hyperedge of \( \mathcal{H} \). Since \( \tau(H) = H \) for any transposition of twins \( \tau \), we have \( \pi(H) = H \) and, hence, \( \lambda(H) = \lambda'(H) \).

To prove the claim in the backward direction, define a slot of a hypergraph \( \mathcal{H} \) to be an inclusion-maximal subset \( S \) of \( V(\mathcal{H}) \) such that each hyperedge \( H \in \mathcal{H} \) contains either all of \( S \) or none of it. Thus, two vertices of \( \mathcal{H} \) are twins iff they are...
in the same slot. It follows that $\pi$ is a permutation of twins exactly when $\pi(S) = S$ for every slot $S$ of $\mathcal{H}$.

Suppose that $\lambda(H) = \lambda'(H)$ for every hyperedge $H \in \mathcal{H}$. Since a slot is a Boolean combination of hyperedges, we have $\lambda(S) = \lambda'(S)$ for every slot $S$ of $\mathcal{H}$ and, therefore, $\lambda' \circ \lambda^{-1}(S) = S$ for all slots. It follows that $\lambda' \circ \lambda^{-1}$ is a permutation of twins, hence $\lambda$ and $\lambda'$ are equivalent up to permutation of twins.

The rotations $x \mapsto (x + s) \mod n$ and the reflection $x \mapsto (-x) \mod n$ will be considered symmetries of the circle $\mathbb{Z}_n$. The segment $[1, n]$ has a unique symmetry, namely the reflection $x \mapsto n + 1 - x$. Note that, if $\lambda$ is an interval or arc representation of $\mathcal{H}$ and $\sigma$ is a symmetry of the circle or the interval respectively, then the composition $\sigma \circ \lambda$ is an interval or arc representation of $\mathcal{H}$ as well.

Let $\lambda: V(\mathcal{H}) \to \mathbb{C}$ and $\lambda': V(\mathcal{H}) \to \mathbb{C}$ be arc or interval representations of $\mathcal{H}$, where $\mathbb{C}$ is, respectively, the circle $\mathbb{Z}_n$ or the interval $[1, n]$. We call $\lambda$ and $\lambda'$ equivalent up to congruence and permutation of twins if $\lambda' = \sigma \circ \lambda \circ \pi$ for some symmetry $\sigma$ and permutation of twins $\pi$. This is an equivalence relation between representations because both symmetries and permutations of twins form groups.

A result stated by Chen and Yesha in [13, Theorem 2] can be recast as follows: if an interval hypergraph is connected with respect to the overlapping relation $\join$, then is has, up to congruence (i.e., reflection) and permutation of twins, a unique interval representation. Another proof of this statement can be found in [26, Lemma 6.2]. In the following result, we relax the overlap-connectedness assumption just to connectedness but claim the uniqueness only for tight orderings. We also extend this to CA hypergraphs but then we have to strengthen the connectedness assumption to strict connectedness.

**Lemma 4.2.**

1. A connected interval hypergraph has, up to congruence (i.e., reflection) and permutation of twins, a unique tight interval representation.

2. A strictly connected CA hypergraph has, up to congruence (i.e., reflection and rotations) and permutation of twins, a unique tight arc representation.

Note that in the case of twin-free hypergraphs, Lemma 4.2 claims that all interval (resp. arc) orderings of a (resp. strictly) connected hypergraph are equal up to reversal.

**Proof.** We will prove Part 2. The proof of Part 1 is virtually the same and even somewhat simpler as not all arc configurations (like one of the dashed configurations in Fig. 1) can occur in the interval.

Let us first explain the strategy of the proof. Given a hypergraph $\mathcal{H}$ with $n$ vertices, we will prove that any strictly connected subhypergraph $\mathcal{K} \subseteq \mathcal{H}$ with $k$ hyperedges has, up to congruence and permutation of twins, a unique representation on the circle of size $n$. This will be done by induction on $k$. The base cases are $k = 1, 2$. In order to make the inductive step, it suffices to show that, whenever $k \geq 2$, any representation $\lambda$ of $\mathcal{K}$ has, up to permutation of twins, a unique extension to a
representation of $\mathcal{K} \cup \{H\}$, for any $H \in \mathcal{H} \setminus \mathcal{K}$ such that $\mathcal{K} \cup \{H\}$ is strictly connected. By Lemma 4.1 this actually means to show that the whole arc $\lambda(H)$ (though not necessarily each point $\lambda(v)$ for $v \in H$) is uniquely determined. Moreover, it suffices to do this job in the case of $k = 2$. The reason is that $\mathcal{K}$ always contains two hyperedges $A$ and $B$ such that the sequence $A, B, H$ forms a strictly connected path.

Before going into detail, we introduce some terminology. Consider two arcs $[a^-, a^+]$ and $[b^-, b^+]$ and suppose that $[b^-, b^+] \pitchfork [a^-, a^+]$ or $[b^-, b^+] \subset [a^-, a^+]$. We will say that $[b^-, b^+] \text{ intersects } [a^-, a^+] \text{ clockwise}$ if $a^+ \in [b^-, b^+]$ and $\text{counter-clockwise}$ if $a^- \in [b^-, b^+]$.

All possible positions of a single hyperedge $A$ on the circle are congruent by rotation. All possible positions of two strictly overlapping hyperedges $A$ and $B$ are congruent by rotation and reflection because the intersection of the corresponding arcs $\lambda(A)$ and $\lambda(B)$ is always an arc of length $|A \cap B|$. If $A$ and $B$ are comparable under inclusion, recall that we only consider tight representations.

For the inductive step, consider three hyperedges $A, B,$ and $H$ such that $A \gg^* B$ and $B \gg^* H$. We have to show that the arc $\lambda(H)$ is completely determined by the arcs $\lambda(A)$ and $\lambda(B)$. The relation $B \gg^* H$ fixes the length of the intersection $\lambda(H) \cap \lambda(B)$ and, hence, leaves for $\lambda(H)$ exactly two possibilities depending on whether this intersection is clockwise or counter-clockwise.

We split our analysis into three cases.

Case 1: $A \pitchfork B$. Without loss of generality, we suppose that $\lambda(B)$ intersects $\lambda(A)$ clockwise; the case of counter-clockwise intersection is symmetric. Consider first the subcase in which $B \pitchfork H$. Looking at the possible configurations for the arc system $\{\lambda(A), \lambda(B), \lambda(H)\}$, all shown in Fig. 1, we see that $\lambda(H)$ intersects $\lambda(B)$ counter-clockwise exactly if the sets $A \setminus B$ and $H \setminus B$ are comparable under inclusion, i.e.,

$$A \setminus B \subseteq H \setminus B \quad \text{or} \quad H \setminus B \subseteq A \setminus B. \quad (4.1)$$

For the remaining subcases, when $B$ and $H$ are comparable under inclusion, all possible configurations for the arc system $\{\lambda(A), \lambda(B), \lambda(H)\}$ are shown in Fig. 2. If $B \subset H$, we see that $\lambda(B)$ intersects $\lambda(H)$ clockwise exactly under the condition (4.1). If $H \subset B$, then $\lambda(H)$ intersects $\lambda(B)$ counter-clockwise iff $A \cap B \subseteq H$.

Case 2: $A \supset B$. Without loss of generality, we suppose that $\lambda(B)$ intersects

![Figure 1: Proof of Lemma 4.2, case 1: $A \pitchfork B$ and $\lambda(B)$ intersects $\lambda(A)$ clockwise; $B \pitchfork H$. On the left side: $\lambda(H)$ intersects $\lambda(B)$ counter-clockwise. On the right side: $\lambda(H)$ intersects $\lambda(B)$ clockwise.](image-url)
Figure 2: Proof of Lemma 4.2.2, case 1: $A \not\subset B$ and $\lambda(B)$ intersects $\lambda(A)$ clockwise; $B$ and $H$ are comparable under inclusion.

Figure 3: Proof of Lemma 4.2.2, case 2: $B \subset A$ and $\lambda(B)$ intersects $\lambda(A)$ clockwise.
\( B \preceq^* H: \)
\[
\begin{align*}
&\lambda(B) \\
&\lambda(A) \\
&\lambda(H) \\
&\lambda(B) \\
&\lambda(A) \\
&\lambda(H)
\end{align*}
\]
\( H \subset B: \)
\[
\begin{align*}
&\lambda(B) \\
&\lambda(A) \\
&\lambda(H) \\
&\lambda(B) \\
&\lambda(A) \\
&\lambda(H)
\end{align*}
\]
\( B \subset H: \)
\[
\begin{align*}
&\lambda(B) \\
&\lambda(A) \\
&\lambda(H) \\
&\lambda(B) \\
&\lambda(A) \\
&\lambda(H)
\end{align*}
\]

Figure 4: Proof of Lemma 4.2.2, case 3: \( A \subset B \) and \( \lambda(A) \) intersects \( \lambda(B) \) clockwise.

\( \lambda(A) \) clockwise; see Fig. 4. If \( B \preceq^* H \), then \( \lambda(H) \) intersects \( \lambda(B) \) counter-clockwise exactly when the familiar condition (4.1) holds true. If \( B \subset H \), then \( \lambda(B) \) intersects \( \lambda(H) \) clockwise exactly under the same condition. Thus, these two subcases do not differ much from the corresponding subcases of Case 1. If \( H \subset B \), then \( \lambda(H) \) is forced to intersect \( \lambda(B) \) counter-clockwise by the condition that \( \{\lambda(A), \lambda(B), \lambda(H)\} \) is a tight arc system.

Case 3: \( A \subset B \). Without loss of generality, we suppose that \( \lambda(A) \) intersects \( \lambda(B) \) clockwise; see Fig. 4. If \( B \preceq^* H \), then \( \lambda(H) \) intersects \( \lambda(B) \) clockwise iff \( H \cap B \subseteq A \). If \( H \subset B \), then \( \lambda(H) \) intersects \( \lambda(B) \) clockwise iff the sets \( H \) and \( A \) are comparable under inclusion. Finally, if \( B \subset H \), then \( \lambda(B) \) is forced to intersect \( \lambda(H) \) clockwise by the tightness condition.

5 Arc and interval representations of graphs

The definition of interval and CA graphs in the introduction refers to continuous segments and arcs. We now give an equivalent discrete version. An arc representation of a graph \( G \) is an isomorphism \( \ell \) from \( G \) to the intersection graph \( \Pi(A) \) of an arc system \( A \). The latter is called an arc model of \( G \). Graphs admitting an arc representations are called circular-arc (abbreviated as CA). Interval graphs and their interval representations and models are defined similarly.

We will call a hypergraph proper if its hyperedges are incomparable w.r.t. the inclusion relation. Tight arc and interval systems were introduced in Section 2.2. An arc or an interval representation of a graph is proper (resp. tight) if the corresponding arc or interval model is proper (resp. tight). Proper CA or interval graphs are those graphs having proper arc or interval models.

Starting from this section, we will often observe a striking similarity between proper CA graphs with non-bipartite complement and proper interval graphs: many
facts that are true for the latter class, with a bit more effort can be shown also for the former class.

In Subsection 5.1 we overview classes of graphs definable in terms of their closed or open neighborhood hypergraphs when one imposes the condition that these hypergraphs are interval or CA. In this way we obtain tight relations between known classes of graphs and hypergraphs, which allows us to use our algorithms for hypergraphs from Section 3 while solving the canonical representation problem for classes of CA graphs. We also give an account of inclusions between the graph classes under consideration. We further elaborate of the relationship between graphs and hypergraphs in Subsection 5.2. Given a proper interval/arc representation of a graph \( G \), we define a geometric order on the vertices of \( G \) according to the order among the corresponding interval/arc. We notice, in particular, that any geometric order is a tight interval/arc ordering for the hypergraph \( \overline{N}(G) \). This is strengthened in Section 5.3 where it is shown that actually any interval ordering of \( \overline{N}(G) \) is tight.

Moreover, any arc interval ordering of \( \overline{N}(G) \) is tight if \( G \) has non-bipartite complement. Together with the uniqueness results in Section 4, this suggests a canonization procedure for interval and non-co-bipartite PCA graphs that will be realized later in Section 6. In Subsection 5.4 we will see that, vice versa, any interval ordering of \( \overline{N}(G) \) can be converted in logspace into a proper interval representation of \( G \), and the analogous is true for arc orderings if \( G \) is non-co-bipartite.

### 5.1 Linking graphs and hypergraphs

**Proposition 5.1 (Roberts [36, 16]).** \( G \) is a proper interval graph iff \( \overline{N}(G) \) is an interval hypergraph.

A proof of this classical result can be found below in Section 5.4.

**Definition 5.2 (Bang-Jensen et al. [4]).** A graph \( G \) is called concave-round (resp. convex-round) if \( \overline{N}(G) \) (resp. \( N(G) \)) is a CA hypergraph.

Since \( \overline{N}(G) = N(\overline{G}) \), concave-round and convex-round graphs are co-classes. We adopt the terminology of [4] to state earlier results of Tucker.

**Proposition 5.3 (Tucker [39]).**

1. Proper CA graphs are concave-round.
2. Concave-round graphs are CA.

For the self-containedness, we will prove Proposition 5.3 in passing below. Part 1 will follow from Lemma 5.6. Part 2 will follow from Lemma 5.18 (for non-co-bipartite graphs) and from Proposition 5.4.1 and Theorem 7.1.1 (for co-bipartite graphs).

Proposition 5.3 justifies using the name Tucker circular-arc graphs synonymously with concave-round graphs. We will often use the abridged form TCA for this class.

Thus, the PCA graphs form an intermediate class between the classes of proper interval and TCA graphs. In the spirit of the characterizations of the latter two
Figure 5: A gallery of graph classes.

classes given by Proposition 5.1 and Definition 5.2, we can find accompanying hypergraphs also for PCA graphs. In Section 7 we will prove that $G$ is a PCA graph iff $\mathcal{N}[G]$ is a tight CA hypergraph; see Proposition 7.4. Note that tight interval hypergraphs do not give rise to any new class of graphs since, by Proposition 5.1 and Lemma 5.6, every interval neighborhood hypergraph is tight.

The following proposition shows that the class of graphs $G$ with $\mathcal{N}(G)$ being an interval hypergraph coincides with the known class of biconvex graphs, defined in Section 2.3.

**Proposition 5.4.**

1. **(ISGCI database [40])** A graph $G$ is biconvex iff it is bipartite convex-round.

2. **(implicitly in [39, Lemma 3])** A graph $G$ is biconvex iff $\mathcal{N}(G)$ is an interval hypergraph.

3. **(Bang-Jensen et al. [4, Lemma 3.11])** If $G$ is connected and $\mathcal{N}(G)$ admits an interval ordering, then each of the two vertex classes of $G$ is an interval w.r.t. this order.

**Proof.** 1. Part 1 follows from the definitions.

2. Suppose that $G$ is biconvex and this is certified by a partition $V(G) = U \cup W$. Since the open neighborhood of every vertex is completely contained either in $U$ or in $W$, the hypergraph $\mathcal{N}(G)$ is the vertex-disjoint union of its restrictions to $U$ and $W$, that is, $\mathcal{N}_U(G)$ and $\mathcal{N}_W(G)$. By definition, both restrictions are interval hypergraphs. Merging the interval orders on $U$ and $W$, we obtain an interval order for $\mathcal{N}(G)$.

For the backward direction, it suffices to show that any graph $G$ with interval neighborhood hypergraph is bipartite. Consider an interval order for $\mathcal{N}(G)$ and define a partition $V(G) = V^- \cup V^+$ such that $v \in V^-$ iff $v$ precedes the interval $N(v)$. If $v$ and $u$ are in $V^-$ and $v$ precedes $u$, then these vertices are not adjacent because $u$ precedes every vertex adjacent with it. Therefore, $V^-$ is independent. By a symmetric argument, $V^+$ is independent too.
3. By Part 2, \( G \) is connected bipartite. Denote its vertex classes by \( U \) and \( W \). With respect to the interval ordering of \( \mathcal{N}(G) \), both \( U \) and \( W \) are unions of intervals that are hyperedges of \( \mathcal{N}_U(G) \) and \( \mathcal{N}_W(G) \), respectively. As follows from the connectedness of \( G \), both hypergraphs \( \mathcal{N}_U(G) \) and \( \mathcal{N}_W(G) \) are connected. This implies that both \( U \) and \( W \) are intervals.

Propositions 5.4.1 and 5.3.2 imply that co-biconvex graphs are circular-arc. Tucker’s argument in [39] gives actually more: even co-convex graphs are CA; see Theorem 7.1.1.

Figure 5 summarizes relations between the classes of graphs mentioned in this section.

To study the class of TCA (i.e., concave-round) graphs, it is practical to split it in two parts, namely the graphs with bipartite complement and the graphs whose complement is non-bipartite.

**Proposition 5.5 (Tucker [39]).** Suppose that \( \overline{G} \) is not bipartite. Then \( G \) is TCA iff \( G \) is PCA.

Proposition 5.5 in one direction follows from Proposition 5.3.1. The other direction is proved by Tucker in [39] Lemma 5 and Theorem 6; see also Bang-Jensen et al. [4] Lemma 3.11. We recast Tucker’s argument in the proof of Lemma 5.18 below.

Assume now that \( \overline{G} \) is bipartite. By Proposition 5.4.1, \( G \) is then TCA iff \( \overline{G} \) is biconvex. For co-bipartite graphs Spinrad [37] Section 13.4.2] shows that, moreover, \( G \) is PCA iff \( \overline{G} \) is a bipartite permutation graph. Recall that permutation graphs are intersection graphs of families of segments connecting two parallel lines.

Figure 6 summarizes the structure of the classes of PCA and TCA graphs.

### 5.2 Geometric order: From a proper arc representation of \( G \) to a tight arc ordering of \( \mathcal{N}[G] \)

Any interval (resp. arc) representation \( \ell \) of a graph \( G \) induces an order structure on the vertex set of \( G \). We define a successor relation \( \prec_{\ell} \) on \( V(G) \) by setting \( u \prec_{\ell} v \) if \( \ell(v) \prec^* \ell(u) \), where \( \prec^* \) denotes the lexicographic order between intervals/arcs in the model of \( G \). Note that \( \ell \) is injective by definition. For arc representations
we suppose that no \( \ell(v) \) equals the whole circle (this is always true when \( G \) has no universal vertex). The relation \( \prec_{\ell} \) determines a linear (resp. circular) order on \( V(G) \), that will be referred to as the geometric order on \( V(G) \) associated with \( \ell \).

**Lemma 5.6.** The geometric order \( \prec_{\ell} \) induced on \( V(G) \) by a proper interval (resp. arc) representation \( \ell \) of \( G \) makes \( N^+[G] \) a tight interval (resp. arc) system.

**Proof.** We prove the lemma for proper arc representations; the proof for proper interval representations is quite similar and simpler. We begin with some notation. Let \( v \) be a non-universal vertex of \( G \) and \( \ell(v) = [a^-, a^+] \). Split \( N[v] \) in two parts, namely \( N^-[v] = \{ u \in N[v] : a^- \in \ell(u) \} \) and \( N^+[v] = \{ u \in N[v] : a^+ \in \ell(u) \} \). Obviously, both \( N^-[v] \) and \( N^+[v] \) are cliques. Note also that the intersection of \( N^-[v] \) and \( N^+[v] \) consists of the single vertex \( v \); if it contained another vertex \( v' \), then \( \ell(v) \) and \( \ell(v') \) would cover the whole circle, both intersecting any other arc \( \ell(w) \), contradictory to the assumption that \( v \) is non-universal.

First, we have to show that, for any vertex \( v \in V(G) \), its neighborhood \( N[v] \) is a segment of successive vertices w.r.t. the successor relation \( \prec_{\ell} \).

Suppose that \( v \) is non-universal for else the claim is trivial. Assume that \( u \in N^+[v] \) and \( v \prec_{\ell} u_1 \prec_{\ell} \ldots \prec_{\ell} u_k \prec_{\ell} u \). Then everyone of \( i \) is in \( N^+[v] \). Indeed, by the definition of \( \prec_{\ell} \), we have \( \ell(v) \prec^* \ell(u_1) \prec^* \ldots \prec^* \ell(u_k) \prec^* \ell(u) \). If \( \ell(u) = [b^-, b^+] \) and \( \ell(u_i) = [c^-, c^+] \), we see that \( c^- \in (a^-, b^-) \), \( c^+ \in (a^+, b^+) \) and, hence, \( a^+ \in [c^-, c^+] \). It follows that \( N^+[v] \) is a segment of successive vertices starting at \( v \). By a symmetric argument, \( N^-[v] \) is a segment of successive vertices finishing at \( v \). Merging the two segments at \( v \), we prove the claim for \( N[v] \).

It remains to show that the arc system \( N^+[G] \) on the circle \( (V(G), \prec_{\ell}) \) is tight. Suppose that \( N[u] \subseteq N[v] \) and \( v \) is non-universal. Given a vertex \( w \), we denote the start point of \( N[w] \) by \( w^- \) and the end point of \( N[w] \) by \( w^+ \). Assume that \( u \in (v, v^+) \). Since \( [v, v^+] \) is a clique, \( u \) and \( v^+ \) are adjacent and, therefore, \( u^+ = v^+ \). If \( u \in (v^-, v) \), the symmetric argument shows that \( u^- = v^- \).

In what follows we keep using the notation \( N[w] = [w^-, w^+] \), \( N^-[w] = [w^-, w] \), and \( N^+[w] = [w, w^+] \), introduced in the proof of Lemma 5.6 w.r.t. the order \( \prec_{\ell} \).

**Lemma 5.7.** Let \( \ell \) be a proper arc representation of a graph \( G \). Suppose that \( G \) has no universal vertex.

1. For any two vertices \( x \) and \( y \) we have
   \[
   y \in N^+[x] \text{ iff } x \in N^-[y].
   \] (5.1)

2. If the two conditions in (5.1) are true, then
   \[
   y^- \in N^-[x] \text{ and } x^+ \in N^+[y].
   \] (5.2)

3. If, in addition, \( x \prec_{\ell} y \), then \( x^- \), \( y^- \), \( x \), \( y \), \( x^+ \), and \( y^+ \) occur under the geometric order exactly in this circular sequence, where some of the neighboring vertices except \( x^- \) and \( y^+ \) may coincide.
4. Moreover, in the last case we cannot have \( y^+ \prec_{\ell} x^- \) unless \( G \) is co-bipartite.

**Proof.** 1. Let \( x \neq y \). Assume that \( y \in N^+[x] \). If \( x \in N^+[y] \), then \( V(G) \) would be covered by two cliques \( N^+[x] \) and \( N^+[y] \), both containing \( x \). Then we would have \( N[x] = V(G) \) contradictory to the assumption that there is no universal vertex. Therefore, \( x \in N^-[y] \). The backward implication in (5.1) follows by the symmetric argument.

2. If the two conditions in (5.1) are true, then \( y^- \), \( x \), and \( y \) occur in this circular order. Since \( [y^-, y] \) is a clique, all vertices in \([y^-, x]\) are adjacent to \( x \) and, hence, \( y^- \in N^-[x] \). The second containment in (5.2) follows by the symmetric argument.

3. The containments in (5.1) and in (5.2) imply that \( x^- \), \( y^- \), \( x \), \( y \), \( x^+ \), and \( y^+ \) occur in this circular order, though \( y^+ \) and \( x^- \) can perhaps be swapped or can coincide. To show that the condition \( x \prec_{\ell} y \) rules out the last two possibilities, assume for a while that one of them is the case and consider a vertex \( z \) in \([x^-, y^+]\), for example, \( z = x^- \). Then \( z \in N^-[x] \) and \( z \in N^+[y] \). Applying the equivalence (5.1) to the pairs \((x,y) := (z,x)\) and \((x,y) := (y,z)\), we see that \([z,x] \subseteq N^+[z] \) and \([y,z] \subseteq N^-[z] \). Thus, \( z \) should be universal, and this contradiction completes the proof of the claim. All possible mutual positions of \( N[x] \) and \( N[y] \) are shown in Fig. 7(a).

4. If \( x \prec_{\ell} y \) and \( y^+ \prec_{\ell} x^- \), then \( V(G) \) is split into two cliques \([x^-, x]\) and \([y,y^+]\).

The following lemma will be needed in Section 8.

**Lemma 5.8.**

1. Let \( \ell \) be a proper interval representation of a twin-free graph \( G \). Then \( v \prec_{\ell} u \) exactly when \( N[v] \prec_{\ell}^* N[u] \), where \( \prec_{\ell}^* \) denotes the lexicographic order on \( N[G] \) with respect to \( \prec_{\ell} \).

2. If \( \ell \) is a proper arc representation of a twin-free graph \( G \) with no universal vertex, then \( v \prec_{\ell} u \) exactly when \( N[v] \prec_{\ell}^* N[u] \).
Proof. We prove the lemma for proper arc representations; the proof for proper interval representations is quite similar and simpler. Note in this respect that Lemma 5.7.1–3 holds true for interval representations, even without the assumption that there is no universal vertex: Part 1 is completely obvious, the proof of Part 2 goes through, the interval analog of Part 3 trivially follows from Part 2.

It suffices to assume that \( v \prec_{\ell} u \) and derive from here that \( N[v] \prec^*_{\ell} N[u] \). This will be done once we show that, for any third vertex \( w \), the arcs \( N[v], N[w], \) and \( N[u] \) cannot appear in this sequence under the circular order determined by \( \prec^*_{\ell} \). Suppose first that \( v \) and \( u \) are adjacent.

Applying Lemma 5.7.3 to the pair \( (x, y) := (v, u) \), we see that the vertices \( v^- \), \( u^{-} \), \( v \), \( u \), \( v^+ \), and \( u^+ \) go in this circular order, as shown in Fig. 7(b). We split our analysis into three cases, depending on the position of \( w \) in the circle \( V(G) \).

If \( w \in (u, u^+] \), then by Lemma 5.7.2 for the pair \( (x, y) := (u, w) \) we have \( w^- \in [u^{-}, v^-] \). If \( w^- \neq u^- \), then \( N[v], N[u], \) and \( N[w] \) appear in this circular order. The same holds true if \( w^- = u^- \) because then, by Lemma 5.7.2, the arc \( [w^- , u^+] \) is longer than the arc \( [u^- , u^+] \). The case of \( w \in (v^-, v^-) \) is symmetric. If \( w \in (u^+, v^-) \), then \( w^- \in (u, v^-) \) and, again, \( N[w] \) cannot be intermediate.

Suppose now that \( v \) and \( u \) are not adjacent. In the arc model of \( G \), the arcs \( \ell(v) \) and \( \ell(u) \) are disjoint, and no arc \( \ell(x) \) lies after \( \ell(v) \) before \( \ell(u) \) in the clockwise direction. It follows that \( N[v] = [v^-, v] \) and \( N[u] = [u, u^+] \). Note also that both neighborhoods are cliques.

As above, we have to show that, for no third vertex \( w \), the arc \( N[w] \) lies after \( N[v] \) before \( N[u] \) in the clockwise direction w.r.t. the lexicographic order in the circle \( (V(G), \prec_{\ell}) \). This is clear if \( w^- \in (u^-, v^-) = (u, v^-) \). This is also true if \( w^- = u^- (= u) \) because then the arc \( [u^- , u^+] \), being a clique in \( G \), must be shorter than the arc \( [w^- , u^+] \).

The remaining case of \( w^- \in [v^-, u^-] = [v^-, v] \) is actually impossible. Indeed, in this case \( u \notin N[w] \) for else non-adjacent vertices \( u \) and \( v \) would belong to the clique \( \{w, w^+\} \). Therefore, we should have the inclusion \( [w^- , u^+] \subset [v^-, v^+] = N[v] \), contradicting the fact that \( N[v] \) is a clique. \( \blacksquare \)

Lemma 5.9. If \( G \) is a connected PCA graph with non-bipartite complement \( \overline{G} \), then \( N'[G] \) is strictly connected.

Proof. Let \( \ell \) be a proper arc representation of \( G \). By Lemma 5.6 \( N'[G] \) is an arc system w.r.t. the geometric order \( \prec_{\ell} \). No vertex \( u \) of \( G \) is universal because otherwise \( V(G) \) would be covered by two cliques \( N^-[u] \) and \( N^+[u] \). Since \( G \) is connected, there is at most one pair of non-adjacent vertices \( x \) and \( y \) satisfying the relation \( x \prec_{\ell} y \). Therefore, all vertices of \( G \) can be arranged into a path \( v_1, \ldots, v_n \) such that \( v_i \) and \( v_{i+1} \) are adjacent and \( v_i \prec_{\ell} v_{i+1} \) for every \( 1 \leq i < n \). By Lemma 5.7.3–4, we have \( N[v_i] \triangleright^a N[v_{i+1}] \), which gives us a strictly connected path passing through all hyperedges of \( N'[G] \). \( \blacksquare \)

Lemma 5.10.
1. If $G$ is a connected proper interval graph, then $\mathcal{N}[G]$ has, up to congruence and permutation of twins, a unique tight interval representation.

2. If $G$ is a connected PCA graph with non-bipartite complement, then $\mathcal{N}[G]$ has, up to congruence and permutation of twins, a unique tight arc representation.

Proof. 1. By Lemma 5.6, $\mathcal{N}[G]$ has at least one, namely geometric, tight interval ordering. Since the graph $G$ is connected, so is the hypergraph $\mathcal{N}[G]$. By Lemma 4.2.1, a tight interval representation of $\mathcal{N}[G]$ is unique up to congruence and permutation of twins (or, equivalently, a tight interval ordering of $\mathcal{N}[G]$ is unique up to reversal and permutation of twins).

2. The proof goes quite similarly. In order to apply Lemma 4.2.2, we need $\mathcal{N}[G]$ be strictly connected. This is true by Lemma 5.9.

5.3 Non-co-bipartite TCA graphs: any arc ordering of $\mathcal{N}[G]$ is tight

In the preceding subsection we established some useful properties of a geometric order on graph vertices, in particular, for any proper interval and non-co-bipartite PCA graph $G$. Here we will see that all these properties hold true actually for any interval/arc ordering of $\mathcal{N}[G]$ if $G$ is in one of the two aforementioned classes, and we will conclude from here that, in fact, any ordering of $\mathcal{N}[G]$ is geometric.

Suppose that we are given an interval ordering of the hypergraph $\mathcal{N}[G]$. Then, for any $v \in V(G)$, the closed neighborhood $N[v]$ can be regarded as an interval of linearly ordered vertices. Like in the preceding subsection, we let $N^-[v] = [v^-, v]$ denote the part of this interval from the start point $v^-$ up to $v$ and $N^+[v] = [v, v^+]$ denote the part from $v$ up to the end point $v^+$. This notation makes sense also if we have an arc ordering, unless the vertex $v$ is universal.

Lemma 5.11. A TCA graph $G$ with non-bipartite $\overline{G}$ has no universal vertex.

Proof. Assume to the contrary that there is a universal vertex $u$. Split the set of non-universal vertices of $G$ into two parts. Let $V^+$ consist of those vertices $v$ for which $u \in N^+[v]$ and $V^-$ of those $v$ for which $u \in N^-[v]$. For arbitrary two vertices $v_1$ and $v_2$ in $V^+$, either $v_1 \in N^+[v_2]$ or $v_2 \in N^+[v_1]$. In either case $v_1$ and $v_2$ are adjacent, which implies that $V^+$ is a clique. By similar reasons $V^-$ is a clique too. Adding the universal vertices to $V^+$ or $V^-$, we obtain a cover of $V(G)$ by two cliques. This contradicts the assumption that $\overline{G}$ is not bipartite.

The following observation applies to proper interval graphs and generalizes Lemma 5.7.1. Part 2 of the next lemma was obvious for geometric interval orders.

Lemma 5.12. Suppose that $\mathcal{N}[G]$ admits interval orderings. For any of them, the following two statements are true.

1. $u \in N^-[v]$ iff $v \in N^+[u]$. 
2. For any vertex $v$ of $G$, both $N^-[v]$ and $N^+[v]$ are cliques.

**Proof.** 1. Both of the conditions tell that $u$ is left of $v$ with respect to the given linear order on $V(G)$.

2. This follows from Part 1. Let us show, for example, that any two vertices $u$ and $u'$ in $N^-[v] \setminus \{v\}$ are adjacent. Without loss of generality, suppose that $u'$ lies between $u$ and $v$, that is, $u' \in (u, v)$. By Part 1, $v \in N^+[u]$. It follows, that $N[u]$ includes $[u, v]$ and, hence, contains $u'$.

Lemma 5.12 extends to TCA graphs provided that they have non-bipartite complement.

**Lemma 5.13.** Let $\overline{G}$ be non-bipartite. Suppose that $N[G]$ admits arc orderings. For any of them, the following two statements are true.

1. $u \in N^-[v]$ iff $v \in N^+[u]$.

2. For any vertex $v$ of $G$, both $N^-[v]$ and $N^+[v]$ are cliques.

**Proof.** Part 2 follows from Part 1 exactly as in the proof of Lemma 5.12. The proof of Part 1 requires more care as two vertices $u$ and $v$ can now be joined by an arc in two ways, clockwise and counter-clockwise. Call a vertex $u \in N[v]$ a close neighbor of the vertex $v$ if

$$u \in N^-[v] \text{ and } v \in N^+[u] \text{ or } u \in N^+[v] \text{ and } v \in N^-[u]$$

and a distant neighbor otherwise. Note that $v$ is a close neighbor of itself and that $u \neq v$ is a distant neighbor of $v$ iff

$$u \in N^-[v] \text{ and } v \in N^-[u] \text{ or } u \in N^+[v] \text{ and } v \in N^+[u].$$

Denote the sets of close neighbors of $v$ in $N^-[v]$ and $N^+[v]$ by $C^-[v]$ and $C^+[v]$ respectively. The sets of distant neighbors of $v$ in $N^-[v]$ and $N^+[v]$ will be denoted, respectively, by $D^-[v]$ and $D^+[v]$. Each of the four sets $C^-[v]$, $C^+[v]$, $D^-[v]$, and $D^+[v]$ is a clique. The proof of this fact for $C^-[v]$ and $C^+[v]$ is the same as the proof of Part 2 of Lemma 5.12. For $D^-[v]$ and $D^+[v]$ we can argue quite similarly. For example, if $u$ and $u'$ are two vertices in $D^+(v)$, then $v$ belongs to both $N^+[u]$ and $N^+[u']$; now the argument used for Part 2 of Lemma 5.12 goes through.

Note that Part 1 of the lemma actually states that $D^-[v] = D^+[v] = \emptyset$ for every $v$. Assume to the contrary that $D^+[v]$ for some $v$ is not empty and contains a vertex $u$; see Fig. 8 (the assumption $D^-[v] \neq \emptyset$ can be treated similarly). Since $v \in N^+[u]$, the sets $N^+[v] \cap [v, u]$ and $N^+[u] \cap [u, v]$ cover $V(G)$. Splitting the former set into $C^+[v] \cap [v, u]$ and $D^+[v] \cap [v, u]$ and the latter into $C^+[u] \cap [u, v]$ and $D^+[u] \cap [u, v]$, consider the cover of $V(G)$ by two sets $(C^+[v] \cap [v, u]) \cup (D^+[u] \cap [u, v])$ and $(D^+[v] \cap [v, u]) \cup (C^+[u] \cap [u, v])$ and show that they are cliques. This will give us a contradiction with the assumption that $\overline{G}$ is not bipartite. By symmetry, it suffices to prove that $(D^+[v] \cap [v, u]) \cup (C^+[u] \cap [u, v])$ is a clique. Since both $D^+[v]$ and $C^+[u]$ are cliques, we have to show that any vertex $w$ in $D^+[v] \cap [v, u]$ is adjacent to all vertices in $C^+[u] \cap [u, v]$. This is true because we have $N[w] \supseteq [w, v] \supseteq [u, v]$ by the definition of $D^+[v]$.
Parts 1 and 2 of the following result generalize Lemma 5.6.

Lemma 5.14.

1. Any interval ordering of $\mathcal{N}[G]$ makes it a tight interval system.

2. If $\overline{G}$ is non-bipartite, then any arc ordering of $\mathcal{N}[G]$ makes it a tight arc system.

3. In both cases, if $N[u] \subseteq N[v]$, then $u^{-} = v^{-}$ whenever $u \in N^{-}[v]$ and $u^{+} = v^{+}$ whenever $u \in N^{+}[v]$.

Proof. Consider first interval orderings. Let $N[u] \subseteq N[v]$. Suppose that $u \in N^{+}[v]$ (the other case is symmetric). By Lemma 5.12.2, $N^{+}[v] = [v, v^{+}]$ is a clique. Therefore, this interval is contained in $N[u]$, which implies that $u^{+} = v^{+}$.

For arc orderings, we have to use Lemma 5.13.2 instead of Lemma 5.12.2. \hfill \blacksquare

Now we can strengthen Lemma 5.10.

Lemma 5.15.

1. If $G$ is a connected proper interval graph, then $\mathcal{N}[G]$ has, up to congruence and permutation of twins, a unique interval representation.

2. If $G$ is a connected PCA graph with non-bipartite complement, then $\mathcal{N}[G]$ has, up to congruence and permutation of twins, a unique arc representation.

Proof. Note that $\mathcal{N}[G]$ has at least one, namely geometric, interval/arc ordering; see Lemma 5.6. The uniqueness follows from Lemmas 5.14 and 5.10. \hfill \blacksquare

Lemma 5.15 implies that, if $G$ is proper interval or non-co-bipartite PCA, then $\mathcal{N}[G]$ has no other interval/arc ordering than geometric. It is also related to the concepts of straight and round orientations of a graph studied in [15]. An orientation of a graph $G$ is straight (resp. round) if there is an interval (resp. arc) ordering of $\mathcal{N}[G]$ such that each $N^{+}[v]$ consists of the vertex $v$ and its out-neighbors and each $N^{-}[v]$ consists of $v$ and its in-neighbors. [15] Corollary 2.5 says that a twin-free
connected proper interval graph has, up to reversal, a unique straight orientation, and \cite{15} Corollary 2.9 extends this uniqueness result to round orientations of twin-free connected PCA graphs with non-bipartite complement. In fact, these results and our Lemma 5.15 in the twin-free case are equivalent. More precisely, one can show that there is a one-to-one correspondence between interval orderings of $\mathcal{N}[G]$ and straight orientations of a twin-free connected proper interval graph $G$ as well as between arc orderings of $\mathcal{N}[G]$ and round orientations of a twin-free connected non-co-bipartite PCA graph $G$.

5.4 From an arc ordering of $\mathcal{N}[G]$ (back) to a proper arc representation of $G$

In Lemma 5.6 we observed that any proper interval/arc representation of a graph $G$ determines an interval/arc ordering of $\mathcal{N}[G]$. Now our aim is to show that, vice versa, any interval/arc ordering of $\mathcal{N}[G]$ determines a proper interval/arc representation of $G$ (in the case of arc orderings we assume that $G$ is non-co-bipartite). Note first that it suffices to achieve a more modest goal, namely to construct a tight arc representation of the graph.

Lemma 5.16.

1. Any tight interval representation $\ell$ of a graph $G$ can be converted in logarithmic space to a proper interval representation $\ell'$ of $G$.

2. Let $\ell$ be a tight arc representation of a graph $G$ where no arc equals the whole circle. Then $\ell$ can be converted in logarithmic space to a proper arc representation $\ell'$ of $G$.

This has been observed by Tucker \cite{39}, and Chen \cite{8,10} showed that the conversion can be implemented in $\text{AC}^1$. We show that it is even possible in logspace.

Proof. We will prove Part 2; the proof of Part 1 is similar and simpler.

Suppose that $\ell$ is defined on a circle $C$. The new representation $\ell'$ will be defined on a circle $C'$ obtained from $C$ by replacing each point $x$ with a segment of new points. Specifically, if $x$ occurs in the arc model of $G$ $k$ times as the start point of an arc and $m$ times as the end point, then it will be replaced with the segment of points $x_1\ldots,x_k,x_1^+\ldots,x_m^+$, exactly in this clockwise order. Let $v_1,\ldots,v_k$ be the vertices such that $\ell(v_i)$ starts at $x$, listed according to the lexicographic order of the arcs $\ell(v_i)$, that is, $|\ell(v_i)| < |\ell(v_{i+1})|$. Then the arc $\ell'(v_i)$ starts at the point $x_i^-$. Furthermore, let $u_1,\ldots,u_m$ be the vertices such that $\ell(u_i)$ ends at $x$, listed according to the lexicographic order of the arcs $\ell(u_i)$, that is, $|\ell(u_i)| > |\ell(u_{i+1})|$. Then the arc $\ell'(u_i)$ finishes at the point $x_i^+$. This defines a mapping $\ell'$ of vertices of $G$ to arcs of $C'$. Note that, if $\ell(v) = \{x,y_1,\ldots,y_s,z\}$, then $\ell'(v)$ consists of all clones of each inner point $y_i$, a segment of clones of the start point $x$, and a segment of clones of the end point $y$. In particular, if $\ell(v) = [x,x]$, then $\ell'(v) = [x_1^-,x_m^+]$. Note also that

\footnote{or connected}
\( \ell' \) is injective and \( \ell'(v) \neq \emptyset \) for any \( v \). It remains to prove the claimed properties of \( \ell' \).

\( \ell' \) is a representation of \( G \). Indeed, if \( \ell(v) \) and \( \ell(u) \) do not intersect, then \( \ell'(v) \) and \( \ell'(u) \) do not intersect either. If \( \ell(v) \) and \( \ell(u) \) intersect and the intersection contains a point \( x \) inner in one of the arcs \( \ell(v) \) and \( \ell(u) \), then one of the arcs \( \ell'(v) \) and \( \ell'(u) \) contains all the clones of \( x \) while the other contains at least one clone of \( x \). Assume now that \( \ell(v) \) and \( \ell(u) \) share an extreme point \( x \). We can suppose that \( \ell(u) \) ends at \( x \) and \( \ell(v) \) starts at \( x \) for else \( \ell(v) \cap \ell(u) \) would contain an inner point of one of the arcs. Then \( \ell'(v) \) and \( \ell'(u) \) intersect because \( \ell'(u) \) contains all the points \( x_i^- \) and some of the points \( x_j^+ \) while \( \ell'(v) \) contains all the points \( x_j^+ \) and some of the points \( x_i^- \). Thus, \( \ell'(v) \succ \succ \ell'(u) \) iff \( \ell(v) \succ \succ \ell(u) \) and, therefore, \( \ell' \) is a representation of \( G \) as well as \( \ell \).

\( \ell' \) is proper. If \( \ell(u) \not\subset \ell(v) \), then \( \ell'(u) \not\subset \ell'(v) \). Indeed, if \( x \in \ell(u) \setminus \ell(v) \), then \( \ell'(u) \) contains some clones of \( x \) while \( \ell'(v) \) contains none of them. Let \( \ell(u) \subset \ell(v) \). Suppose that \( \ell(v) \) and \( \ell(u) \) share a start point \( x \). Note that \( |\ell(u)| < |\ell(v)| \) and, hence, \( \ell(u) \) lexicographically precedes \( \ell(v) \) in the segment of arcs starting at \( x \). By construction, \( \ell'(u) \) starts at some \( x_i^- \) and \( \ell'(v) \) starts at some \( x_j^- \), where \( i < j \). Therefore, \( x_i^- \in \ell'(u) \setminus \ell'(v) \) and \( \ell'(u) \not\subset \ell'(v) \). The case of \( \ell(v) \) and \( \ell(u) \) sharing an end point is symmetric.

**Lemma 5.17.** Given an interval ordering of \( N[G] \), one can in logspace construct a proper interval representation of \( G \).

**Proof.** Given an interval ordering \( \prec \) of \( N[G] \), for each vertex \( v \) of \( G \) we define \( \ell(v) = N^+[v] \) (which is an interval w.r.t. \( \prec \)). The map \( \ell \) is injective. Note that this is an interval representation of \( G \). Indeed, suppose that \( u \) and \( v \) are adjacent. By Lemma 5.12.1, either \( u \in N^+[v] \) or \( v \in N^+[u] \). In either case \( N^+[u] \cap N^+[v] \neq \emptyset \). If \( u \) and \( v \) are not adjacent, \( u \notin N^+[v] \) and \( v \notin N^+[u] \), which implies that the intervals \( N^+[u] \) and \( N^+[v] \) are disjoint.

The representation \( \ell \) is tight because, if \( N^+[u] \subset N^+[v] \), then \( u^+ = v^+ \) by Lemma 5.14.3.

Constructing the representation \( \ell \) is not much harder than constructing the neighborhood hypergraph \( N[G] \) for a given graph \( G \) and can be done in logspace. The algorithm of Lemma 5.16.1 transforms \( \ell \) in logspace into a proper representation.

Note that, together with Lemma 5.6, Lemma 5.17 proves the Roberts theorem stated above as Proposition 5.1.

We now prove an analog of Lemma 5.17 for TCA graphs with non-bipartite complement.

**Lemma 5.18.** Suppose that a graph \( G \) has non-bipartite complement. Given an arc ordering of \( N[G] \), one can in logspace construct a proper arc representation of \( G \).

**Proof.** The proof is virtually the same as the proof of Lemma 5.17. We only have to use Lemma 5.13.1 instead of Lemma 5.12.1, and Part 2 of Lemma 5.16 instead
of its Part 1. We also have to note that the tight arc representation \( \ell(v) = N^+[v] \) constructed in the proof has the property that \( \ell(v) \neq V(G) \) for any \( v \) (otherwise \( v \) would be a universal vertex, which is impossible by Lemma 5.1).

Together with Lemma 5.6, Lemma 5.18 proves Tucker’s results stated above as Proposition 5.5.

6 Canonical arc representations of non-co-bipartite graphs

We now turn to the canonical representation problem for TCA and PCA graphs. In the most general setting, for a given CA graph we have to compute its arc representation such that the resulting arc models are equal for isomorphic input graphs. This problem is similarly defined for interval graphs. If considered for proper interval or PCA graphs, the canonical representation problem assumes computing proper interval or proper arc representations.

Theorem 6.1. *The canonical arc representation problem for TCA graphs is solvable by a logspace algorithm. Moreover, the algorithm produces a proper arc representation whenever an input graph is PCA.*

The proof of Theorem 6.1 takes this and the next sections. In this section we prove it in the case that the complement \( \overline{G} \) of an input graph is not bipartite. As shown by Tucker (see Proposition 5.5), the classes of TCA and PCA graphs coincide under this assumption, and our goal is to construct a canonical proper arc representation of \( G \).

For any class of intersection graphs, a canonical representation algorithm readily implies a canonical labeling algorithm of the same complexity. Vice versa, a canonical representation algorithm readily follows from a canonical labeling algorithm and an algorithm representing a graph by an intersection model (not necessarily canonical). We first compute an isomorphism \( \alpha \) from an input graph \( G \) to its canonical form \( G^\ast \), then compute a representation \( \ell \) of \( G^\ast \) by an intersection model, and return the composition \( \ell \circ \alpha \) as a canonical representation of \( G \). Following this scheme, we split our task in two parts. Given a non-co-bipartite PCA graph \( G \), we will compute in logspace, first, a proper arc representation of \( G \) and, second, a canonical labeling of \( G \).

**Proper arc representation algorithm.** Since the graph \( G \) is PCA, the hypergraph \( \mathcal{N}[G] \) is CA. The algorithm of Theorem 3.2 computes an arc ordering of \( \mathcal{N}[G] \), and the algorithm of Lemma 5.18 converts it into a proper arc representation of \( G \).

**Canonical labeling algorithm.** Suppose first that \( G \) is a proper interval graph (here we will not use the assumption that \( \overline{G} \) is non-bipartite). This case is recognizable in logspace by the algorithm of Theorem 3.1 because it is equivalent to
the condition that $\mathcal{N}[G]$ is an interval hypergraph; see Proposition 5.1. A canonical labeling algorithm for proper interval graphs follows from the canonical interval representation algorithm designed for this class in [26]. Nevertheless, here we give an independent treatment because all the same ideas will work also in the pure PCA (non-co-bipartite) case.

Moreover, we will suppose that $G$ is connected. The general case reduces to the connected case: Given a disconnected graph, split it into connected components by Reingold’s algorithm [35], compute the canonical labeling for each of them, sort the canonical forms lexicographically, and merge the canonical labelings of connected components according to this order.

As usually, denote the number of vertices in $G$ by $n$. We compute a canonical labeling $\lambda_G: V(G) \rightarrow [1, n]$ as follows. Run the algorithm of Theorem 3.1 to get an interval representation $\lambda_1: V(G) \rightarrow [1, n]$ of the interval hypergraph $\mathcal{N}[G]$. Let $\lambda_2$ be the mirror-symmetric version of $\lambda_1$. Compute two graphs $\lambda_1(G)$ and $\lambda_2(G)$ on the vertex set $[1, n]$. If the adjacency matrix of $\lambda_1(G)$ is lexicographically smaller than the adjacency matrix of $\lambda_2(G)$, output $\lambda_G = \lambda_1$; otherwise output $\lambda_G = \lambda_2$.

Let $G^* = \lambda_G(G)$ denote the corresponding canonical form of $G$. We have to prove that $G^* = H^*$ whenever $G \cong H$.

By Lemma 5.15, any interval representation $\lambda$ of $\mathcal{N}[G]$ is equivalent to $\lambda_1$ or to $\lambda_2$ up to permutation of twins. Since any permutation of twins is an automorphism of $G$, we have $\lambda(G) = \lambda'(G)$ whenever $\lambda$ and $\lambda'$ are equivalent up to permutation of twins. It follows that

$$G^* = \min_\lambda \{ \lambda(G) \},$$

where the minimum is taken over all interval representations $\lambda$ of the hypergraph $\mathcal{N}[G]$ w.r.t. the lexicographic order between adjacency matrices of graphs $\lambda(G)$.

Let $\phi$ be an isomorphism from $H$ to $G$, and hence from $\mathcal{N}[H]$ to $\mathcal{N}[G]$. Then (6.1) reads

$$G^* = \min_\lambda \{ \lambda \circ \phi(H) \},$$

where $\lambda \circ \phi$ is an interval representation of $\mathcal{N}[H]$ as $\lambda$ ranges over interval representations of $\mathcal{N}[G]$. Since any interval representation $\mu$ of $\mathcal{N}[H]$ is representable in the form $\mu = (\mu \circ \phi^{-1}) \circ \phi$, where $\mu \circ \phi^{-1}$ is an interval representation of $\mathcal{N}[G]$, Equality (6.2) implies that

$$G^* = \min_\mu \{ \mu(H) \} = H^*,$$

where $\mu$ ranges over all interval representations of $\mathcal{N}[G]$. The correctness of the algorithm is proved.

Suppose now that $G$ is a connected PCA graph with non-bipartite complement (disconnected PCA graphs are proper interval, where we are already done). We proceed quite similarly. Running the algorithm of Theorem 3.2, we obtain an arc representation $\lambda_1: V(G) \rightarrow \mathbb{Z}_n$ of the arc hypergraph $\mathcal{N}[G]$. Now we have to consider $n$ rotated versions $\lambda_1, \lambda_2, \ldots, \lambda_n$ of this representation and their $n$ reversals $\lambda_1', \lambda_2', \ldots, \lambda_n'$. After computing all graphs $\lambda_1(G)$ and $\lambda'_1(G)$ on the vertex set $\mathbb{Z}_n$ and comparing lexicographically their adjacency matrices, the algorithm returns as $\lambda_G$ that of the $2n$ arc representations of $G$ which attains the minimum (or any such
representation if there are more than one). The proof of the fact that isomorphic graphs receive equal canonical forms is now based on Lemma [5.15] and is essentially the same.

The proof of Theorem 6.1 for non-co-bipartite graphs is complete.

7 Canonical arc representations of co-bipartite graphs

We have just proved Theorem 6.1 for non-co-bipartite TCA graphs. Now we treat the co-bipartite case. Our first goal is modest: we want to construct a canonical arc representation of a TCA graph \( G \) with bipartite complement, not necessarily proper even if \( G \) is PCA. This task can be efficiently fulfilled even for a larger class of graphs. Recall that TCA graphs with bipartite complement are precisely complements of biconvex graphs (see Proposition 5.4.1) and that biconvex graphs form a proper subclass of convex graphs.

Theorem 7.1.

1. Co-convex graphs are CA.

2. The canonical arc representation problem for co-convex graphs is solvable in logarithmic space.

Proof. 1. Given a convex graph \( H \), we have to show that its complement \( G = \overline{H} \) is a CA graph. Though not stated explicitly, this fact is actually proved in Tucker [39], and we essentially follow Tucker’s argument. It suffices to consider the case when \( H \) has no fraternal vertices (those would correspond to twins in \( G \)).

Let \( V(H) = U \cup W \) be a partition of \( H \) into independent sets such that \( N_U(H) \) is an interval hypergraph. Let

\[ U = \{ u_1, \ldots, u_k \} \tag{7.1} \]

be an interval ordering for \( N_U(H) \). We construct an arc representation \( \ell \) for \( G \) on the circle \( \mathbb{Z}_{2k+2} \) by setting \( \ell(u_i) = [i, i+k] \) for each \( u_i \in U \) and \( \ell(w) = [j+k+1, i-1] \) for each \( w \in W \), where \( N_H(w) = [u_i, u_j] \). Note that \( \ell(w) = \mathbb{Z}_{2k+2} \setminus \bigcup_{u \in N_H(w)} \ell(u) \).

In the case that \( N_H(w) = \emptyset \), we set \( \ell(w) = [0, k] \). By construction, all arcs \( \ell(u) \) for \( u \in U \) share a point (even two, \( k \) and \( k+1 \)), the same holds true for all \( \ell(w) \) for \( w \in W \) (they share 0), and any pair \( \ell(u) \) and \( \ell(w) \) is intersecting iff \( u \) and \( w \) are adjacent in \( G \). Thus, \( G \) is CA.

2. The proof of Part 1 is constructive; it readily gives us a logspace algorithm that, given a twin-free co-convex graph \( G \), a partition \( \{U, W\} \) of the complementing graph \( H \), and an interval ordering (7.1) of the hypergraph \( N_U(H) \), constructs an arc representation of \( G \). Note that the preprocessing step, that is, finding \( \{U, W\} \) such that \( N_U(H) \) is an interval hypergraph and computing an interval ordering of \( N_U(H) \), can be done also in logspace. To this end, we first use the Reingold algorithm [35] to split \( H \) into connected components \( H_1, \ldots, H_m \), each with bipartition

\[ \text{The subscript } H \text{ means that the vertex neighborhood is considered in this graph.} \]
$V(H_i) = U_i \cup W_i$. Then we run the algorithm of Theorem 3.1 on each of the hypergraphs $\mathcal{N}_{U_i}(H_i)$ and $\mathcal{N}_{W_i}(H_i)$, computing interval orderings for all those hypergraphs which are interval. Swapping the notation $U_i$ and $W_i$ whenever needed, we obtain interval orderings for all $\mathcal{N}_{U_i}(H_i)$ and merge them, obtaining an interval ordering of $\mathcal{N}_U(H)$, where $U = \bigcup_{i=1}^m U_i$. Thus, we have a logspace algorithm constructing an arc representation of a given twin-free co-convex graph.

A canonical labeling algorithm for convex graphs, hence for all co-convex graphs, is designed in [26] (we can as well use the algorithm of Theorem 3.3 for the larger class of circular convex graphs). The canonical arc representation problem for co-convex graphs is, therefore, solvable in logspace by the forthcoming Lemma 7.2. ■

As noticed in Section 6 we have a canonical representation algorithm for a class of intersection graphs whenever this class admits a canonical labeling algorithm and an algorithm representing a given graph by an intersection model (not necessarily canonical). We now observe that, for CA and PCA graphs, it is enough to have the latter algorithm only in the twin-free case.

Lemma 7.2. Let $C$ be a class of circular-arc graphs admitting a logspace canonical labeling algorithm. Denote the class of twin-free graphs in $C$ by $C_1$. If $C_1$ has a logspace arc representation algorithm (not necessarily canonical), then $C$ has a logspace canonical arc representation algorithm. Moreover, if the former algorithm produces proper arc representations, so does the latter.

Proof. Let $G \in C$ be an input graph and $\alpha$ be the canonical isomorphism from $G$ to its canonical form $G^\ast$. By assumption, $\alpha$ is computable in logspace. Let $G_1^\ast$ denote the quotient graph of $G^\ast$ and $\tau: V(G^\ast) \to V(G_1^\ast)$ denote the quotient map taking any vertex of $G^\ast$ to its twin-class. Since $G_1^\ast$ is twin-free, we can compute an arc representation $\ell_1$ of this graph in logarithmic space. Suppose that $\ell_1$ maps vertices of $G_1^\ast$ into arcs of the circle $\mathbb{Z}_n$. Denote the corresponding arc model of $G_1^\ast$ by $A_1$.

At the first step, we transform $\ell_1$ to another arc representation $\ell_1': V(G_1^\ast) \to A_1'$ on the circle $\mathbb{Z}_{2n}$. We do it by doubling each point: if the arc in $A_1$ contains a point $x$, then the corresponding arc in $A_1'$ will contain two points $2x$ and $2x + 1$. Note that $\ell_1'$ remains an arc representation of $G_1^\ast$ and is proper if $\ell_1$ is proper.

At the next step, we construct an arc system $A$ on the continuous circle of length $2n$ as follows. If a vertex $w$ of $G^\ast$ has $t$ twins, including itself, then the arc $\ell_1'(\tau(w)) = [a, b]$ in $A_1'$ is replaced with $t$ continuous clones $[a+i/k, b+i/k]$, $0 \leq i < t$, where $k$ equals the number of vertices in $G$. Note that we get the same $A$ for any isomorphic copy of $G$ because the construction depends only on the canonical form $G^\ast$ and the number of vertices $k$.

Define a bijection $\ell: V(G) \to A$ by $\ell(v) = [a+i/k, b+i/k]$, where $[a, b] = \ell_1'(\tau(\alpha(v)))$ and $i$ is the number under which the vertex $v$ appears in its twin-class. The enumeration of each twin-class begins with 0 and can be chosen arbitrary; note that while the definition of $\ell$ depends on this choice, the definition of $A$ does not.

Let us check that $\ell$ is an arc representation of $G$.

Make a simple observation about the transformation of $A_1'$ into $A$: cloning of an arc in $A_1'$ consists in shifting it clockwise in a distance strictly less than 1.
If vertices $u$ and $v$ are not adjacent in $G$, then $\ell'_1(\tau(\alpha(u)))$ and $\ell'_1(\tau(\alpha(v)))$ are disjoint and the space between them in the continuous circle is at least 1. Therefore, their clones $\ell(u)$ and $\ell(v)$ remain disjoint. Suppose now that $u$ and $v$ are adjacent (in particular, they can be twins). Let $x \in \ell_1(\tau(\alpha(u))) \cap \ell_1(\tau(\alpha(v)))$. Then the intersection $\ell'_1(\tau(\alpha(u))) \cap \ell'_1(\tau(\alpha(v)))$ contains two consecutive points $2x$ and $2x+1$. It follows that the point $2x+1$ stays in the intersection $\ell(u) \cap \ell(v)$ even if one of the arcs or both are cloned.

Moreover, if $\ell'_1(\tau(\alpha(u))) \not\subset \ell'_1(\tau(\alpha(v)))$, then $\ell(u) \not\subset \ell(v)$. This is true because $\ell'_1(\tau(\alpha(u)))$ contains a point outside $\ell'_1(\tau(\alpha(v)))$ at the distance at least 1 to both extreme points of $\ell'_1(\tau(\alpha(v)))$. Thus, if $\ell_1$ is proper, so is $\ell$.

Finally, we convert $\mathcal{A}$ into a discrete arc model by scaling up the continuous circle of length $2n$ to the circle of length $2kn$. Then every extreme point of an arc, which is a rational of the form $m/k$, becomes an integer. Since the result is the same for any isomorphic copy of $G$, the constructed arc representation is canonical.

Thus, the statement of Theorem 6.1 for co-bipartite TCA graphs is proved. It remains to prove the statement concerning co-bipartite PCA graphs, for which a canonical arc representation should be proper. We follow the same general scheme, provided by Lemma 7.2. Since co-bipartite PCA graphs are co-convex, for them we have canonical labeling in logspace. Now, we only need to construct a proper arc representation in logspace, and this is enough to do for twin-free input graphs.

**Lemma 7.3.** Given a twin-free co-bipartite PCA graph $G$, one can in logspace construct a proper arc representation of $G$.

**Proof.** By Lemma 5.6, the hypergraph $\mathcal{N}[G]$ has a tight arc ordering. We can find it in logspace by running the algorithm of Theorem 3.2 on the tightened hypergraph $(\mathcal{N}[G])^\circ$; see Lemma 2.4. Any tight arc ordering of $\mathcal{N}[G]$ is also a tight arc ordering of $\mathcal{N}(H)$, where $H = \overrightarrow{G}$. Let $V(H) = U \cup W$ be a bipartition of $H$ into two independent sets. Note that the restriction of a tight arc ordering of $\mathcal{N}(H)$ is a tight interval ordering of $\mathcal{N}_U(H)$.

Let us retrace the construction of an arc representation of $G$ described in the proof of Theorem 7.1, where we can now assume that the interval ordering (7.1) is tight. Then the construction produces a tight arc representation $\ell$ of $G$. Indeed, the arc system $\{\ell(u)\}_{u \in U}$ is proper by construction. Furthermore, the arc system $\{\ell(w)\}_{w \in W}$ is tight; this follows by construction from the tightness of $\mathcal{N}_U(H)$. Finally, any two arcs $\ell(u)$ for $u \in U$ and $\ell(w)$ for $w \in W$ are incomparable under inclusion. Indeed, $\ell(w) \not\subset \ell(u)$ because $\ell(w)$ contains 0 while $\ell(u)$ does not. To see that $\ell(u) \not\subset \ell(w)$, notice that, if $N_H(w) = \emptyset$, then $\ell(w)$ contains exactly one extreme point of each $\ell(u)$, and if $N_H(w) \ni u'$, then $\ell(w) \cap \ell(u') = \emptyset$ while $\ell(u')$ contains one extreme point of each $\ell(u)$.

It remains to invoke the logspace algorithm of Lemma 5.16.2 to convert $\ell$ into a proper arc representation of $G$.

The proof of Theorem 6.1 is complete. In conclusion of this section, note that we have already done everything to obtain the characterization of PCA graphs that was announced in Section 5.1.
Proposition 7.4. \( G \) is a PCA graph iff \( N[G] \) is a tight CA hypergraph.

Proof. If \( G \) is PCA, \( N[G] \) is a tight CA hypergraph by Lemma 5.6. Conversely, assume that \( N[G] \) is a tight CA hypergraph. If \( \overline{G} \) is not bipartite, then \( G \) is PCA because all non-co-bipartite TCA graphs are PCA; see Proposition 5.5 or Lemma 5.18. If \( \overline{G} \) is bipartite \( ^4 \) then \( G \) is PCA by the argument in the proof of Lemma 7.3 (it is well enough to show this for a twin-free \( G \)).

8 Star System Problem

We now present our results on the search version of the Star System Problem: Given a hypergraph \( \mathcal{H} \), one has to find a graph \( G \) in a specified class such that \( N[G] = \mathcal{H} \). Referring to this problem, we will use the shorthand SSP.

Theorem 8.1.

1. The SSP for PCA and co-convex graphs is solvable in logarithmic space.
2. If \( G \) is a PCA or co-convex graph, then \( N[G] \cong N[H] \) implies \( G \cong H \).
3. Moreover, if \( G \) is proper interval or non-co-bipartite PCA, then \( N[G] = N[H] \) implies \( G = H \).

Note that, if \( C' \) is a subclass of a class of graphs \( C \), then the logspace solvability of the SSP for \( C \) does still not imply the logspace solvability of the SSP for \( C' \). However, it does if solutions of the problem for \( C \) are unique up to isomorphism and \( C' \) is recognizable in logspace.

Corollary 8.2. The SSP is solvable in logarithmic space for each of the classes of proper interval, PCA, and TCA graphs.

The logspace solvability of the SSP for proper interval graphs follows also from [17]. Our approach is different. We present it in Subsection 8.1 in full detail, as the same idea allows us to treat non-co-bipartite PCA graphs in Subsection 8.2. Note that Part 3 of Theorem 8.1 is a new result even for proper interval graphs.

The SSP can be posed also for a class of hypergraphs \( \mathcal{C} \), where for a given hypergraph \( \mathcal{H} \in \mathcal{C} \) one has to find an (arbitrary) graph \( G \) such that \( N[G] = \mathcal{H} \). If there is a class of graphs \( C \) such that \( G \in C \) iff \( N[G] \in \mathcal{C} \), then the SSP for \( \mathcal{C} \) and the SSP for \( C \) are equivalent.

Corollary 8.3.

1. The SSP for CA hypergraphs is solvable in logarithmic space.

\(^4\)In this case the statement could be alternatively derived from the fact that co-bipartite PCA graphs are exactly complements of bipartite permutation graphs [37, Section 13.4.2] and from the characterization of the latter class in [38, Theorem 1].
2. The solution is unique up to isomorphism. Moreover, the solution is unique if the input is an interval hypergraph.

The proof of Theorem 8.1 takes the rest of this section. We begin with the simplest case of the SSP for proper interval graphs, then consider non-co-bipartite PCA graphs, and conclude with co-convex graphs. Note that the last case includes co-bipartite TCA and PCA graphs.

8.1 Proper interval graphs

We first make a general remark: A graph $G$ is connected iff the hypergraph $\mathcal{N}[G]$ is connected and, hence, there is a one-to-one correspondence between the connected components of $G$ and the connected components of $\mathcal{N}[G]$. It readily follows that, if we are able to solve the SSP on connected input hypergraphs, we can do it as well in general. The same applies to proving the uniqueness of a solution.

Suppose, therefore, that we are given a connected hypergraph $\mathcal{H}$. By Proposition 5.1, the SSP on $\mathcal{H}$ has a solution among proper interval graphs iff $\mathcal{H}$ is interval and, moreover, no solution outside this class of graphs is then possible. Whether or not $\mathcal{H}$ is interval can be checked by the algorithm of Theorem 3.1. Suppose, therefore, that $\mathcal{H} = \mathcal{N}[G]$ for an unknown proper interval graph $G$. Assume first that $\mathcal{H}$ is twin-free, which is true exactly when $G$ is twin-free. By Lemma 5.15.1, $\mathcal{H}$ admits a unique, up to reversal, interval order $<$ on its vertices. This order is computable in logspace by the algorithm of Theorem 3.1. Let $\ast$ denote the corresponding lexicographic order on the hyperedges of $\mathcal{H}$, which is logspace computable from $<$. By Lemma 5.8.1, we have $v < u$ iff $\mathcal{N}[v] < \ast \mathcal{N}[u]$. Thus, the orders $<$ and $\ast$ allow us to establish the $v$-to-$\mathcal{N}[v]$ correspondence, that is, for each hyperedge $H \in \mathcal{H}$, to find a vertex $v$ such that $\mathcal{N}[v] = H$.

Since the $v$-to-$\mathcal{N}[v]$ correspondence is uniquely reconstructible in the class of proper interval graphs and no solution outside this class exists, the solution $G$ is unique.5

Consider now the case that $\mathcal{H}$ has twins. In the quotient-hypergraph $\mathcal{H}'$, the vertices are the twin-classes of $\mathcal{H}$, and a set of twin-classes is a hyperedge in $\mathcal{H}'$ iff the union of these twin-classes is a hyperedge in $\mathcal{H}$. Note that $\mathcal{H}'$ is always twin-free and interval whenever $\mathcal{H}$ is interval. Given a graph $G$, we will denote its quotient graph by $G'$.

If $\mathcal{H}$ is not interval, there is no solution. Otherwise, if $\mathcal{H} = \mathcal{N}[G]$, then $\mathcal{H}' = \mathcal{N}'[G']$. Since $\mathcal{H}'$ is interval and twin-free, $G'$ is unique and constructible in logspace. It remains to note that $G$ is uniquely reconstructible from $G'$, which can very easily be done in logspace.

5The uniqueness results of Harary and McKee 13 and Fomin et al. 17 imply a somewhat weaker fact, namely the uniqueness within the class of proper interval graphs.
8.2 Non-co-bipartite PCA graphs

We first solve the recognition version of the problem. By Proposition 7.4, the equality $H = N[G]$ with a given hypergraph $H$ and an unknown graph $G$ has a solution in the class of PCA graphs iff $H$ is a tight CA hypergraph, which is equivalent to the condition that the tightened hypergraph $H^\equiv$ is CA. Assume that $H = N[G]$ for a PCA graph $G$. Since $H$ is CA in this case, $G$ must be convex-round. By Proposition 5.4, $G$ is bipartite iff $H = N(G)$ is an interval hypergraph. It follows that the SSP on $H$ has a solution in the class of non-co-bipartite PCA graphs iff $H^\equiv$ is CA and $H$ is not interval. Both conditions are verifiable by the logspace algorithms of Section 3. Note also that, if $H$ satisfies the two conditions, then any solution is a non-co-bipartite PCA graph.

Let us turn to the search version. The case of an interval $H$ is just considered above. Suppose, therefore, that $H$ is a tight CA, non-interval hypergraph. Note that $H$ is connected in this case. Suppose also that $\overline{H}$ is not interval and, hence, $H = N[G]$ for some non-co-bipartite PCA graph $G$. Once again in this paper, we treat the case of non-co-bipartite PCA graphs quite similarly to the case of proper interval graphs.

Assume first that $H$ is twin-free. By Lemma 5.8.2, $H$ admits a unique, up to reversal, arc ordering $\prec$ on its vertices. Recall that we write $\prec$ to denote the corresponding circular successor relation. Let $V(H) = \{v_0, v_2, \ldots, v_{n-1}\}$, where the vertices are listed so that $v_i \prec v_{i+1}$ and $v_{n-1} \prec v_0$. This order is computable in logspace by the algorithm of Theorem 3.2. Let $\prec^*$ denote the corresponding lexicographic circular order on the hyperedges of $H$, which is logspace computable from $\prec$. Let $H = \{H_0, H_2, \ldots, H_{n-1}\}$, where $H_i \prec^* H_{i+1}$ and $H_{n-1} \prec^* H_0$. By Lemma 5.8.2, we have $v \prec u$ iff $N[v] \prec^* N[u]$. Like in the case of proper interval graphs, the orders $\prec$ and $\prec^*$ give us the $v$-to-$N[v]$ correspondence, but this time only up to a cyclic shift. More specifically, an assignment $N[v_0] = H_k$ of the neighborhood to one vertex determines the whole assignment to all vertices, namely, $N[v_i] = H_{i+k}$, where the addition is modulo $n$. For each of the $n$ possibilities, we have to check if the assignment really corresponds to some graph, that is, if for all $i$ and $j$ it holds $v_i \in H_{i+k}$ and the conditions $v_i \in H_j$ and $v_j \in H_i$ are true or false simultaneously.

Under the assumptions made about $H$, we will succeed at least once. W.l.o.g. suppose that the assignment $H_i = N[v_i]$ corresponds to some non-co-bipartite PCA graph $G$. Let us show that any other assignment $H_i \mapsto N[v_{i+k}]$ for $1 \leq k < n$ does not correspond to any graph. Let $H_i = [v_{s_i}, v_{t_i}]$, where the arc notation is w.r.t. $\prec$. As usually, denote $N^+[v_i] = [v_i, v_{t_i}]$ and $N^-[v_i] = [v_{s_i}, v_i]$. By Lemma 5.13.1, $v_j \in N^+[v_i]$ iff $v_i \in N^-[v_j]$. This allows us to construct in a consistent way a round orientation of $G$, as defined at the end of Section 5.3. We orient an edge $\{v_i, v_j\}$ as an ordered pair $(v_i, v_j)$ if $v_j \in N^+[v_i]$. The sum of outdegrees over the vertices of $G$ is equal to $\sum_{i=0}^{n-1} |(v_i, v_{t_i})|$, while the sum of indegrees is equal to $\sum_{i=0}^{n-1} |(v_{s_i}, v_i)|$. As a general fact about graph orientations, the two sums must be equal, each being equal to the number of edges in $G$. Now, suppose that $H_i \mapsto N[v_{i+k}]$ is another legitimate correspondence between the hyperedges of $H$ and the vertices of some non-co-bipartite PCA graph. Since $H$ is not an interval hypergraph and, hence, no
graph with closed neighborhood hypergraph \( H \) is proper interval, \( v_{i+k} \) cannot be an extreme point of \( H_i \), that is, \( v_{i+k} \in (v_{s_i}, v_{t_i}) \) for all \( i \). Assume that \( v_k \in (v_0, v_{t_0}) \). It follows by Lemma \( 5.7.2 \) that we also have \( v_{1+k} \in (v_1, v_{t_1}) \). A finite induction gives us that \( v_{i+k} \in (v_i, v_{t_i}) \) for all \( i \). Observe that

\[
\sum_{i=0}^{n-1} |(v_{i+k}, v_{t_i})| < \sum_{i=0}^{n-1} |(v_i, v_{t_i})|,
\]

while

\[
\sum_{i=0}^{n-1} |(v_{s_i}, v_{i+k})| > \sum_{i=0}^{n-1} |(v_{s_i}, v_i)|,
\]

where the inequality actually holds for each pair of summands. This makes a contradiction because the equality of the sums of in- and outdegrees is violated. The analysis of the case when \( v_k \in (v_{s_0}, v_0) \) is symmetric; we only have to run the finite induction in the counter-clockwise direction.

Thus, the \( v \)-to-\( N[v] \) correspondence is uniquely reconstructible in the class of non-co-bipartite PCA graphs. Since no solution exists outside this class, the solution \( G \) is unique.

The case when \( H \) has twins reduces to the twin-free case similarly to proper interval graphs.

### 8.3 Co-convex graphs

We will need the following auxiliary fact.

**Lemma 8.4.** Let \( K \) be a hypergraph with no isolated vertex. Suppose that \( K = N(K) \) for a graph \( K \) and that \( L \) is a connected component of \( K \). Denote \( U = V(L) \). Then either \( U \) is an independent set in the graph \( K \) or it spans a connected component of \( K \). Moreover, if \( U \) is independent, then there is a connected component of \( K \) that is a bipartite graph with \( U \) being one of its vertex classes.

**Proof.** If \( U \) is not independent in \( K \), it contains at least two adjacent vertices \( u_1 \) and \( u_2 \). Let \( K' \) denote the connected component of \( K \) containing \( u_1 \) and \( u_2 \). By connectedness of \( L \), the set \( U \) contains both neighborhoods \( N_K(u_1) \) and \( N_K(u_2) \). We can apply this observation to each edge along any path in \( K' \). It readily follows that \( V(K') \subseteq U \). In fact, \( V(K') = U \) because otherwise \( L \) would be disconnected.

Assume now that \( U \) is independent in \( K \). Since the hypergraph \( K \) has no isolated vertex, the graph \( K \) also has none. Consider a vertex \( u \in U \) and a vertex \( w \) adjacent to \( u \) in \( K \). Let \( L' \) be the connected component of \( K \) containing \( w \). As shown above, the set of vertices \( W = V(L') \) is independent in \( K \). By connectedness of \( L \) and \( L' \), once we have an edge \( uv \) between \( U \) and \( W \), we have \( N_K(w) \subseteq U \) and \( N_K(w) \subseteq W \). Let \( K' \) denote now the connected component of \( K \) containing \( u \) and \( w \). This observation is applicable to each edge along any path in \( K' \). It follows that \( K' \) is bipartite with one vertex class included in \( U \) and the other in \( W \). In fact, the vertex classes of \( K' \) coincide with \( U \) and \( W \) by connectedness of \( L \) and \( L' \). 

\( \blacksquare \)
Given an input hypergraph $\mathcal{H}$, assume that $\mathcal{H} = \mathcal{N}[G]$ for a co-convex graph $G$. To facilitate the exposition, suppose first that the complementary graph $\overline{G}$ is connected, with vertex classes $U$ and $W$. In this case, 

$$\overline{\mathcal{H}} = \overline{\mathcal{N}[G]} = \mathcal{N}(\overline{G}) = N_U(\overline{G}) \cup N_W(\overline{G}),$$

where the vertex-disjoint hypergraphs $U = N_U(\overline{G})$ and $W = N_W(\overline{G})$ are dual (i.e., $U^* \cong W$), both connected, and one of them is interval, say, $U$.

Denote $K = \overline{\mathcal{H}}$ and assume that $K = \mathcal{N}(K)$ for another graph $K$. Lemma 8.4 implies that either both $U$ and $W$ are independent in a connected bipartite $K$ or both span connected subgraphs of $K$, $K_1$ on $U$ and $K_2$ on $W$. However, the latter is impossible. Indeed, since $\mathcal{N}(K_1) = U$ is interval, $K_1$ should be bipartite by Proposition 5.4, contradictory to the connectedness of $U$. Therefore, $K$ is connected bipartite with vertex classes $U$ and $W$. Since $K$ is isomorphic to the incidence graph of the hypergraph $U$ (as well as $W$), it is logspace reconstructible from $K$ up to isomorphism and, in particular, $K \cong \overline{G}$. Thus, the solution to the SSP on $\mathcal{H}$ is unique up to isomorphism.

After this analysis we are able to describe a logspace algorithm solving the SSP for the class of co-convex co-connected graphs. Given a hypergraph $\mathcal{H}$, we first check if $\overline{\mathcal{H}}$ has exactly two connected components, say $U$ and $W$. This can be done by running the Reingold algorithm for the connectivity problem [35] on the intersection graph $I(\overline{\mathcal{H}})$. If this is not the case, there is no solution in the desired class. Otherwise, we construct the incidence graph $H$ of the hypergraph $U$ (or of $W$, which should give the same result up to isomorphism) and take its complement $\overline{H}$. Note that this works well even if $\overline{\mathcal{H}}$ has twins: the twins in $V(U)$ are explicitly present, while the twins in $V(W)$ are represented by multiple hyperedges in $U$.

This is still not a complete solution to the SSP because the closed neighborhood hypergraph $\mathcal{H}' = \mathcal{N}[\overline{\mathcal{H}}]$, though isomorphic to $\mathcal{H}$, may be not equal to $\mathcal{H}$. In this case we have to find an isomorphism $\phi$ from $\mathcal{H}'$ to $\mathcal{H}$ or, the same, from $\overline{\mathcal{H}}$ to $\overline{\mathcal{H}}$. This can be done by the algorithms of Theorem 3.1 and Corollary 3.5 because at least one of the connected components of $\overline{\mathcal{H}} \cong \overline{\mathcal{H}}$ is an interval hypergraph and the other component is isomorphic to the dual of an interval hypergraph. The isomorphic image $G = \phi(\overline{\mathcal{H}})$ is the desired solution to the SSP on $\mathcal{H}$.

If we do not succeed with establishing an isomorphism between $\mathcal{H}'$ and $\mathcal{H}$, this implies that there is no solution in the desired class. Alternatively, we could check from the very beginning whether one of the hypergraphs $U$ and $W$ is interval and $U^* \cong W$.

It remains to consider the general case when $\mathcal{H} = \mathcal{N}[G]$ for a co-convex graph $G$ with not necessary connected complement $\overline{G}$. Note that universal vertices of $G$ are easy to identify in $\mathcal{H}$: those are the vertices contained in every hyperedge of $\mathcal{H}$. We can remove all such vertices from $\mathcal{H}$, solve the SSP for the reduced hypergraph, and then restore a solution for $\mathcal{H}$. The last step can be done in a unique way. We will, therefore, assume that $G$ has no universal vertex or, equivalently, $\overline{\mathcal{H}} = \mathcal{N}(\overline{G})$ has no isolated vertex.

---

6The uniqueness result of Boros et al. [7] implies a somewhat weaker fact, namely the uniqueness up to isomorphism within the class of co-convex graphs.
Given the lack of progress since Hsu’s graphs, how efficiently can the canonical arc representation problem be solved for is solvable in logspace. Co-convex graphs form a proper subclass of co-bipartite CA can be modified to compute canonical arc representations, it also makes sense to consider further subclasses of CA graphs.

In this analysis, we consider the SSP on H is unique up to isomorphism. This analysis suggests the following logspace algorithm solving the SSP for the class of co-convex graphs without universal vertices. Given a hypergraph H, we first check if H has an even number of connected components that can be split into pairs H and W_i so that H_i is an interval hypergraph and W_i ∼= U_i. This step can be done by using the Reingold algorithm and the algorithm of Theorem 3.1. A desired solution exists if and only if this is possible.

Note that some of the hypergraphs W_i can also be interval. Then the set \{U_i\}_{i=1}^{k} can be chosen in essentially different (non-isomorphic) ways; however, all these choices will give isomorphic outcomes (as all choices of \{U_i\}_{i=1}^{k} are equivalent up to isomorphism and taking duals).

Then, for each i, we construct the incidence graph H_i of the hypergraph U_i, form the graph H as the vertex-disjoint union of all H_i, and take its complement \overline{H}.

By the already established uniqueness, the closed neighborhood hypergraph H' = \mathcal{N}(H) is isomorphic to H. We find an isomorphism \phi from H' to H, and the same, from \overline{H'} to \overline{H}. We do it componentwise by running the algorithms of Theorem 3.1 and Corollary 3.5 on the connected components of \overline{H'} and \overline{H}. The isomorphic image G = \phi(H) is a solution as \mathcal{N}(\phi(H)) = \phi(\mathcal{N}(H)) = H.

9 Questions and comments

Canonical arc representations. It remains an intriguing open question to construct canonical arc models for general CA graphs in linear time, NC, or logspace. Given the lack of progress since Hsu’s O(m + n) isomorphism algorithm [19] (which can be modified to compute canonical arc representations), it also makes sense to consider further subclasses of CA graphs.

By Theorem 7.1, the canonical arc representation problem for co-convex graphs is solvable in logspace. Co-convex graphs form a proper subclass of co-bipartite CA graphs. How efficiently can the canonical arc representation problem be solved for
the latter class? The structural properties of co-bipartite CA graphs are surveyed in [31, Section 7].

A graph is Helly CA if it admits a Helly arc representation, where each clique can be assigned a single point on the circle that is contained in the arcs of all vertices in that clique. Joeris et al. show how to construct a Helly arc model of a Helly CA graph in linear time [22], and Curtis gives linear time canonical representation [14]. Can this also be done in NC or logspace?

**Unit arc representations.** Graphs admitting unit arc models, where all arcs have the same length, are called unit CA. They form a proper subclass of PCA graphs; see Fig. 5. Though we can construct canonical proper arc models of unit CA graphs in logarithmic space, they are not necessarily unit. Lin and Szwarcfiter construct unit arc models in linear time [30]. It remains an interesting open question if unit arc representations can be found in NC or logspace.

**Star System Problem.** $C_4$- and $C_3$-free graphs. The algorithm of Fomin et al. [17] for $C_4$-free graphs is implementable in logarithmic space in a straightforward way. The implication (1.1) for two $C_4$-free graphs $G$ and $H$ follows directly from the argument of [17].

The approach of [17] to $C_3$-free graphs is based on the following observation: If $G$ is $C_3$-free and $H = \mathcal{N}[G]$, then for any pair of vertices $u$ and $v$ adjacent in $G$ there are exactly two hyperedges $X$ and $Y$ in $H$ containing both $u$ and $v$. It follows that $\mathcal{N}_G[v] \in \{X, Y\}$ and, moreover, the assumption that $\mathcal{N}_G[v] = X$ forces the conclusion that $\mathcal{N}_G[u] = Y$.

Let us show how to derive from here the logspace solvability of the SSP for $C_3$-free graphs. Since the composition of logspace computable functions is logspace computable, we can split the whole algorithm into a few steps, each doable in logspace. W.l.o.g. assume that an input hypergraph $\mathcal{H}$ is connected. We first construct an auxiliary graph $F$. The vertices of $F$ are all pairs $(v, X)$ such that $v \in X \in \mathcal{H}$. Two vertices $(v, X)$ and $(u, Y)$ are adjacent in $F$ iff $X$ and $Y$ are the only two hyperedges of $\mathcal{H}$ containing both $v$ and $u$.

Fix an arbitrary vertex $v$ of $\mathcal{H}$. For each vertex $(v, X)$ of $F$, we now try to construct a vertex-hyperedge assignment $A_X$ as follows. Assign $X$ to $v$. To each other $u$ we assign an $Y$ such that $(u, Y)$ is reachable from $(v, X)$ in $F$; here we use the Reingold reachability algorithm [35]. For some $u$, the choice of $Y$ may be impossible or ambiguous.

For each successfully constructed assignment $A_X$, we then try to construct a graph $G_X$ by connecting each $u$ with all other vertices in the assigned hyperedge $Y$. For each successfully constructed $G_X$, it remains to check if $\mathcal{N}[G_X] = \mathcal{H}$ and if $G_X$ is $C_3$-free. The description of the algorithm is complete.

Note a useful fact that follows from the above discussion: If a hypergraph $\mathcal{H}$ is connected, then for any hyperedge $X \in \mathcal{H}$ and vertex $v \in X$ there is at most one $C_3$-free graph $G$ such that $\mathcal{H} = \mathcal{N}[G]$ and $X = \mathcal{N}_G[v]$. Thus, the SSP on $\mathcal{H}$ has at most $\min_{X \in \mathcal{H}} |X|$ triangle-free solutions, and all of them can be computed in logspace. It readily follows that the SSP is solvable in logspace for any logspace recognizable class consisting of $C_3$-free graphs. In particular, this applies to the class of bipartite graphs.
Circular-arc graphs. The results of Fomin et al. [17] for \( C_4 \)-free graphs imply, in particular, that the SSP for interval graphs is solvable in logspace. For the whole class of CA graphs this remains an open problem.

Recall that the SSP for co-bipartite graphs is equivalent to the graph isomorphism problem [1], and any two co-bipartite solutions are isomorphic [1,7]. What is the complexity of the SSP for classes between co-bipartite and co-convex graphs, in particular for co-bipartite CA graphs?

Another interesting open class consists of claw-free CA graphs. The class of TCA graphs is properly included here. By [17, Corollary 16], the SSP for all claw-free graphs is NP-hard.

Reconstructibility of \( G \) from \( N[G] \). Call a graph \( G \) (exactly) reconstructible if, for any graph \( H \), the equality \( N[G] = N[H] \) implies that \( G = H \). If only a weaker conclusion \( G \simeq H \) is true, we call such a graph \( G \) reconstructible up to isomorphism. Note that vertex-disjoint unions of reconstructible (up to isomorphism) graphs are reconstructible (up to isomorphism). In this terminology, our Theorem 8.1.2–3 says that proper interval and non-co-bipartite PCA graphs are exactly reconstructible, while TCA and co-convex graphs are reconstructible up to isomorphism. To our best knowledge, the exact reconstructibility was previously known only for complete graphs and the reconstructibility up to isomorphism was known for co-forests (Aigner and Triesch [1]).

Note that \( C_4 \) and \( P_5 \), which are co-bipartite PCA graphs, are reconstructible up to isomorphism but not exactly. Therefore, Theorem 8.1.2 cannot be improved to exact reconstructibility.

Can a reconstructibility result be proved for CA graphs? In the weakest form, such a result would claim that the implication (1.1) holds true whenever both \( G \) and \( H \) are CA; even this seems open.

Finally, it would be interesting to know the complexity of recognition if a given graph is exactly reconstructible. This problem is in coNP. We mention also a related problem. Call a hypergraph \( \mathcal{H} \) uniquely realizable if there is a unique graph \( G \) such that \( \mathcal{H} = N[G] \). By our Corollary 8.3, interval hypergraphs are uniquely realizable. A graph \( G \) is exactly reconstructible iff its hypergraph \( N[G] \) is uniquely realizable. The problem of recognizing uniquely realizable hypergraphs belongs to the complexity class US introduced by Blass and Gurevich [5]. The precise complexity status of this problem is a related open question.

Acknowledgement
We thank Bastian Laubner for useful discussions at the early stage of this work.

References

[1] M. Aigner and E. Triesch. Reconstructing a graph from its neighborhood lists. Combinatorics, Probability & Computing, 2:103–113, 1993.
[2] F. S. Annexstein and R. P. Swaminathan. On testing consecutive-ones property in parallel. *Discrete Applied Mathematics*, 88(1-3):7–28, 1998.

[3] J. Bang-Jensen, J. Huang, and L. Ibarra. Recognizing and representing proper interval graphs in parallel using merging and sorting. *Discrete Applied Mathematics*, 155(4):442–456, 2007.

[4] J. Bang-Jensen, J. Huang, and A. Yeo. Convex-round and concave-round graphs. *SIAM J. Discrete Math.*, 13(2):179–193, 2000.

[5] A. Blass and Y. Gurevich. On the unique satisfiability problem. *Information and Control*, 55(1-3):80–88, 1982.

[6] K. Booth and G. Lueker. Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms. *J. Comput. Syst. Sci.*, 13(3):335–379, 1976.

[7] E. Boros, V. Gurvich, and I. E. Zverovich. Neighborhood hypergraphs of bipartite graphs. *Journal of Graph Theory*, 58(1):69–95, 2008.

[8] L. Chen. Efficient parallel recognition of some circular arc graphs, I. *Algorithmica*, 9(3):217–238, 1993.

[9] L. Chen. Graph isomorphism and identification matrices: Parallel algorithms. *IEEE Trans. Parallel Distrib. Syst.*, 7(3):308–319, 1996.

[10] L. Chen. Efficient parallel recognition of some circular arc graphs, II. *Algorithmica*, 17(3):266–280, 1997.

[11] L. Chen. Graph isomorphism and identification matrices: Sequential algorithms. *J. Comput. Syst. Sci.*, 59(3):450–475, 1999.

[12] L. Chen. A selected tour of the theory of identification matrices. *Theor. Comput. Sci.*, 240(2):299–318, 2000.

[13] L. Chen and Y. Yesha. Parallel recognition of the consecutive ones property with applications. *J. Algorithms*, 12(3):375–392, 1991.

[14] A. R. Curtis. Linear-time Helly circular arc graph isomorphism. Manuscript, available at http://www.cs.uwaterloo.ca/~a2curtis/papers/2008/hca-iso.pdf, March 2008.

[15] X. Deng, P. Hell, and J. Huang. Linear-time representation algorithms for proper circular-arc graphs and proper interval graphs. *SIAM J. Comput.*, 25(2):390–403, 1996.

[16] P. Duchet. Classical perfect graphs. An introduction with emphasis on triangulated and interval graphs. *Perfect graphs, Ann. Discrete Math.*, 21, 67-96 (1984)., 1984.
[17] F. V. Fomin, J. Kratochvíl, D. Lokshtanov, F. Mancini, and J. A. Telle. On the complexity of reconstructing $H$-free graphs from their Star Systems. *Journal of Graph Theory*, 68(2):113–124, 2011.

[18] F. Harary and T. A. McKee. The square of a chordal graph. *Discrete Mathematics*, 128(1-3):165–172, 1994.

[19] W.-L. Hsu. $O(MN)$ algorithms for the recognition and isomorphism problems on circular-arc graphs. *SIAM J. Comput.*, 24(3):411–439, 1995.

[20] W.-L. Hsu. A simple test for the consecutive ones property. *J. Algorithms*, 43(1):1–16, 2002.

[21] W.-L. Hsu and R. M. McConnell. PC trees and circular-ones arrangements. *Theoretical Computer Science*, 296(1):99–116, 3 2003.

[22] B. L. Joeris, M. C. Lin, R. M. McConnell, J. P. Spinrad, and J. L. Szwarcfiter. Linear time recognition of helly circular-arc models and graphs. *Algorithmica*, 59(2):215–239, 2 2011.

[23] H. Kaplan and Y. Nussbaum. Certifying algorithms for recognizing proper circular-arc graphs and unit circular-arc graphs. *Discrete Applied Mathematics*, 157(15):3216–3230, 2009.

[24] H. Kaplan and Y. Nussbaum. A simpler linear-time recognition of circular-arc graphs. *Algorithmica*, 61(3):694–737, 2011.

[25] P. N. Klein. Efficient parallel algorithms for chordal graphs. *SIAM J. Comput.*, 25(4):797–827, 1996.

[26] J. Köbler, S. Kuhnert, B. Laubner, and O. Verbitsky. Interval graphs: Canonical representations in Logspace. *SIAM J. on Computing*, 40(5):1292–1315, 2011.

[27] F. Lalonde. Le probleme d’etoiles pour graphes est np-complet. *Discrete Mathematics*, 33(3):271–280, 1981.

[28] Y. Liang and N. Blum. Circular convex bipartite graphs: Maximum matching and Hamiltonian circuits. *Inf. Process. Lett.*, 56(4):215–219, 1995.

[29] M. C. Lin, F. J. Soulignac, and J. L. Szwarcfiter. A simple linear time algorithm for the isomorphism problem on proper circular-arc graphs. In J. Gudmundsson, editor, *Proceedings of the 11th Scandinavian Workshop on Algorithm Theory*, volume 5124 of *Lecture Notes in Computer Science*, pages 355–366. Springer, 2008.

[30] M. C. Lin and J. L. Szwarcfiter. Unit circular-arc graph representations and feasible circulations. *SIAM Journal on Discrete Mathematics*, 22(1):409–423, 2008.
[31] M. C. Lin and J. L. Szwarcfiter. Characterizations and recognition of circular-arc graphs and subclasses: A survey. *Discrete Mathematics*, 309(18):5618–5635, 2009.

[32] G. Lueker and K. Booth. A linear time algorithm for deciding interval graph isomorphism. *J. ACM*, 26(2):183–195, 1979.

[33] R. M. McConnell. Linear-time recognition of circular-arc graphs. *Algorithmica*, 37(2):93–147, 2003.

[34] J. Reif. Symmetric complementation. *J. ACM*, 31(2):401–421, 1984.

[35] O. Reingold. Undirected connectivity in log-space. *J. ACM*, 55(4), 2008.

[36] F. Roberts. Indifference graphs. Proof Tech. Graph Theory, Proc. 2nd Ann Arbor Graph Theory Conf. 1968, 139-146 (1969), 1969.

[37] J. Spinrad. *Efficient graph representations*. Number 19 in Field Institute Monographs. AMS, 2003.

[38] J. Spinrad, A. Brandstädt, and L. Stewart. Bipartite permutation graphs. *Discrete Appl. Math.*, 18:279–292, 1987.

[39] A. Tucker. Matrix characterizations of circular-arc graphs. *Pac. J. Math.*, 39:535–545, 1971.

[40] Universität Rostock. *Information System on Graph Classes and their Inclusions*. http://wwwteo.informatik.uni-rostock.de/isgci/.