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

Johannes Köbler, Sebastian Kuhnert, and Oleg Verbitsky

Humboldt-Universität zu Berlin, Institut für Informatik, Unter den Linden 6, 10099 Berlin, Germany. {koebler,kuhnert,verbitsk}@informatik.hu-berlin.de

4 March 2016

We present a logspace algorithm that constructs a canonical intersection model for a given proper circular-arc graph, where *canonical* means that isomorphic graphs receive identical models. This implies that the recognition and the isomorphism problems for these graphs are solvable in logspace. For the broader class of concave-round graphs, which still possess (not necessarily proper) circular-arc models, we show that a canonical circular-arc model can also be constructed in logspace. As a building block for these results, we design a logspace algorithm for computing canonical circular-arc models of circular-arc hypergraphs. This class of hypergraphs corresponds to matrices with the *circular ones property*, which play an important role in computational genomics. Our results imply that there is a logspace algorithm that decides whether a given matrix has this property.

Furthermore, we consider the Star System Problem that consists in reconstructing a graph from its closed neighborhood hypergraph. We show that this problem is solvable in logarithmic space for the classes of proper circular-arc, concave-round, and co-convex graphs.

Note that solving a problem in logspace implies that it is solvable by a parallel algorithm of the class AC^1. For the problems under consideration, at most AC^2 algorithms were known earlier.

1 Introduction

With a family of sets $\mathcal{H}$ we associate the *intersection graph* $\mathbb{I}(\mathcal{H})$ on vertex set $\mathcal{H}$ where two sets $A, B \in \mathcal{H}$ are adjacent if and only if they have a nonempty intersection. We call $\mathcal{H}$ an

*A preliminary version of this article appeared in the proceedings of FSTTCS 2012 [23]. The second author was supported by DFG grant KO 1053/7–2. The third author was supported by DFG grant VE 652/1–1 and initiated this work under support by the Alexander von Humboldt Fellowship. He is on leave from the Institute for Applied Problems of Mechanics and Mathematics, Lviv, Ukraine.

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 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 there is a (proper) interval model of G. 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 logspace algorithm that for a given PCA graph computes a canonical representation by a proper arc model, where *canonical* means that isomorphic graphs receive identical models. Note that this algorithm provides a simultaneous solution in logspace of both the recognition and the isomorphism problems for the class of PCA graphs.

In [21], along with Bastian Laubner we gave a logspace solution for the canonical representation problem of proper interval graphs. Though PCA graphs may at first glance appear close relatives of proper interval graphs, the extension of the result of [21] achieved here is far from being straightforward. Combinatorial differences between these two classes of graphs are well known, and they are responsible for the fact that algorithms for PCA graphs often need new ideas and are much more involved than the algorithms for the same problems on proper interval graphs; cf. [8, 10, 11, 15, 20, 27, 28, 30, 35]. One combinatorial difference, very important in our context, lies in the relationship of these graph classes 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 defined similarly if, instead of integer intervals, we consider arcs in a discrete circle. With any graph G, we associate its *closed neighborhood hypergraph* $\mathcal{N}[G] = \{N[v]\}_{v \in V(G)}$ on the vertex set of G, where for each vertex v we have the hyperedge $N[v]$ consisting of v and all the vertices adjacent to v. Roberts [33] discovered that G is a proper interval graph if and only if $\mathcal{N}[G]$ is an interval hypergraph. The circular-arc world is more complex. While $\mathcal{N}[G]$ is a CA hypergraph whenever 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 [3] and *Tucker graphs* by Chen [7]. The latter name is justified by Tucker's result [38] saying that all these graphs are CA (although not necessarily proper CA). Hence, it is natural to consider the problem of constructing arc representations for concave-round graphs. We solve this problem in logspace and also in a canonical way.

Our working tool is a logspace algorithm for computing canonical representations of CA hypergraphs. This algorithm can also be used to test in logspace whether a given Boolean matrix has the *circular ones property*, that is, whether the columns can be permuted so that the 1-entries in each row form a segment up to a cyclic shift. Note that a matrix has this property if and only if it is the incidence matrix of a CA hypergraph. The recognition problem of the circular ones property arises in computational biology, namely in analysis of circular genomes [14, 31].

Our techniques are also applicable to the *Star System Problem* where, for a given hypergraph $\mathcal{H}$, we have to find a graph G such that $\mathcal{H} = \mathcal{N}[G]$, if such a graph exists. In the restriction of the problem to a class of graphs C, we seek for G only in C. We give logspace algorithms solving the Star System Problem for PCA and for 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 [11], by Kaplan and Nussbaum [20], and by Soulignac [36]; and in AC^2 by Chen [8]. 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 AC^1. The isomorphism problem for PCA graphs was solved in linear time by Lin, Soulignac, and Szwarcfiter [27]; their algorithm computes canonical representations. Curtis et al. give a linear time isomorphism test for the larger class of concave-round graphs [10].

Chen [7] showed that the isomorphism problem for concave-round graphs is in AC^2. Circular-arc models for concave-round graphs were known to be constructible also in AC^2 (Chen [6]).

Extending these upper bounds to the class of all CA graphs remains a challenging problem. While this class can be recognized in linear time by McConnell's algorithm [30] (along with constructing an intersection model), no polynomial-time isomorphism test for CA graphs is currently known (see the discussion in [10], where a counterexample to the correctness of Hsu's algorithm [16] is given). This provides further evidence that CA graphs are algorithmically harder than interval graphs. For the latter class we have linear-time algorithms for recognition [4] and canonical representation [29] due to the seminal work by Booth and Lueker; logspace algorithms for these tasks are designed in [21].

The aforementioned circular ones property and the related *consecutive ones property* (requiring that the columns can be permuted so that the 1-entries in each row form a segment) were studied in [4, 17, 18], where linear-time algorithms are given; parallel AC^2 algorithms were suggested in [9, 2].

Star System Problem The decision version of the Star System Problem for general graphs is NP-complete (Lalonde [25]). It stays NP-complete if restricted to non-co-bipartite graphs (Aigner and Triesch [1]) or to H-free graphs for H being a cycle or a path on at least 5 vertices (Fomin et al. [13]). The restriction to co-bipartite graphs has the same complexity as the general graph isomorphism problem [1]. Polynomial-time algorithms are known for H-free graphs for H being a cycle or a path on at most 4 vertices [13] and for bipartite graphs (Boros et al. [5]). An analysis of the algorithms in [13] 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 [22]. Moreover, the problem is solvable in logspace for any logspace-recognizable class of C_4-free graphs, in particular, for chordal, interval, and proper interval graphs; see [22].

2 Basic definitions

We use the standard graph-theoretic terminology as, e.g., in [12]. The vertex set of a graph G is denoted by $V(G)$. The *complement of a graph* G is the graph $\overline{G}$ with $V(\overline{G}) = V(G)$ such that two vertices are adjacent in $\overline{G}$ if and only if they are not adjacent in G. 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 $N[v] = N(v) \cup \{v\}$. We call vertices u and v *twins* if $N[u] = N[v]$ and *fraternal* if $N(u) = N(v)$. A vertex u is *universal* if $N[u] = V(G)$. A set $X \subseteq V(G)$ is *independent* if no two vertices in X are adjacent; X is a *clique* if all vertices in X are pairwise adjacent.

An *isomorphism from a graph G to a graph H* is a bijection $\varphi\colon V(G) \to V(H)$ such that any two vertices u and v are adjacent in G if and only if the vertices $\varphi(u)$ and $\varphi(v)$ are adjacent in H. If such an isomorphism exists, G and H are called *isomorphic*, which is denoted as $G \cong H$. The *canonical labeling problem* for a class of graphs C is, given a graph $G \in \mathsf{C}$ with n vertices, to compute a bijection $\lambda_G\colon V(G) \to \{1, \ldots, n\}$ so that $\lambda_G(G) = \lambda_H(H)$ whenever $G \cong H$, where the graph $\lambda_G(G)$ is the image of G under λ_G on the vertex set $\{1, \ldots, n\}$. We say that λ_G is a *canonical labeling* and that $\lambda_G(G)$ 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 the hypergraph $\mathcal{H}$. An *isomorphism from a hypergraph $\mathcal{H}$ to a hypergraph $\mathcal{K}$* is a bijection $\varphi\colon V(\mathcal{H}) \to V(\mathcal{K})$ with $H \in \mathcal{H}$ exactly when $\varphi(H) \in \mathcal{K}$ for all $H \subseteq V(\mathcal{H})$, where $\varphi(H) = \{\,\varphi(u) : u \in H\}$. We will allow *multi-hyperedges*; in this case an isomorphism has to respect multiplicities.

The *complement of a hypergraph $\mathcal{H}$* is the hypergraph $\overline{\mathcal{H}} = \{\overline{H}\}_{H \in \mathcal{H}}$ on the same vertex set, where $\overline{H} = V(\mathcal{H}) \setminus H$. Each hyperedge $\overline{H}$ of $\overline{\mathcal{H}}$ inherits the multiplicity of H in $\mathcal{H}$. With a graph G we associate two hypergraphs defined on the vertex set $V(G)$. The *closed* (resp. *open*) *neighborhood hypergraph* of G is defined by $\mathcal{N}[G] = \{N[v]\}_{v \in V(G)}$ (resp. by $\mathcal{N}(G) = \{N(v)\}_{v \in V(G)}$). *Twins in a hypergraph* are two vertices such that every hyperedge contains either both or none of them. Note that two vertices are twins in $\mathcal{N}[G]$ if and only if they are twins in G.

Let $X = \{x_1, \ldots, x_n\}$. Saying that the sequence $x_1, \ldots, x_n$ is *circularly ordered*, we mean that X is endowed with the (circular successor) relation $\prec$ under which $x_i \prec x_{i+1}$ for $i < n$ and $x_n \prec x_1$. Such a relation $\prec$ will be referred to as a *circular order* on X. In particular, we will use $\mathbb{C}_n$ to denote the initial segment of n positive integers with the circular order $1 \prec 2 \prec \ldots \prec n \prec 1$. Note that a circularly ordered set $(X, \prec)$ can be viewed as a directed cycle. The vertices of this cycle will sometimes be referred to as *points*. An ordered pair of elements $a^-, a^+ \in X$ determines an *arc* $A = [a^-, a^+]$ that consists of the points appearing in the directed path from a^- to a^+. In addition, the set $A = \emptyset$ will be called the *empty arc*. If $A = [a^-, a^+]$ is not the *complete arc* $A = X$, the elements a^- and a^+ will be referred to, respectively, as the *left* and *right extreme points* of A. We stress that in the rest of the paper we will use this terminology and the notation $A = [a^-, a^+]$ only under the assumption that the extreme points of the arc A are uniquely determined by the set A (i.e., when $A \neq \emptyset$ and $A \neq X$). A hypergraph $\mathcal{H}$ with $V(\mathcal{H}) = X$ is called an *arc system* on $(X, \prec)$ if all of its hyperedges form arcs. In this case, the circular order $\prec$ on X will be called a *CA order* of $\mathcal{H}$.

A hypergraph $\mathcal{H}$ is called a *circular-arc (CA) hypergraph* if there is a circular order $\prec$ on $X = V(\mathcal{H})$ such that $\mathcal{H}$ is an arc system on $(X, \prec)$ (or, in other words, if $\mathcal{H}$ admits a CA order $\prec$). Note that a hypergraph $\mathcal{H}$ on n vertices is circular-arc if and only if there is a hypergraph isomorphism ρ from $\mathcal{H}$ to an arc system $\mathcal{A}$ on $\mathbb{C}_n$. Any such isomorphism ρ will be called an *arc representation* of $\mathcal{H}$, and the corresponding arc system $\mathcal{A}$ will be called an *arc model* of $\mathcal{H}$.

An arc system $\mathcal{A}$ is *tight* if any two arcs $A = [a^-, a^+]$ and $B = [b^-, b^+]$ in $\mathcal{A}$ have the following property: if $A \subseteq B$, then $a^- = b^-$ or $a^+ = b^+$ (note that this condition does not apply to the empty or to the complete arc, which can be in $\mathcal{A}$). A CA order of $\mathcal{H}$ is *tight* if it makes $\mathcal{H}$ a tight arc system. Furthermore, we call a CA hypergraph *tight* if it admits a tight CA order or, equivalently, a tight arc model. Recognition of tight CA hypergraphs reduces in logspace to recognition of CA hypergraphs. To see this, given a hypergraph $\mathcal{H}$, define its *tightened hypergraph* $\mathcal{H}^{\Subset}$ by $\mathcal{H}^{\Subset} = \mathcal{H} \cup \{\, A \setminus B : A, B \in \mathcal{H}\}$. Then $\mathcal{H}$ is a tight CA hypergraph if and only if $\mathcal{H}^{\Subset}$ is a CA hypergraph. This equivalence is obvious in the forward direction. For the backward direction, note that any CA order $\prec$ of $\mathcal{H}^{\Subset}$ must be a tight CA order of $\mathcal{H}$ for if $A \subseteq B \setminus \{b^-, b^+\}$ for some arcs $A = [a^-, a^+]$ and $B = [b^-, b^+]$ in $(V(\mathcal{H}), \prec)$, then $B \setminus A = [b^-, a^-) \cup (a^+, b^+]$ could not be an arc.

The notions of an *interval representation*, an *interval model*, and an *interval order* of a hypergraph are introduced similarly to the above, where *interval* means an interval of consecutive integers within $\{1, \ldots, n\}$. Hypergraphs having interval representations are called *interval hypergraphs*. Since any interval representation is an arc representation, they form a subclass of CA hypergraphs.

Given a circular order $\prec$ of a set X, consider the set of all arcs $A \subset X$ in $(X, \prec)$ excepting the empty arc $\emptyset$ and the complete arc X. The relation $\prec$ induces a (lexicographic) circular order $\prec^*$ on this set, where $A \prec^* B$ if $(a^- = b^- \wedge a^+ \prec b^+)$ or $(a^- \prec b^- \wedge |A| = n-1 \wedge |B| = 1)$. The latter condition applies if A is the longest among all arcs with left extreme point a^- and B is the shortest among all arcs with left extreme point b^-. Let $\mathcal{H}$ be an arc system on $(X, \prec)$ containing neither $\emptyset$ nor $V(\mathcal{H}) = X$. By "restricting" $\prec^*$ to the hyperedge set $\mathcal{H}$ we obtain a circular order $\prec_{\mathcal{H}}$ on $\mathcal{H}$: For $A, B \in \mathcal{H}$ we define $A \prec_{\mathcal{H}} B$ if either $A \prec^* B$ or there exists a nonempty sequence of arcs $X_1, \ldots, X_k \notin \mathcal{H}$ such that $A \prec^* X_1 \prec^* \ldots \prec^* X_k \prec^* B$. We say that the circular order $\prec_{\mathcal{H}}$ on $\mathcal{H}$ is *lifted from* the circular order $\prec$ on $V(\mathcal{H})$.

An *arc representation of a graph* G is an isomorphism $\alpha\colon V(G) \to \mathcal{A}$ from G to the intersection graph $\mathbb{I}(\mathcal{A})$ of an arc system $\mathcal{A}$ on $\mathbb{C}_k$. Here we suppose that $\mathcal{A}$ does not contain the empty arc. If $\mathcal{A}$ also does not contain the complete arc $\mathbb{C}_k$, we use the lifted circular order $\prec_{\mathcal{A}}$ on $\mathcal{A}$ to define a circular order $\prec_\alpha$ on $V(G)$, where $u \prec_\alpha v$ if and only if $\alpha(v) \prec_{\mathcal{A}} \alpha(u)$. We call $\prec_\alpha$ the *geometric order* on $V(G)$ associated with α. An arc representation α (and the corresponding arc model) is *proper* if $\alpha(u) \not\subseteq \alpha(v)$ for all vertices $u \neq v$. If α is a proper arc representation of a graph G with more than one vertex, then no arc $\alpha(v)$ is complete, and the geometric order $\prec_\alpha$ is well defined. Graphs having proper arc representations are called *proper circular-arc (PCA)*. An example of a PCA graph along with its proper arc representation can be seen in Fig. 1 below.

Roadmap In Section 3 we show how to compute canonical arc representations for CA hypergraphs in logspace. This procedure will serve as a building block for our algorithms on PCA and concave-round graphs. The connections of these classes of graphs to CA hypergraphs are outlined in Section 4. In particular, we make use of the fact that the neighborhood hypergraph $\mathcal{N}[G]$ of a non-co-bipartite PCA graph G admits a unique CA order, which coincides with the geometric order $\prec_\alpha$ for any proper arc representation α of G. Based on this, in Section 5 we compute canonical representations of non-co-bipartite PCA graphs in logspace. To achieve the same for co-bipartite PCA graphs G (and all concave-round graphs), we use the fact that $\mathcal{N}(\overline{G})$ is in this case an interval hypergraph and show how to convert an interval representation of $\mathcal{N}(\overline{G})$ into an arc representation of G. Finally, in Section 6 we apply the techniques of Section 3 and 4 to the Star System Problem.

3 Canonical arc representations of hypergraphs

In order to solve the *canonical representation problem* for CA hypergraphs we have to compute for a given hypergraph some arc representation (if it exists) such that isomorphic CA hypergraphs obtain identical arc models.

Theorem 3.1. *The canonical representation problem for CA hypergraphs is solvable in logspace.*

Proof. We solve the problem by reducing it to the canonical representation problem for interval hypergraphs, which is already known to be in logspace [21].

Let $\mathcal{H}$ be a given hypergraph with n vertices. For simplicity of exposition we first assume that $\mathcal{H}$ has no multi-hyperedges. For each vertex $x \in V(\mathcal{H})$ we construct the hypergraph $\mathcal{H}_x = \{H_x\}_{H \in \mathcal{H}}$ on the same vertex set, where $H_x = H$ if $x \notin H$ and $H_x = \overline{H} = V(\mathcal{H}) \setminus H$ otherwise. Observe that every $\mathcal{H}_x$ is an interval hypergraph provided that $\mathcal{H}$ is a CA hypergraph; cf. [38, Theorem 1]. Canonizing each $\mathcal{H}_x$ with the algorithm from [21], we obtain n interval representations $\rho_x \colon V(\mathcal{H}) \to \{1, \ldots, n\}$; recall that $V(\mathcal{H}_x) = V(\mathcal{H})$. Whereas each ρ_x is an interval representation of $\mathcal{H}_x$, note that it is also an arc representation of $\mathcal{H}$. Indeed, $\rho_x(\mathcal{H})$ is an arc system because it can be obtained from the canonical interval model $\rho_x(\mathcal{H}_x)$ of $\mathcal{H}_x$ by backward complementing those intervals in $\rho_x(\mathcal{H}_x)$ that correspond to the hyperedges $H_x \in \mathcal{H}_x$ such that $H_x = \overline{H}$. We regard the arc representations ρ_x of $\mathcal{H}$ for all $x \in V(\mathcal{H})$ as n candidates for the canonical arc representation of $\mathcal{H}$. In order to choose one of them, we compare the corresponding arc models $\rho_x(\mathcal{H})$ lexicographically (assuming a natural encoding of arc systems over a finite vocabulary) and output the representation with the lexicographically least arc model.

There is a subtle point in this procedure: We need to distinguish between complemented and non-complemented hyperedges when canonizing $\mathcal{H}_x$; otherwise reverting the complementation could lead to non-equal models for isomorphic CA hypergraphs. For this reason we modify each interval hypergraph $\mathcal{H}_x$ by changing the multiplicities of hyperedges. Specifically, the hyperedge H_x is assigned the multiplicity $c_x(H_x) = 1$ if $x \notin H$ and $c_x(H_x) = 2$ if $x \in H$. Thus, the multiplicity 2 means that the hyperedge is complemented and the multiplicity 1 means that it stays the same in $\mathcal{H}_x$ as in $\mathcal{H}$. There is yet another, special case that both H and $\overline{H}$ are present in $\mathcal{H}$; then H_x occurs in $\mathcal{H}_x$ with multiplicity 2 from the very beginning. To distinguish this case, we assign $c_x(H_x) = 3$ for all such H.

In the general case, an input hypergraph $\mathcal{H}$ contains each hyperedge H with multiplicity $c(H)$. We set $c(H) = 0$ for any set $H \notin \mathcal{H}$. Fix a logspace computable bijection $p \colon \mathbb{Z}_+ \times \mathbb{Z}_+ \to \mathbb{Z}_+$, where $\mathbb{Z}_+$ denotes the set of non-negative integers. For example, p can be chosen to be the Cantor pairing function $p(i, j) = (i + j)(i + j + 1)/2 + j$. Our algorithm constructs the hypergraphs $\mathcal{H}_x$ along with their interval representations ρ_x in the same way as described above, but now each hyperedge $K \in \mathcal{H}_x$ has multiplicity $c_x(K) = p\big(c(K), c(\overline{K})\big)$. Since $(\mathcal{H}_x, c_x)$ retains the isomorphism type of $\mathcal{H}$, the arc representation ρ_x yielding the lexicographically least arc model of $\mathcal{H}$ is canonical. $\square$

Translated into the language of matrices, Theorem 3.1 has algorithmic consequences for testing the circular ones property that was defined in the introduction.

Corollary 3.2. *There is a logspace algorithm that decides whether a given Boolean matrix has the circular ones property and, if affirmative, the algorithm also computes an appropriate permutation of the columns.*

The *canonical labeling problem* for a class of hypergraphs C is defined in the same way as for graphs. Notice a similarity between the pairs of notions *canonical labeling/canonical form* and *canonical representation/canonical model* for CA hypergraphs. The canonical representation algorithm given by Theorem 3.1 also solves the canonical labeling problem for CA hypergraphs in logarithmic space. We conclude this section by noting that it can also be used to compute a canonical labeling for the duals of CA hypergraphs; this will be needed in Section 6.

Given a hypergraph $\mathcal{H}$ and a vertex $v \in V(\mathcal{H})$, let $v^* = \{ H \in \mathcal{H} : v \in H \}$. The hypergraph $\mathcal{H}^* = \{ v^* : v \in V(\mathcal{H}) \}$ on the vertex set $V(\mathcal{H}^*) = \mathcal{H}$ is called the *dual hypergraph* of $\mathcal{H}$ (multi-hyperedges in $\mathcal{H}$ become twin vertices in $\mathcal{H}^*$). The map $\varphi\colon v \mapsto v^*$ is an isomorphism from $\mathcal{H}$ to $(\mathcal{H}^*)^*$. If $\mathcal{H}^*$ is a CA hypergraph, this map can be combined with a canonical labeling λ of $\mathcal{H}^*$ in order to obtain a canonical labeling $\hat{\lambda}$ of $\mathcal{H}$. More precisely, $\hat{\lambda}$ is obtained from the map $\lambda'(v) = \{ \lambda(H) : v \in H \}$ by sorting and renaming the values of λ'.

Corollary 3.3. *The canonical labeling problem for hypergraphs whose duals are CA can be solved in logspace.*

4 Linking PCA graphs and tight CA hypergraphs

Bang-Jensen et al. [3] call a graph G *concave-round* (resp. *convex-round*) if $\mathcal{N}[G]$ (resp. $\mathcal{N}(G)$) is a CA hypergraph. Since $\overline{\mathcal{N}[G]} = \mathcal{N}(\overline{G})$, concave-round and convex-round graphs are co-classes. Using this terminology, a result of Tucker [38] says that PCA graphs are concave-round, and concave-round graphs are CA.

To connect the canonical representation problem for PCA and concave-round graphs to that of CA hypergraphs, we use the fact that the graph classes under consideration can be characterized in terms of neighborhood hypergraphs. For concave-round graphs, this directly follows from their definition, and we can find accompanying hypergraphs also for PCA graphs. The following fact is illustrated by an example in Fig. 1.

Theorem 4.1. *A graph G is PCA if and only if $\mathcal{N}[G]$ is a tight CA hypergraph.*

The forward direction of Theorem 4.1 follows from Lemma 4.2 below. To prove the other direction, we distinguish two cases. If $\overline{G}$ is not bipartite, then a result of Tucker [38] says that G is a PCA graph whenever $\mathcal{N}[G]$ is a CA hypergraph. For the remaining case we show in Section 5 that if $\overline{G}$ is bipartite, then any tight arc model for $\mathcal{N}[G]$ can be transformed into a proper arc model for G. Thus, the proof of Theorem 4.1 will be completed in Section 5; note that we will use this result only later in Section 6.

The following lemma shows that every proper arc representation of a graph G induces a structure on the vertex set of G that agrees very well with the hypergraph $\mathcal{N}[G]$ in the sense that it provides a tight CA order for it. It is known that every PCA graph admits a proper arc representation α having the additional property that no two arcs in the corresponding model cover the whole circle [39] (see also [20] for a very short and useful argument). For proper arc representations with this additional property the conclusion of the lemma can be deduced from

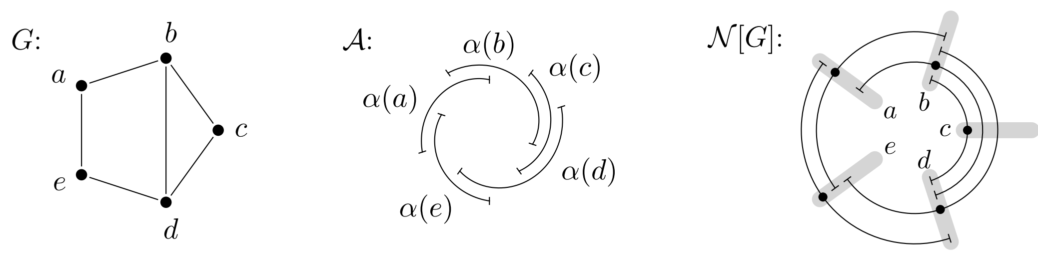

Figure 1: A graph G and a proper arc representation $\alpha\colon V(G) \to \mathcal{A}$ of G, where $\mathcal{A}$ is an arc system on $\mathbb{C}_{10}$. The geometric order $\prec_\alpha$ is a tight arc order of $\mathcal{N}[G]$. In the picture of $\mathcal{N}[G]$, a hyperedge $N[u]$ is marked with a dot at the point u.

the fact that orienting each edge $\{u, v\}$ of G as (u, v) if $\alpha(u)$ contains the right extreme point of $\alpha(v)$ results in a *transitive local tournament* [15, 34]. For the reader's convenience, we give a short proof that works for arbitrary proper arc representations and which is adapted to our framework. A similar argument will be used once again in the proof of Lemma 6.2.

Lemma 4.2. *The geometric order $\prec_\alpha$ on $V(G)$ associated with a proper arc representation α of a graph G is a tight CA order for the hypergraph $\mathcal{N}[G]$.*

Proof. We first show that the neighborhood $N[u]$ of any vertex $u \in V(G)$ is an arc with respect to the order $\prec_\alpha$. Suppose that u is not universal, otherwise the claim is trivial. We will denote the left extreme point of the arc $\alpha(u)$ by $\alpha^-(u)$ and its right extreme point by $\alpha^+(u)$, that is, $\alpha(u) = \big[\alpha^-(u), \alpha^+(u)\big]$. We split $N(u)$ in two parts, namely $N^-(u) = \big\{ v \in N(u) : \alpha^-(u) \in \alpha(v)\big\}$ and $N^+(u) = \big\{ v \in N(u) : \alpha^+(u) \in \alpha(v)\big\}$. Indeed, no vertex v is contained in both $N^-(u)$ and $N^+(u)$. Otherwise, since $\mathcal{A}$ is proper, the arcs $\alpha(v)$ and $\alpha(u)$ would cover the whole circle, both intersecting any other arc $\alpha(w)$, contradicting the assumption that u is non-universal.

Now let $v \in N^+(u)$ and assume that $u \prec_\alpha v_1 \prec_\alpha \cdots \prec_\alpha v_k \prec_\alpha v$. We claim that every vertex v_i is in $N^+(u)$. Indeed, by the definition of $\prec_\alpha$, we have $\alpha(u) \prec_{\mathcal{A}} \alpha(v_1) \prec_{\mathcal{A}} \cdots \prec_{\mathcal{A}} \alpha(v_k) \prec_{\mathcal{A}} \alpha(v)$. Therefore, $\alpha^-(v_i) \in \big(\alpha^-(u), \alpha^-(v)\big)$ and $\alpha^+(v_i) \in \big(\alpha^+(u), \alpha^+(v)\big)$, which implies $\alpha^+(u) \in \big[\alpha^-(v_i), \alpha^+(v_i)\big]$. It follows that $N^+(u) \cup \{u\}$ is an arc starting at u. By a symmetric argument, $N^-(u) \cup \{u\}$ is an arc ending at u. Hence, also $N[u]$ is an arc, implying that $\prec_\alpha$ is a CA order for $\mathcal{N}[G]$.

It remains to show that the CA order $\prec_\alpha$ is tight. Let x be a non-universal vertex of G. Then its neighborhood $N[x]$ forms an arc $N[x] = [x^-, x^+]$ with respect to $\prec_\alpha$ having extreme points $x^-, x^+ \in V(G)$. If $y \in (x, x^+)$, then $\alpha(x)$, $\alpha(y)$, and $\alpha(x^+)$ appear in this sequence with respect to $\prec_{\mathcal{A}}$, implying that $x^+ \in (y, y^+]$. In other words, if x and y are adjacent and $x^+ \neq y^+$, then the vertices x, y, y^+, x^+ never appear in this sequence with respect to $\prec_\alpha$ (ignoring other vertices inbetween). Similarly, the sequence x^-, y^-, y, x is impossible for adjacent x and y unless $x^- = y^-$. It now readily follows that the inclusion $N[u] \subset N[v]$ implies that either $u^- = v^-$ or $u^+ = v^+$. $\square$

Theorem 4.1 suggests that knowing a tight CA order of $\mathcal{N}[G]$ might be useful for constructing a proper arc representation for a PCA graph G. In the case that $\overline{G}$ is not bipartite, the geometric order of such a representation can be obtained by computing an arbitrary CA order of $\mathcal{N}[G]$, which will be necessarily tight. This follows from Lemma 4.2 and the following proposition.

Proposition 4.3. *If G is a connected twin-free PCA graph and $\overline{G}$ is not bipartite, then $\mathcal{N}[G]$ has a unique CA order up to reversing.*

Proposition 4.3 can be derived from a result of Deng, Hell, and Huang [11, Corollary 2.9], which is based on [19, Theorem 4.9] (see also a proof of this fact in [24, Theorem 3.7.1]).

We close this section by giving a characterization of concave-round graphs G with bipartite complement in terms of $\mathcal{N}(\overline{G})$. Given a bipartite graph H and a bipartition $V(H) = U \cup W$ of its vertices into two independent sets, by $\mathcal{N}_U(H)$ we denote the hypergraph $\{N(w)\}_{w \in W}$ on the vertex set U. Note that $\mathcal{N}_U(H)$ and $\mathcal{N}_W(H)$ are dual hypergraphs, i.e., $(\mathcal{N}_U(H))^* \cong \mathcal{N}_W(H)$. A bipartite graph H is called *convex* if its vertex set admits splitting into two independent sets U and W, such that $\mathcal{N}_U(H)$ is an interval hypergraph. If both $\mathcal{N}_U(H)$ and $\mathcal{N}_W(H)$ are interval hypergraphs, H is called *biconvex* [37]. As G is co-bipartite concave-round if and only if its complement $H = \overline{G}$ is bipartite convex-round, the following fact gives the desired characterization.

Proposition 4.4 (Theorem 2.2 in [39])**.** *A graph H is bipartite convex-round if and only if it is biconvex, which in turn is equivalent to $\mathcal{N}(H)$ being an interval hypergraph.*

5 Canonical arc representations of concave-round and PCA graphs

We are now ready to present our canonical representation algorithm for concave-round and PCA graphs. For a given concave-round graph, we have to compute an arc representation such that isomorphic concave-round graphs obtain identical arc models and, moreover, all PCA graphs obtain proper arc models.

Theorem 5.1. *There is a logspace algorithm that solves the canonical arc representation problem for the class of concave-round graphs. Moreover, this algorithm outputs a proper arc representation whenever the input graph is PCA.*

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* a representation algorithm (not necessarily a canonical one). Proving Theorem 5.1 according to this scheme, we split our task into two parts: We first compute a canonical labeling λ of the input graph G and then we compute an arc representation α of the canonical form $\lambda(G)$. Then the composition $\alpha \circ \lambda$ is a canonical arc representation of G. As twins can be easily re-inserted in a (proper) arc representation, it suffices to compute α for the twin-free version of $\lambda(G)$, where in each twin-class we only keep one vertex.

We distinguish two cases depending on whether $\overline{G}$ is bipartite; see Fig. 2 for an overview of the involved graph classes.

5.1 Non-co-bipartite concave-round graphs

As mentioned before, any concave-round graph G whose complement is not bipartite is actually a PCA graph [38]. Hence, we have to compute a proper arc representation in this case.

Canonical labeling We first transform G into its twin-free version G', where we only keep one vertex in each twin-class. Let n be the number of vertices in G'. We use the algorithm given by Theorem 3.1 to compute an arc representation ρ' of $\mathcal{N}[G']$. By Proposition 4.3, $\mathcal{N}[G']$ has a

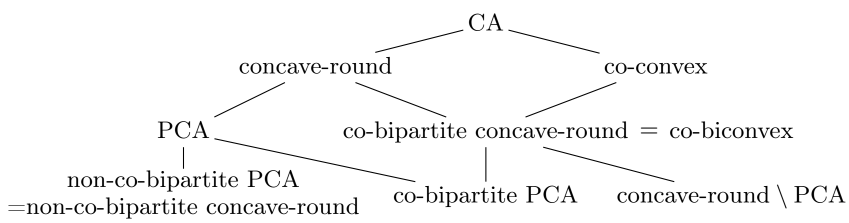

Figure 2: Inclusion structure of the classes of graphs under consideration.

CA order which is unique up to reversing. Hence, in order to determine a canonical labeling of G, it suffices to consider the $2n$ arc representations $\rho_1, \ldots, \rho_{2n}$ of $\mathcal{N}[G]$ that can be obtained from ρ' by cyclic shifting and reversing the points of $\mathbb{C}_n$ and by re-inserting all the removed twins. As a canonical labeling ρ_i of G, we appoint one of these $2n$ variants that gives the canonical form $\rho_i(G)$ of G with the lexicographically least adjacency matrix.

Proper arc representation As mentioned above, it suffices to find such a representation for the twin-free graph G'. The arc representation ρ' of $\mathcal{N}[G']$ that we already have computed provides us with a CA order $\prec$ for $\mathcal{N}[G']$. By Lemma 4.2 and Proposition 4.3, $\prec$ coincides with the geometric order $\prec_\alpha$ for any proper arc representation α of G'. Every PCA graph G' admits a proper arc representation $\alpha\colon V(G') \to \mathcal{A}$ such that no two arcs $\alpha(v) = \big[\alpha^-(v), \alpha^+(v)\big]$ and $\alpha(u) = \big[\alpha^-(u), \alpha^+(u)\big]$ in $\mathcal{A}$ share an extreme point and that $V(\mathcal{A})$ consists of exactly $2n$ points. Such an α is reconstructible in logspace from $\prec$ (uniquely up to the circle rotation) as the left extreme points $\alpha^-(v)$ appear in the circle according to $\prec$, the same holds true for the right extreme points $\alpha^+(v)$, and each right extreme point $\alpha^+(v)$ lies between the left extreme point $\alpha^-(v^+)$ and the following left extreme point, where v^+ is the right extreme point of the arc $N[v]$ with respect to $\prec$. Note that the extreme points of $N[v] = [v^-, v^+]$ are well defined because no vertex v can be universal; otherwise the arcs containing the extreme points of $\alpha(v)$ would correspond to two cliques covering the whole vertex set $V(G')$.

5.2 Co-bipartite concave-round graphs

By Proposition 4.4, co-bipartite concave-round graphs are precisely the co-biconvex graphs. In fact, even all co-convex graphs are circular-arc (this is implicit in [38]) and we can compute a canonical arc representation actually for this larger class of graphs.

Canonical labeling A logspace algorithm for canonical labeling of convex graphs, and hence also co-convex graphs, is designed in [21].

(Proper) arc representation We first recall Tucker's argument [38] showing that, if the complement of G is a convex graph, then G is CA. We can assume that $\overline{G}$ has no fraternal vertices as those would correspond to twins in G.

Let $V(G) = U \cup W$ be a partition of $\overline{G}$ into independent sets such that $\mathcal{N}_U(\overline{G})$ is an interval hypergraph. Let $u_1, \ldots, u_k$ be an interval order on U for $\mathcal{N}_U(\overline{G})$. We construct an arc representation α for G on the cycle $\mathbb{C}_{2k+2}$ (see Fig. 3 for an example) by setting $\alpha(u_i) = [i+1, i+k+1]$ for each $u_i \in U$ and $\alpha(w) = [j+k+2, i]$ for each $w \in W$, where $N_{\overline{G}}(w) = [u_i, u_j]$ and the subscript $\overline{G}$ means that the vertex neighborhood is considered in the complement of G. Note that $\alpha(w) = \mathbb{C}_{2k+2} \setminus \bigcup_{u \in N_{\overline{G}}(w)} \alpha(u)$. In the case that $N_{\overline{G}}(w) = \emptyset$, we set $\alpha(w) = [1, k+1]$. By construction, all arcs $\alpha(u)$ for $u \in U$ share a point (even two, $k+1$ and $k+2$), the same holds true for all $\alpha(w)$ for $w \in W$ (they share the point 1), and any pair $\alpha(u)$ and $\alpha(w)$ is intersecting if and only if u and w are adjacent in G. Thus, α is indeed an arc representation for G.

In order to compute α in logspace, it suffices to compute a suitable bipartition $\{U, W\}$ of $\overline{G}$ and an interval order of the hypergraph $\mathcal{N}_U(\overline{G})$ in logspace. Finding a bipartition $\{U, W\}$ such that $\mathcal{N}_U(\overline{G})$ is an interval hypergraph can be done by splitting $\overline{G}$ into connected components $H_1, \ldots, H_k$ (using Reingold's algorithm [32]) and finding such a bipartition $\{U_i, W_i\}$ for each

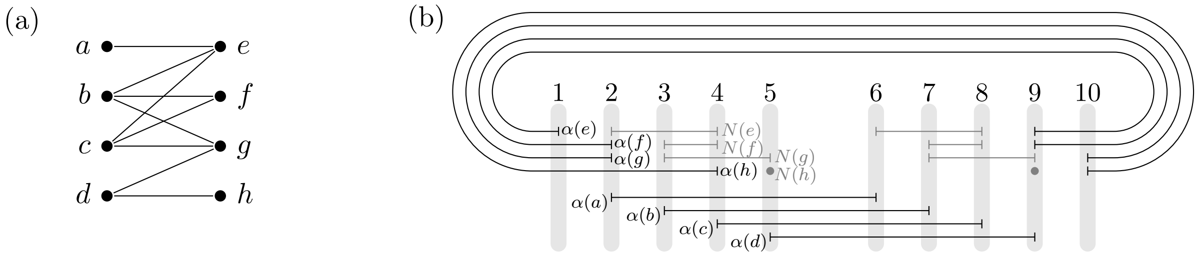

Figure 3: (a) The complement $\overline{G}$ of a co-bipartite concave-round graph G with the bipartition $U = \{a, b, c, d\}$ and $W = \{e, f, g, h\}$. (b) A tight interval order of $\mathcal{N}_U(\overline{G})$ (two shifted copies of which are depicted in gray) is used to construct a tight arc representation α for G on the circle $\mathbb{C}_{10}$ (depicted in black); see the text for details.

component H_i. By using the logspace algorithm of [21] we can actually compute interval orders of the hypergraphs $\mathcal{N}_{U_i}(H_i)$ which can be easily pasted together to give an interval order of $\mathcal{N}_U(\overline{G})$. Together with the canonical labeling algorithm this implies that the canonical arc representation problem for co-convex graphs and, in particular, for co-bipartite concave-round graphs is solvable in logspace.

It remains to show that for co-bipartite PCA graphs we can actually compute a proper arc representation in logspace. As above, we assume that G is twin-free. By Lemma 4.2, the hypergraph $\mathcal{N}[G]$ has a tight CA order $\prec$. We can compute $\prec$ in logspace by running the algorithm given by Theorem 3.1 on the tightened hypergraph $(\mathcal{N}[G])^{\Subset}$. Now, our goal is to convert the tight CA order of $\mathcal{N}[G]$ into a proper arc representation of G. This will also complete the proof of Theorem 4.1 stated in Section 4.

Like any tight CA order of $\mathcal{N}[G]$, $\prec$ is also a tight CA order of $\mathcal{N}(\overline{G})$. Let $V(G) = U \cup W$ be a bipartition of $\overline{G}$ into two independent sets. Note that the restriction of $\prec$ to $\mathcal{N}_U(\overline{G})$ is a tight interval order of the interval hypergraph $\mathcal{N}_U(\overline{G})$. Retracing Tucker's construction of an arc representation α for a co-convex graph G (which is outlined above) in the case that the interval order of $\mathcal{N}_U(\overline{G})$ is tight, we see that α now gives us a tight arc model for G. Note that, by construction, this model contains no complete arc.

It remains to note that any tight α with this property can be converted into a proper arc representation β. Tucker [38] described such a transformation, and Chen [8] observed that it can be implemented in AC^1. This transformation slightly extends contained arcs on the side where their extreme point coincides with an extreme point of the containing arc. The resulting proper arc representation β has the property that the associated geometric orders of α and β on $V(G)$ coincide, that is, $u \prec_\alpha v$ if and only if $u \prec_\beta v$. In order to implement this idea in logspace, we replace each arc $\alpha(v) = \big[\alpha^-(v), \alpha^+(v)\big]$ by an arc $\beta(v) = \big[\beta^-(v), \beta^+(v)\big]$ as described below. First of all, $\beta(v)$ is an arc on the circle $\mathbb{C}_{2n}$, where n is the number of vertices in G. Note that every point of $\mathbb{C}_{2n}$ must be an extreme point of exactly one arc. Without loss of generality, we assume that $\alpha^-(v) = 1$ for some vertex v. Then the left extreme point of $\beta(v)$ is determined by

$$\begin{aligned}\beta^-(v) = &\Big|\Big\{u \in V(G) : \alpha^+(u) \in \big[1, \alpha^-(v)\big)\Big\}\Big| \\ &+ \Big|\Big\{u \in V(G) : \alpha^-(u) \in \big[1, \alpha^-(v)\big) \vee \big(\alpha^-(u) = \alpha^-(v) \wedge \alpha^+(u) \in \alpha(v)\big)\Big\}\Big|.\end{aligned}$$

As there can be more than one vertex v such that $\alpha^-(v) = 1$, let v_0 denote such a vertex with the shortest arc $\alpha(v_0)$. It is straightforward to check that $\beta^-(v_0) = 1$. The definition of $\beta^-(v)$

ensures that the arc $[1, \beta^-(v)]$ is exactly long enough to contain

- the left extreme vertices $\beta^-(u)$ for all $u \in [v_0, v]$ and
- the right extreme vertices $\beta^+(u)$ for all $u \in [v_0^-, v^-)$,

where the arc notation on $V(G)$ is with respect to $\prec_\alpha$ (recall that $N[v] = [v^-, v^+]$). Since the arc $\beta(v)$ has to contain exactly one extreme point of every neighbor of v, the right extreme point of $\beta(v)$ is determined by

$$\beta^+(v) = \beta^-(v) + |N(v)| + 1;$$

whenever the right hand side exceeds $2n$, it has to be decreased by this number.

This completes the proof of Theorem 5.1, and we have additionally proved the following corollary.

Corollary 5.2. *The canonical arc representation problem for co-convex graphs is solvable in logspace.*

6 Solving the Star System Problem

In this section, we present logspace algorithms for the Star System Problem: Given a hypergraph $\mathcal{H}$, find a graph G in a specified graph class C such that $\mathcal{N}[G] = \mathcal{H}$ (if such a graph exists). The term *star* refers to the closed neighborhood of a vertex in G. In this terminology, the problem is to identify the center of each star H in the star system $\mathcal{H}$. To denote this problem, we use the abbreviation *SSP*. Note that a logspace algorithm $\mathcal{A}$ solving the SSP for a class C cannot be directly used for solving the SSP for a subclass C' of C: If $\mathcal{A}$ on input $\mathcal{H}$ outputs a solution G in $\mathsf{C} \setminus \mathsf{C}'$, then we don't know whether there is another solution G' in C'. However, if the SSP for C has unique solutions and if membership in C' is decidable in logspace, then it is easy to convert $\mathcal{A}$ into a logspace algorithm $\mathcal{A}'$ solving the SSP for C'.

Theorem 6.1.

1. *The SSP for PCA and for co-convex graphs is solvable in logspace.*
2. *If G is a co-convex graph, then $\mathcal{N}[G] \cong \mathcal{N}[G']$ implies $G \cong G'$.*

The implication stated in Theorem 6.1.2 is known to be true also for concave-round graphs (Chen [7]). As a consequence, since concave-round graphs form a logspace decidable subclass of the union of PCA and co-convex graphs, we can also solve the SSP for concave-round graphs in logspace.

The proof of Theorem 6.1 is given in the rest of this section. We design logspace algorithms $\mathcal{A}_1$ and $\mathcal{A}_2$ solving the SSP for non-co-bipartite PCA graphs and for co-convex graphs, respectively. Since by Theorem 6.1.2, the output of $\mathcal{A}_2$ is unique up to isomorphism, we can easily combine the two algorithms to obtain a logspace algorithm $\mathcal{A}_3$ solving the SSP for all PCA graphs: On input $\mathcal{H}$ run $\mathcal{A}_1$ and $\mathcal{A}_2$ and check if one of the resulting graphs is PCA (recall that co-bipartite PCA graphs are co-convex; see Fig. 2).

Clearly, it suffices to consider the case that the input hypergraph $\mathcal{H}$ is connected.

6.1 Non-co-bipartite PCA graphs

Let $\mathcal{H}$ be the given input hypergraph and assume that $\mathcal{H} = \mathcal{N}[G]$ for a PCA graph G. By Theorem 4.1, $\mathcal{H}$ has to be a tight CA hypergraph, a condition that can be checked by testing if the tightened hypergraph $\mathcal{H}^{\Subset}$ is CA. Since G is concave-round, Proposition 4.4 implies that G is co-bipartite if and only if $\mathcal{N}(\overline{G}) = \overline{\mathcal{H}}$ is an interval hypergraph. It follows that the SSP on $\mathcal{H}$ can only have a non-co-bipartite PCA graph as solution if $\mathcal{H}^{\Subset}$ is CA and $\overline{\mathcal{H}}$ is not interval. Both conditions can be checked in logspace using the algorithms given by Theorem 3.1 and [21]. Further, it follows by Theorem 4.1 and Proposition 4.4 that in this case any SSP solution for $\mathcal{H}$ is a non-co-bipartite PCA graph (which is also connected because $\mathcal{H}$ is assumed to be connected).

Assume first that the hypergraph $\mathcal{H}$ is twin-free. In order to reconstruct G from $\mathcal{H}$, we have to choose the center in each star $H \in \mathcal{H}$. The following lemma considerably restricts this choice.

Lemma 6.2. *Let G be a connected, non-co-bipartite and twin-free PCA graph and let $\prec$ be a circular order on $V(G)$ that is a CA order of $\mathcal{N}[G]$. Then $u \prec v$ holds exactly when $N[u] \prec_{\mathcal{N}[G]} N[v]$, where $\prec_{\mathcal{N}[G]}$ is the circular order on $\mathcal{N}[G]$ lifted from $\prec$.*

Proof. First of all, note that the circular order $\prec_{\mathcal{N}[G]}$ on $\mathcal{N}[G]$ is correctly defined because a non-co-bipartite PCA graph has no universal vertex (we observed this fact in Section 5). By the same reason we can use the notation $N[x] = [x^-, x^+]$ with respect to $\prec$.

By Lemma 4.2 and Proposition 4.3, there is a proper arc representation α of G such that $\prec$ coincides with the associated geometric order $\prec_\alpha$. We will use the following fact observed in the proof of Lemma 4.2. Suppose that x and y are adjacent vertices of G. Then the vertices x, y, y^+, x^+ never appear in this sequence (not necessary consecutively) with respect to $\prec$ unless $x^+ = y^+$ and, similarly, the vertices x^-, y^-, y, x never appear in this sequence unless $x^- = y^-$.

To prove the lemma, it suffices to show that $u \prec v$ implies $N[u] \prec_{\mathcal{N}[G]} N[v]$. To this end we show that there is no third vertex w such that the arcs $N[u]$, $N[w]$, and $N[v]$ appear in this sequence under the circular order $\prec_{\mathcal{N}[G]}$.

Suppose first that u and v are adjacent. Then the vertices u^-, v^-, u, v, u^+, and, v^+ appear in this circular sequence, as shown Fig. 4(a) (though it is not excluded that some of these vertices can coincide, for example, $u^- = v^-$ or $u^+ = v^+$). We split our analysis into three cases, depending on the position of w on the cycle $(V(G), \prec)$. If $w \in (v, v^+]$, then $w^- \in [v^-, v]$. If $w^- \neq v^-$, then $N[u]$, $N[v]$, and $N[w]$ clearly appear in this sequence under $\prec_{\mathcal{N}[G]}$. The same holds true if $w^- = v^-$ because then the arc $[w^-, w^+]$ has to be longer than the arc $[v^-, v^+]$. If $w \in [u^-, u)$, then u^- must belong to $[w^-, w]$. $N[w]$, $N[u]$, and $N[v]$ obviously appear in this sequence if $w^- \neq u^-$. This is also true if $w^- = u^-$ because $[w^-, w^+]$ must be shorter than $[u^-, u^+]$ in this case. If $w \in (v^+, u^-)$, then $w^- \in (v, u^-)$, and again $N[w]$ cannot be intermediate.

Figure 4: Two cases in the proof of Lemma 6.2.

Suppose now that u and v are not adjacent. It follows that $N[u] = [u^-, u]$ and $N[v] = [v, v^+]$; see Fig. 4(b). Therefore, both $N[u]$ and $N[v]$ are cliques. Again we have to show that for no third vertex w, the arcs $N[u]$, $N[w]$, and $N[v]$ appear in this sequence under $\prec_{\mathcal{N}[G]}$. This is clear if $w^- \in (v, u^-)$. This is also so if $w^- = v$, because then the arc $[v, v^+]$ must be shorter than the arc $[w^-, w^+]$. Finally, note that the remaining case $w^- \in [u^-, v)$ is not possible. Indeed, in this case $v \notin N[w]$, for else the non-adjacent vertices u and v would belong to the clique $[w, w^+]$. Hence, it would follow that $N[w] = [w^-, w^+] \subsetneq [u^-, u^+] = N[u]$, contradicting the fact that $N[u]$ is a clique. □

Lemma 6.2 states that the mapping $v \mapsto N[v]$ is an isomorphism between the two directed cycles $(V(G), \prec)$ and $(\mathcal{N}[G], \prec_{\mathcal{N}[G]})$. Since there are exactly n such isomorphisms, we get exactly n candidates $f_1, \ldots, f_n$ for the mapping $v \mapsto N[v]$. Hence, all we have to do is to use the algorithm given by Theorem 3.1 to compute a CA order $\prec$ of $\mathcal{H}$ and the corresponding lifted order $\prec_{\mathcal{H}}$ in logspace. Now for each isomorphism f between $(V(\mathcal{H}), \prec)$ and $(\mathcal{H}, \prec_{\mathcal{H}})$ we have to check if selecting v as the center of the star $f(v)$ results in a graph G, that is, if for all $v, u \in V(\mathcal{H})$ it holds that $v \in f(v)$ and that $v \in f(u)$ exactly when $u \in f(v)$.

This completes the solution in the case that the input hypergraph $\mathcal{H}$ is twin-free. We will also need the fact that the SSP has at most one solution in this case. Note that the aforementioned result by Chen [7] ensures the uniqueness up to isomorphism. In order to prove the uniqueness up to equality, let $v_0, v_1, \ldots, v_{n-1}$ be the list of the vertices according to a circular order $\prec$, and suppose that the assignment $v_i \mapsto H_i$ corresponds to some non-co-bipartite PCA graph G, that is, $H_i = N_G[v_i]$. Let $H_i = [v_i^-, v_i^+]$, where the arc notation is with respect to $\prec$. We can orient each edge $\{v_i, v_j\}$ of G as the ordered pair $v_i v_j$ if $v_j \in (v_i, v_i^+]$ and $v_j v_i$ if $v_j \in [v_i^-, v_i)$. This rule is consistent because $\prec$ is equal to $\prec_\alpha$ for some proper arc representation α of G, and then an edge $\{v_i, v_j\}$ is oriented as $v_i v_j$ exactly when $\alpha(v_j)$ contains the right extreme point of $\alpha(v_i)$. Note that this is a *round orientation* of G as defined by Deng, Hell, and Huang [11]. The sum of outdegrees over the vertices of G is equal to $\sum_{i=0}^{n-1} \left|(v_i, v_i^+]\right|$, while the sum of indegrees is equal to $\sum_{i=0}^{n-1} \left|[v_i^-, v_i)\right|$. As a general fact about graph orientations, the two sums must be equal, each being equal to the number of edges in G.

We now show that the other $n-1$ possible assignments $v_i \mapsto H_{i+k}$, where $1 \le k < n$ and the addition is modulo n, do not correspond to any graph. Fix k and suppose that $v_i \in H_{i+k}$ for all i; otherwise the claim is clear. Assume that $v_j \in [v_{j+k}^-, v_{j+k})$ for some j (the case that $v_j \in (v_{j+k}, v_{j+k}^+]$ for some j is symmetric). This implies that $v_{j+1} \in [v_{j+1+k}^-, v_{j+1+k})$. Indeed, if $v_{j+1} \in (v_{j+1+k}, v_{j+1+k}^+]$, then the arcs $[v_{j+k}^-, v_{j+k}]$ and $[v_{j+1+k}, v_{j+1+k}^+]$ would cover the entire vertex set, yielding a covering of the graph G by two cliques. We conclude that $v_i \in [v_{i+k}^-, v_{i+k})$ for all i. This implies that $\left|(v_i, v_{i+k}^+]\right| > \left|(v_{i+k}, v_{i+k}^+]\right|$ and $\left|[v_{i+k}^-, v_i)\right| < \left|[v_{i+k}^-, v_{i+k})\right|$ for every i. It follows that there is no G' such that $N_{G'}(v_i) = H_{i+k}$ for all i as the equality of the sums of in- and outdegrees for the round orientation of such G' would be violated (recall that any such G' must be a non-co-bipartite PCA graph). Therefore, the SSP on $\mathcal{H}$ can have no other solution than G.

We are now ready to solve the SSP in the case that the input hypergraph $\mathcal{H}$ has twins. If $\mathcal{H} = \mathcal{N}[G]$, then any two vertices that are twins in $\mathcal{H}$ are also twins in G. Therefore, $\mathcal{H}$ must have multi-hyperedges for else the SSP has no solution. Let $\mathcal{H}'$ be obtained from $\mathcal{H}$ by removing twins (more precisely, in each twin-class we leave a single vertex) and by removing multi-hyperedges. However, for each $H \in \mathcal{H}'$ we will remember the multiplicity of the corresponding hyperedge in $\mathcal{H}$, denoting it by $\mu(H)$. Moreover, for a vertex v of $\mathcal{H}'$, let $\nu(v)$ denote the number of twins that v has in $\mathcal{H}$ (including itself). Given a graph G, denote its twin-free version by G'. Note that $\mathcal{H} = \mathcal{N}[G]$ implies $\mathcal{H}' = \mathcal{N}[G']$ (assuming that the twin-classes in $\mathcal{H}$ in G are reduced in the same way). The converse implication is true if and only if $\nu(v) = \mu(N_{G'}[v])$ for all $v \in V(G')$. This yields the following reduction of the SSP to the twin-free case, that works for any class of graphs C that, like the class of non-co-bipartite PCA graphs, has the following two properties:

- C is closed under removing and adding twins,
- whenever the SSP has a solution in C, it is unique.

Given a hypergraph $\mathcal{H}$ with twins, we construct $\mathcal{H}'$ and compute the functions $\nu\colon V(\mathcal{H}') \to \mathbb{Z}$ and $\mu\colon \mathcal{H}' \to \mathbb{Z}$ as defined above. Then we try to find a graph G' in C such that $\mathcal{H}' = \mathcal{N}[G']$. If such a graph G' does not exist, the SSP has no solution also on $\mathcal{H}$. If such a G' exists, it is unique, and there is no other way to get a solution G on $\mathcal{H}$ than to clone each vertex v of G' with multiplicity $\nu(v)$ (more precisely, the restored twin-class of v in G must be equal to the twin-class of v in $\mathcal{H}$). The graph G satisfies the equality $\mathcal{H} = \mathcal{N}[G]$ if $\nu(v) = \mu(N_{G'}[v])$ for all $v \in V(G')$; otherwise there is no solution. This reduction is clearly implementable in logspace.

6.2 Co-convex graphs

Let $\mathcal{H}$ be the given hypergraph and assume that $\mathcal{H} = \mathcal{N}[G]$ for a co-convex graph G. To facilitate the exposition, suppose first that the bipartite complement $\overline{G}$ is connected. Recall that the vertex set of a connected bipartite graph is uniquely split into two independent sets; they are referred to as the *vertex classes* of the graph. Denote the vertex classes of $\overline{G}$ by U and W. Then $\overline{\mathcal{H}} = \mathcal{N}(\overline{G}) = \mathcal{N}_U(\overline{G}) \cup \mathcal{N}_W(\overline{G})$, where the vertex-disjoint hypergraphs $\mathcal{U} = \mathcal{N}_U(\overline{G})$ and $\mathcal{W} = \mathcal{N}_W(\overline{G})$ are dual (i.e., $\mathcal{U}^* \cong \mathcal{W}$), both connected, and at least one of them is interval, say, $\mathcal{U}$. Note also that, since $\overline{G}$ is connected, $\overline{\mathcal{H}}$ has no isolated vertex, that is, every vertex is contained in some hyperedge. We need a simple auxiliary fact.

Lemma 6.3. *Let K be a graph without isolated vertices and let $\mathcal{L}$ be a connected component of $\mathcal{N}(K)$. Denote $U = V(\mathcal{L})$. Then either U is an independent set in K or U induces 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 $\mathcal{L}$, 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 $\mathcal{L}$ would be disconnected.

Assume now that U is independent in K. Consider a vertex $u \in U$ and a vertex w adjacent to u in K. Let $\mathcal{L}'$ be the connected component of $\mathcal{N}(K)$ containing w. As shown above, the set of vertices $W = V(\mathcal{L}')$ is independent in K (otherwise W would contain u). By connectedness of $\mathcal{L}$ and $\mathcal{L}'$, once we have an edge uw between U and W, we have $N_K(w) \subseteq U$ and $N_K(u) \subseteq W$. 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 $\mathcal{L}$ and $\mathcal{L}'$. □

Denote $\mathcal{K} = \overline{\mathcal{H}}$ and assume that $\mathcal{K} = \mathcal{N}(K)$ for some graph K, possibly different from $\overline{G}$. Since $\mathcal{K}$ has no isolated vertex, K also has none. Lemma 6.3 implies either that K is a connected bipartite graph with vertex classes U, W or that K has two connected components K_1 and K_2 with $V(K_1) = U$ and $V(K_2) = W$. However, the second possibility leads to a contradiction. Indeed, since the hypergraph $\mathcal{N}(K_1) = \mathcal{U}$ is interval, Proposition 4.4 implies that K_1 is bipartite, contradicting the connectedness of $\mathcal{U}$. Therefore, K must be connected and bipartite with vertex classes U, W.

Recall that the *incidence graph* of a hypergraph $\mathcal{X}$ is the bipartite graph with vertex classes $V(\mathcal{X})$ and $\mathcal{X}$ where $x \in V(\mathcal{X})$ and $X \in \mathcal{X}$ are adjacent if $x \in X$ (if X has multiplicity k in $\mathcal{X}$, it contributes k fraternal vertices in the incidence graph). Since K is isomorphic to the incidence graph of the hypergraph $\mathcal{U}$ (as well as $\mathcal{W}$), K is reconstructible from $\mathcal{K}$ up to isomorphism and, in particular, $K \cong \overline{G}$. Thus, the solution to the SSP on $\mathcal{H}$ is unique up to isomorphism.[1]

After these considerations we are ready to describe our logspace algorithm for 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 $\mathcal{U}$ and $\mathcal{W}$. This can be done by running Reingold's reachability algorithm [32] on the intersection graph $\mathbb{I}(\overline{\mathcal{H}})$. If this is not the case, there is no solution in the desired class. Otherwise, we construct the incidence graph F of the hypergraph $\mathcal{U}$ (or of $\mathcal{W}$, which should give the same result up to isomorphism) and take its complement $\overline{F}$. Note that this works well even if $\overline{F}$ has twins: the twins in $V(\mathcal{U})$ are explicitly present, while the twins in $V(\mathcal{W})$ are represented by multi-hyperedges in $\mathcal{U}$.

As argued above, if the SSP on $\mathcal{H}$ has a co-convex co-connected solution, then the closed neighborhood hypergraph $\mathcal{F} = \mathcal{N}[\overline{F}]$ of $\overline{F}$ is isomorphic to $\mathcal{H}$. However, it may not be equal to $\mathcal{H}$. In this case we compute an isomorphism φ from $\mathcal{F}$ to $\mathcal{H}$ or, the same task, from $\overline{\mathcal{F}}$ to $\overline{\mathcal{H}}$. This can be done by the algorithms of [21] and Corollary 3.3, because at least one of the connected components of $\overline{\mathcal{F}} \cong \overline{\mathcal{H}}$ is an interval hypergraph and the other component is isomorphic to the dual of an interval hypergraph. Now, the isomorphic image $G = \varphi(\overline{F})$ of $\overline{F}$ is the desired solution to the SSP on $\mathcal{H}$ as $\mathcal{N}[\varphi(\overline{F})] = \varphi(\mathcal{N}[\overline{F}]) = \mathcal{H}$.

If we do not succeed with establishing an isomorphism between $\mathcal{F}$ 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 $\mathcal{U}$ and $\mathcal{W}$ is interval and $\mathcal{U}^* \cong \mathcal{W}$.

Consider now 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 first have to check that the number of such vertices is equal to the multiplicity of the hyperedge $V(\mathcal{H})$ in $\mathcal{H}$ for else the SSP has no solution on $\mathcal{H}$. If the two numbers are equal, we can remove all universal vertices from $\mathcal{H}$ along with all copies of the hyperedge $V(\mathcal{H})$, 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.

If $\overline{G}$ consists of k connected components $H_1, \ldots, H_k$, where H_i is a bipartite graph with vertex classes U_i and W_i, then $\mathcal{K} = \overline{\mathcal{H}}$ consists of $2k$ connected components $\mathcal{U}_i = \mathcal{N}_{U_i}(H_i)$ and $\mathcal{W}_i = \mathcal{N}_{W_i}(H_i)$, each pair being dual. Moreover, it can be supposed that all $\mathcal{U}_i$ are interval hypergraphs.

Assume that $\mathcal{K} = \mathcal{N}(K)$ for any other graph K. By Lemma 6.3, for each connected component $\mathcal{L} \in \{\mathcal{U}_i, \mathcal{W}_i\}_{i=1}^k$ either $V(\mathcal{L}) \in \{U_i, W_i\}_{i=1}^k$ induces a connected component of K or there is another connected component $\mathcal{L}'$ such that $V(\mathcal{L}) \cup V(\mathcal{L}')$ induces a connected component of K

[1]The uniqueness result of Boros et al. [5] implies a somewhat weaker fact, namely the uniqueness up to isomorphism *within* the class of co-convex graphs.

that is a bipartite graph. Note that in the latter case $\mathcal{L}$ and $\mathcal{L}'$ have to be dual hypergraphs, i.e., $\mathcal{L}' \cong \mathcal{L}^*$. Recall that, by Proposition 4.4, no U_i can alone induce a connected component of K. It readily follows that K consists of k connected bipartite components $K_1, \ldots, K_k$, where the vertex classes Y_i and Z_i of each K_i induce connected components of $\mathcal{K}$. Moreover, we can enumerate $K_1, \ldots, K_k$ so that the components of $\mathcal{K}$ induced by Y_i and Z_i are isomorphic to $\mathcal{U}_i$ and $\mathcal{W}_i$. Since both H_i and K_i are isomorphic to the incidence graph of the hypergraph $\mathcal{U}_i$ (as well as $\mathcal{W}_i$), the graphs K and $\overline{G}$ are isomorphic and the solution to the SSP on $\mathcal{H}$ 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 $\mathcal{H}$, we first check if $\overline{\mathcal{H}}$ has an even number of connected components that can be split into pairs $\mathcal{U}_i$ and $\mathcal{W}_i$ so that $\mathcal{U}_i$ is an interval hypergraph and $\mathcal{W}_i \cong \mathcal{U}_i^*$. This step can be done by using Reingold's algorithm and the algorithm of [21]. A desired solution exists if and only if this is possible.

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

Then, for each i, we construct the incidence graph F_i of the hypergraph $\mathcal{U}_i$, form the graph F as the vertex-disjoint union of all F_i, and take its complement $\overline{F}$.

By the already established uniqueness, the closed neighborhood hypergraph $\mathcal{F} = \mathcal{N}[\overline{F}]$ is isomorphic to $\mathcal{H}$. We find an isomorphism φ from $\mathcal{F}$ to $\mathcal{H}$ or, the same, from $\overline{\mathcal{F}}$ to $\overline{\mathcal{H}}$. We do it componentwise by running the algorithms of [21] and Corollary 3.3 on the connected components of $\overline{\mathcal{F}}$ and $\overline{\mathcal{H}}$. The isomorphic image $G = \varphi(\overline{F})$ is a solution as $\mathcal{N}[\varphi(\overline{F})] = \varphi(\mathcal{N}[\overline{F}]) = \mathcal{H}$.

7 Concluding remarks

By Theorem 5.1, there is a logspace algorithm that solves the canonical arc representation problem for PCA graphs, where the constructed models are proper. *Unit CA graphs* are CA graphs that admit a PCA model where all arcs have equal length. The unit arc representation problem for such graphs can be solved in linear time [28, 20]. In a previous version of this article we asked whether it can also be solved in logspace. Soulignac answered this positively [35], employing our PCA representation algorithm as a subroutine. The *unit interval* representation problem is solved in logspace in [21].

In Section 6, we solve the Star System Problem for PCA graphs and concave-round graphs in logspace. Is this also possible for other classes of circular-arc graphs? Furthermore, can one extend the result of Theorem 6.1.2 about the uniqueness of a solution to this problem?

In analogy to convex graphs, Liang and Blum [26] call a bipartite graph G with vertex classes U and V *circular convex*, if $\mathcal{N}_U(G) = \{N_G(u)\}_{u \in U}$ is a CA hypergraph. We remark that our logspace algorithm for canonical representation of CA hypergraphs can be used to solve the canonical labeling problem for circular convex graphs in logspace. Indeed, in [21] we converted a canonical representation algorithm for interval hypergraphs into a canonical labeling algorithm for convex graphs, and this approach can be easily adapted to the circular-arc setting.

Acknowledgement We thank Bastian Laubner for useful discussions at the early stage of this work and an anonymous referee for numerous helpful comments on the paper.

References

References

[1] Aigner, M., Triesch, E.: Reconstructing a graph from its neighborhood lists. Combinatorics, Probability & Computing **2**, 103–113 (1993)

[2] Annexstein, F.S., Swaminathan, R.P.: On testing consecutive-ones property in parallel. Discrete Applied Mathematics **88**(1-3), 7–28 (1998)

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

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

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

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

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

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

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

[10] Curtis, A.R., Lin, M.C., McConnell, R.M., Nussbaum, Y., Soulignac, F.J., Spinrad, J.P., Szwarcfiter, J.L.: Isomorphism of graph classes related to the circular-ones property. Discrete Mathematics & Theoretical Computer Science **15**(1), 157–182 (2013)

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

[12] Diestel, R.: Graph theory. 4th ed. Berlin, Springer (2010)

[13] Fomin, F.V., Kratochvíl, J., Lokshtanov, D., Mancini, F., Telle, J.A.: On the complexity of reconstructing H-free graphs from their Star Systems. Journal of Graph Theory **68**(2), 113–124 (2011)

[14] Gavril, F., Pinter, R.Y., Zaks, S.: Intersection representations of matrices by subtrees and unicycles on graphs. Journal of Discrete Algorithms **6**(2), 216–228 (2008)

[15] Hell, P., Huang, J.: Lexicographic orientation and representation algorithms for comparability graphs, proper circular arc graphs, and proper interval graphs. J. Algorithms **20**(3), 361–374 (1995)

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

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

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

[19] Huang, J.: On the structure of local tournaments. Journal of Combinatorial Theory **24**(3), 411–439 (1995)

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

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

[22] Köbler, J., Kuhnert, S., Verbitsky, O.: Around and beyond the isomorphism problem for interval graphs. Bulletin of the EATCS **107**, 44–71 (2012)

[23] Köbler, J., Kuhnert, S., Verbitsky, O.: Solving the canonical representation and star system problems for proper circular-arc graphs in logspace. In: Proceedings of the IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, no. 18 in LIPIcs, pp. 387–399. Leibniz-Zentrum für Informatik, Dagstuhl (2012)

[24] Köbler, J., Kuhnert, S., Verbitsky, O.: Circular-arc hypergraphs: Rigidity via connectedness. E-print: `http://arxiv.org/abs/1312.1172` (2013)

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

[26] Liang, Y.D., Blum, N.: Circular convex bipartite graphs: Maximum matching and hamiltonian circuits. Inform. Proc. Lett. **56**(4), 215–219 (1995)

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

[28] Lin, M.C., Szwarcfiter, J.L.: Unit circular-arc graph representations and feasible circulations. SIAM J. on Discrete Mathematics **22**(1), 409–423 (2008)

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

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

[31] Ouangraoua, A., Bergeron, A., Swenson, K.M.: Theory and practice of ultra-perfection. Journal of Computational Biology **18**(9), 1219–1230 (2011)

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

[33] Roberts, F.S.: On the compatibility between a graph and a simple order. Journal of Combinatorial Theory, Series B **11**(1), 28–38 (1971)

[34] Skrien, D.J.: A relationship between triangulated graphs, comparability graphs, proper interval graphs, proper circular-arc graphs, and nested interval graphs. Journal of Graph Theory **6**(3), 309–316 (1982)

[35] Soulignac, F.J.: Minimal and short representations of unit interval and unit circular-arc graphs. E-print: `http://arxiv.org/abs/1408.3443v2` (2014)

[36] Soulignac, F.J.: Fully dynamic recognition of proper circular-arc graphs. Algorithmica **71**(4), 904–968 (2015)

[37] Spinrad, J.P.: Efficient graph representations. No. 19 in Field Institute Monographs. AMS (2003)

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

[39] Tucker, A.: Structure theorems for some circular-arc graphs. Discrete Mathematics **7**, 167–195 (1974)