

Recognizing H -Graphs – Beyond Circular-Arc Graphs

Deniz Ağaoğlu Çağırıcı ✉

Faculty of Informatics, Masaryk University,
Brno, Czech Republic

Jan Derbisz ✉

Theoretical Computer Science Department,
Faculty of Mathematics and Computer Science,
Jagiellonian University, Kraków, Poland
Doctoral School of Exact and Natural Sciences,
Jagiellonian University, Kraków, Poland

Petr Hliněný ✉ 

Faculty of Informatics, Masaryk University,
Brno, Czech Republic

Tomasz Krawczyk ✉ 

Theoretical Computer Science Department,
Faculty of Mathematics and Computer Science,
Jagiellonian University, Kraków, Poland

Onur Çağırıcı ✉

Toronto Metropolitan University, Canada

Tim A. Hartmann ✉ 

CISPA Helmholtz Center for Information Security,
Saarbrücken, Germany

Jan Kratochvíl ✉

Department of Applied Mathematics, Faculty of
Mathematics and Physics, Charles University,
Prague, Czech Republic

Peter Zeman ✉

Technical University of Denmark,
Lyngby, Denmark

Abstract

In 1992 Biró, Hujter and Tuza introduced, for every fixed connected graph H , the class of H -graphs, defined as the intersection graphs of connected subgraphs of some subdivision of H . Such classes of graphs are related to many known graph classes: for example, K_2 -graphs coincide with interval graphs, K_3 -graphs with circular-arc graphs, the union of T -graphs, where T ranges over all trees, coincides with chordal graphs. Recently, quite a lot of research has been devoted to understanding the tractability border for various computational problems, such as recognition or isomorphism testing, in classes of H -graphs for different graphs H .

In this work we undertake this research topic, focusing on the recognition problem. Chaplick, Töpfer, Voborník, and Zeman showed an XP-algorithm testing whether a given graph is a T -graph, where the parameter is the size of the tree T . In particular, for every fixed tree T the recognition of T -graphs can be solved in polynomial time. Tucker showed a polynomial time algorithm recognizing K_3 -graphs (circular-arc graphs). On the other hand, Chaplick et al. showed also that for every fixed graph H containing two distinct cycles sharing an edge, the recognition of H -graphs is NP-hard.

The main two results of this work narrow the gap between the NP-hard and P cases of H -graph recognition. First, we show that the recognition of H -graphs is NP-hard when H contains two distinct cycles. On the other hand, we show a polynomial-time algorithm recognizing L -graphs, where L is a graph containing a cycle and an edge attached to it (which we call *lollipop graphs*). Our work leaves open the recognition problems of M -graphs for every unicyclic graph M different from a cycle and a lollipop.

2012 ACM Subject Classification Theory of computation → Graph algorithms analysis

Keywords and phrases H-graphs, Intersection Graphs, Helly Property

Digital Object Identifier 10.4230/LIPIcs.MFCS.2023.8

Related Version *Full Version*: <https://arxiv.org/abs/2212.05433>

Funding *Deniz Ağaoğlu Çağırıcı*: Research of this author is supported by the Czech Science Foundation project no. 20-04567S.

Jan Derbisz: Research of this author was partially funded by Polish National Science Center (NCN) grant 2021/41/N/ST6/03671 and by the Priority Research Area SciMat under the program Excellence Initiative Research University at the Jagiellonian University in Kraków.



© Deniz Ağaoğlu Çağırıcı, Onur Çağırıcı, Jan Derbisz, Tim A. Hartmann, Petr Hliněný, Jan Kratochvíl, Tomasz Krawczyk, and Peter Zeman;
licensed under Creative Commons License CC-BY 4.0

48th International Symposium on Mathematical Foundations of Computer Science (MFCS 2023).

Editors: Jérôme Leroux, Sylvain Lombardy, and David Peleg; Article No. 8; pp. 8:1–8:14



Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Petr Hliněný: Research of this author is supported by the Czech Science Foundation project no. 20-04567S.

Peter Zeman: Research of this author is supported by the Carlsberg Foundation Young Researcher Fellowship CF21-0682 - “Quantum Graph Theory”.

1 Introduction

One of the most important and most widely studied types of graph representations is an *intersection model*, in which the vertices are represented by sets and the edges by the pairs of intersecting sets. Due to practical and theoretical applications, intersection graphs of simple geometric objects are among the most intensively studied. In this paper, we consider a special kind of intersection graphs, called *H -graphs*, introduced by Biró, Hujter and Tuza [5]. Since H -graphs generalize many known geometric intersection graph classes, they form a good background that allows to study basic computational problems in some systematic way. We first define H -graphs formally using the terminology we adapt throughout the paper.

Let H be a connected graph. An *H -model* of a graph G is a pair (H^ϕ, ϕ) , where H^ϕ is a subdivision of H and ϕ is a mapping from $V(G)$ to the subsets of $V(H^\phi)$, such that:

- for every $v \in V(G)$, the subgraph of H^ϕ induced by the set $\phi(v)$ is connected,
- for every distinct $u, v \in V(G)$ we have $uv \in E(G)$ iff $\phi(u) \cap \phi(v) \neq \emptyset$.

A graph G is an *H -graph* if G admits an H -model. In particular, every graph G is an H -graph for some graph H , e.g., for $H = G$.

Many known geometric intersection graph classes are H -graphs for an appropriately chosen graph H or are \mathcal{H} -graphs for some simpler class of graphs \mathcal{H} , where by *\mathcal{H} -graphs* we mean the union of the classes of H -graphs over $H \in \mathcal{H}$:

- K_2 -graphs coincide with the class of *interval graphs*, which are defined as the intersection graphs of intervals on the line,
- K_3 -graphs coincide with the class of *circular-arc graphs*, which are defined as the intersection graphs of arcs of a fixed circle,
- \mathcal{T} -graphs, where \mathcal{T} contains all trees, coincide with the class of *chordal graphs*, which are defined as graphs containing no induced cycles of size ≥ 4 [14],
- \mathcal{P} -graphs, where \mathcal{P} contains all planar graphs, coincide with the class of *string graphs*, which are defined as the intersection graphs of curves in the plane.

The recent research on H -graphs, initiated by Chaplick et al. [10], aims to generalize efficient optimization algorithms from simple classes of graphs on wider families of H -graphs, as well as to determine the boundary of “polynomial tractability” for such computational problems as recognition or isomorphism testing. Here we aim for efficient *parameterized algorithms*, whose running time depends on the size n of the input graph and the parameter $|H|$, where $|H|$ is the size of the graph H . First, we search for algorithms that work in polynomial time in n and $|H|$, then for FPT *algorithms* working in time $f(|H|)n^{O(1)}$ for some computable function f , and finally for XP *algorithms* working in time $O(n^{f(|H|)})$ for some computable function f . Various NP-complete problems on H -graphs were studied in the parameterized setting and shown to admit FPT and XP algorithms, e.g., [1, 2, 4, 8, 10, 11, 13, 16]. Some recent research is also focused on studying the combinatorial properties of H -graphs, which can be later used to construct efficient algorithms in these classes of graphs (see e.g. [11, 13]).

In this work we are focusing on the recognition problem. For a graph class \mathcal{G} , the *recognition problem* for \mathcal{G} is to decide whether an input graph G belongs to \mathcal{G} . For a graph class \mathcal{G} defined in a geometrical way, the recognition problem of \mathcal{G} usually boils down to testing whether the input graph has a representation appropriate for the class \mathcal{G} . There are known linear time recognition algorithms for interval graphs [6] and chordal graphs [21].

In the context of our research, circular-arc graphs form an important class of graphs. The recognition problem for circular-arc graphs was initially thought to be NP-hard [7], but since the 1980s we already know that it admits a polynomial time algorithm [22]. Currently, two linear-time algorithms recognizing circular-arc graph are known [17,20], but both of them are rather lengthy and non-trivial. No simple polynomial-time algorithm recognizing circular-arc graphs is known.

Although the recognition of chordal graphs takes linear time, with a tree T on the input deciding whether a graph G is a T -graph is NP-complete [18]. On the other hand, Chaplick et al. [10] gave an XP algorithm parameterized by $|T|$ recognizing T -graphs. It is open whether the problem can be solved by an FPT algorithm (in [9] it is shown that proper T -graphs can be recognized in FPT, where a T -graph G is called *proper* if there exists a T -model (T_ϕ, ϕ) of G such that for no pair $u, v \in V(G)$ we have $\phi(u) \subseteq \phi(v)$). Moreover, Chaplick et al. [10] showed that recognition of H -graphs is NP-complete if H contains a diamond (a cycle on four vertices with a chord) as a minor [10]. That is, recognition of H -graphs is NP-complete for every fixed H which contains two distinct cycles sharing an edge.

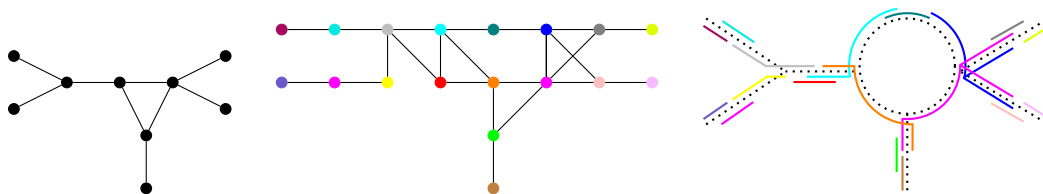
1.1 Our results

Our first result states the following, which extends the hardness result from [10]:

► **Theorem 1.1.** *For every fixed graph H containing two distinct cycles, the recognition of H -graphs is NP-complete.*

Theorem 1.1 raises interests in M -graphs, where M is a *unicyclic* graph (a connected graph containing exactly one cycle). In particular, we are focusing on:

- the recognition problem for the class of M -graphs, where M is any fixed graph that consists of a cycle and some trees attached to it,
- the recognition problem for the class of *medusa graphs*, which are defined as \mathcal{M} -graphs, where \mathcal{M} is the class that contains all unicyclic graphs. Note that medusa graphs extend both circular-arc graphs and chordal graphs.



■ **Figure 1.1** From left to right: a unicyclic graph M , an M -graph G , an M -model (M^ϕ, ϕ) of G .

An \mathcal{M} -model of a graph G is an M -model of G where $M \in \mathcal{M}$. Suppose G is a medusa graph and suppose G admits an \mathcal{M} -model (M^ϕ, ϕ) for some $M \in \mathcal{M}$. A clique C in G satisfies the *Helly property* in (M^ϕ, ϕ) if $\bigcap_{c \in C} \phi(c) \neq \emptyset$, and the model (M^ϕ, ϕ) of G satisfies the *Helly property* if every clique of G satisfies the Helly property in (M^ϕ, ϕ) . A medusa graph G is *Helly* if G admits an \mathcal{M} -model that satisfies the Helly property. Figure 1.1 shows a fixed unicyclic graph M , another graph G which is an M -graph, and an M -model (M^ϕ, ϕ) of G . Since (M^ϕ, ϕ) satisfies the Helly property, G is a Helly medusa graph. We show the following regarding medusa graph:

► **Theorem 1.2.**

1. *The problem of recognizing medusa graphs is NP-complete.*
2. *The problem of recognizing Helly medusa graphs is polynomial time solvable.*

Our most important (and perhaps most difficult) result concerns the class of L -graphs (which we call *lollipop graphs*), where L is a unicyclic graph that consists of a cycle with an edge attached. Note that L -graphs extend the class of circular-arc graphs.

► **Theorem 1.3.** *The problem of recognizing L -graphs is polynomial time solvable.*

Our research reveals connections between problems we consider and certain problems related to the Helly property studied in the class of circular-arc graphs. In particular, we introduce the *Helly Cliques* problem, in which for a given circular-arc graph G and its cliques C_1, \dots, C_k we need to decide whether G has a circular-arc model in which all the cliques C_1, \dots, C_k satisfy the Helly property. We show that the recognition of medusa graphs is polynomial time equivalent to the Helly Cliques problem. We refer to [3, 12] for two different proofs that the Helly Cliques problem is NP-complete. Also, to devise a polynomial algorithm recognizing L -graphs, we exploit an FPT algorithm for the Helly Cliques problem (for $k = 1$) devised in [12].

2 Preliminaries

We refer to the full version of this paper for the full version of this section.

2.1 Graphs and posets

All graphs considered in this paper are *simple*, that is, they have no multiedges and no loops. We denote a complete graph and a cycle on n vertices by K_n and C_n , respectively. A *hole* in a graph is an induced cycle on at least four vertices.

A *unicyclic* graph is a connected graph that has exactly one cycle. For a unicyclic graph M , we denote by M_O the set of vertices of the unique cycle of M .

We assume that the reader has some basic knowledge on partially ordered sets (*posets*).

2.2 M -graphs

Suppose M is a fixed unicyclic graph. Let (M^ϕ, ϕ) be an M -model of a graph G . If the subdivision M^ϕ of M is not relevant for our considerations, we denote the model (M^ϕ, ϕ) simply by (M, ϕ) or even by ϕ (if M is clear from the context). In this case we treat (M, ϕ) as the intersection model of G in which every set $\phi(v)$ forms an arcwise connected subset of some fixed plane drawing of the unicyclic graph M . Then M_O is the part of the drawing which contains the points of the drawing corresponding to the vertices and contained in the curves representing the edges from the unique cycle of M .

Let M be a unicyclic graph. We say that a graph G is a *saturated M -graph* if G has an M -model and has no M^* -model for any proper minor M^* of M .

► **Observation 2.1.** *Suppose M^* is a minor of M . If G has an M^* -model, then G has an M -model.*

2.3 Interval and Circular-Arc Graphs

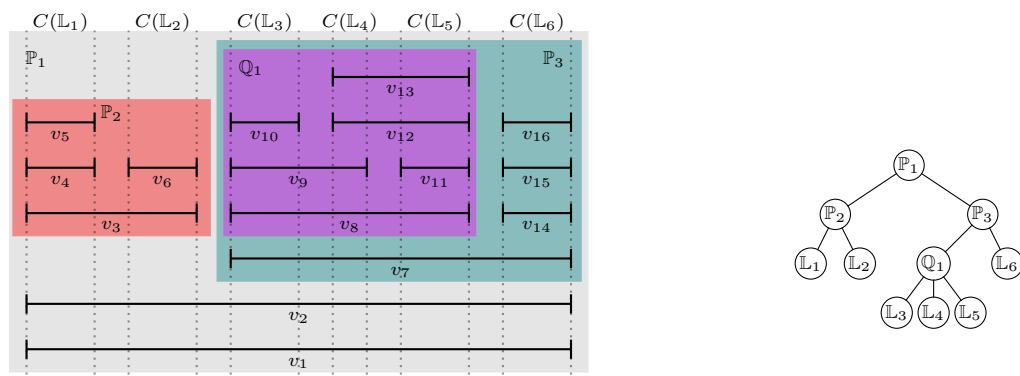
We assume that the reader has some basic knowledge of interval and circular-arc graphs. In our context we can define them as K_2 -graphs and K_3 -graphs.

Given an interval graph H and an interval model ϕ of H , for every $x \in \mathbb{R}$ we denote by $C(x)$ the set $\{v \in V(H) : x \in \phi(v)\}$. A *sector* S of ϕ is a maximal interval in \mathbb{R} such that $C(x) = C(y)$ holds for every $x, y \in S$. Given a sector S of ϕ , the *clique set* $C(S)$ of S is

equal to $C(x)$, where x is any point inside S . Clearly, every two sectors of ϕ are disjoint and the union of all sectors of ϕ covers \mathbb{R} (note that ϕ has at least two sectors S such that $C(S) = \emptyset$). We say that a sector S of ϕ is:

- *maximal* if $C(S)$ is a maximal clique in H ,
- *minimal* if $C(S) \subsetneq C(S')$ for any sector S' adjacent to S .

An interval model ϕ of H is *normalized* if ϕ has exactly $2c + 1$ sectors (we count also the sectors with the empty clique set), where c is the number of maximal cliques in H . We refer to the full version of this paper for more details on the relations between normalized interval models of H and *consecutive orderings* of maximal cliques of H represented by the *PQ-tree* of H (see Figure 2.1).



(a) A normalized interval model ϕ of H . Sectors of ϕ are separated by dashed lines.

(b) *PQ*-tree T of H with leaf order $L_1, L_2, L_3, L_4, L_5, L_6$ corresponding to consecutive clique ordering $C_1, C_2, C_3, C_4, C_5, C_6$.

■ **Figure 2.1**

Observe that we can obtain any interval model of H in the following way: first we choose a normalized model ϕ of H (which is equivalent to picking a consecutive ordering of maximal cliques of H) and then for each maximal sector S of ϕ we shift (by a little) the endpoints of the intervals of ϕ that lie on the borders of S (see Figure 2.2).

Since the definition of normalized models for circular-arc graphs is technical, we refer to the full version of this paper for the details. Here we only mention that in such models the relative relation between intersecting arcs (containment, covering the circle, or overlapping) depends on the relative relation between the closed neighbourhoods of the corresponding vertices of the graph.

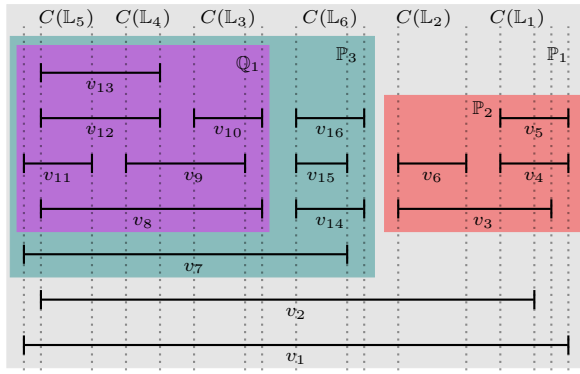
3 (Helly) Medusa Graphs

We refer to the full version of this paper for the full version of this section.

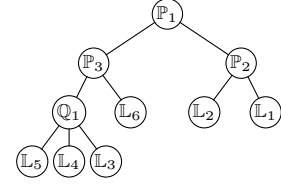
Recall that an \mathcal{M} -model of a graph G is an M -model of G where $M \in \mathcal{M}$. We introduce *normalized* \mathcal{M} -models, based on the following partition of $V(G)$ into the *circle part* V_C and the *tree part* V_T . Start with $V_C = \emptyset$ and repeatedly add to V_C :

- $V(C)$, if C is a hole in G ,
- $V(P)$, if P is an induced path in G joining two non-adjacent vertices from V_C .

Finally, let $V_T = V \setminus V_C$. Such a partition $V_C \cup V_T$ of $V(G)$ is unique and polynomial time computable. We call an \mathcal{M} -model (M, ϕ) of G *normalized* if:



(a) An interval model ϕ of H . Sectors of ϕ are separated by dashed lines.



(b) PQ-tree \mathbb{T} of H with leaf order $L_5, L_4, L_3, L_6, L_2, L_1$ corresponding to clique ordering $C_5, C_4, C_3, C_6, C_2, C_1$.

■ Figure 2.2

- $\phi(v) \cap M_O \neq \emptyset$ for every $v \in V_C$,
- $\phi(v) \cap M_O = \emptyset$ for every $v \in V_T$,
- $\{\phi(v) \cap M_O : v \in V_C\}$ is a normalized circular-arc model of $G[V_C]$.

► **Lemma 3.1.** *Every \mathcal{M} -graph G has a normalized \mathcal{M} -model ψ .*

Sketch of the proof: Let $V_C \cup V_T$ be a partition of $V(G)$ into the circle part V_C and the tree part V_T of G . Let (M, ψ) be an \mathcal{M} -model of G . The model (M, ψ) already satisfies $\psi(v) \cap M_O \neq \emptyset$ for every $v \in V_C$. Let T_1, \dots, T_k be a partition of V_T into connected components of $G[V_T]$. Let $N_C(T_i)$ be the neighbourhood of T_i in the cycle part of G , i.e. $N_C(T_i) = N(T_i) \cap V_C$. First we prove that for every $i \in [k]$ the graph $G[T_i \cup N_C(T_i)]$ is chordal and the set $N_C(T_i)$ forms a clique in $G[V_C]$. In particular, each $G[T_i \cup N_C(T_i)]$ has an F_i -model ψ_i for some tree F_i . Next, we note that $\psi|_{V_C}$ restricted to M_O forms a circular arc model of $G[V_C]$ in which each clique $N_C(T_i)$ is Helly. We normalize this model. Finally, we obtain a normalized \mathcal{M} -model (M', ϕ) of G by joining a point of F_i contained in $\bigcap \psi_i(N_C(T_i))$ to a point of M_O contained in $\bigcap \phi(N_C(T_i))$ and then by merging the models ψ and ψ_i for $i \in [k]$ appropriately. ◀

This yields our main theorem characterizing medusa graphs.

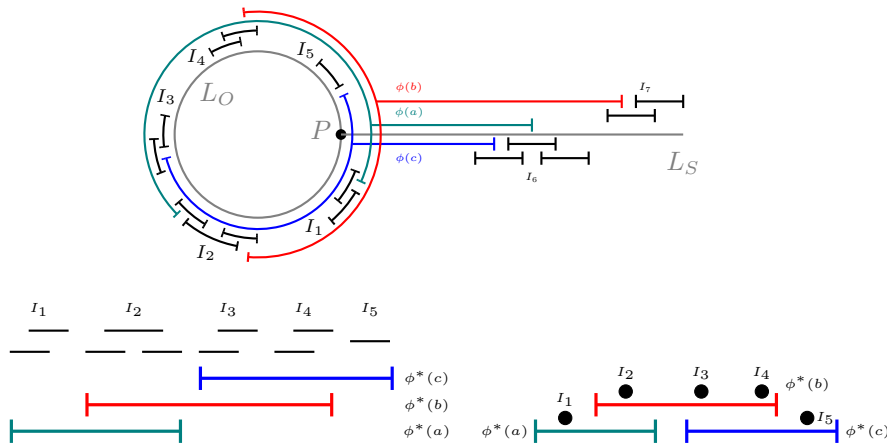
► **Theorem 3.2.** *(see the full version of this paper for the full proof) Let G be a graph, let $V(G) = V_C \cup V_T$ be a partition of $V(G)$ into the circle part V_C and the tree part V_T of G , and let T_1, \dots, T_k be a partition of V_T into connected components of $G[V_T]$. Then:*

1. G is a medusa graph if and only if $G[V_C]$ is a circular arc graph in which for every $i \in [k]$ the set $N_C(T_i)$ induces a clique in $G[V_C]$, and $G[V_C]$ admits a normalized circular-arc model in which every clique $N_C(T_i)$ is Helly.
2. G is a Helly medusa graph if and only if $G[V_C]$ is a Helly circular arc graph.

The conclusions of Theorem 3.2 bring our attention to the Helly Cliques problem.

► **Lemma 3.3.** *(see the full version of this paper for the full proof) Recognition of medusa graphs is poly-time equivalent to the Helly Cliques problem.*

We can summarize the section with the following theorem (which extends Theorem 1.2).



■ **Figure 4.1** Above: $\{a, b, c\}$ -centered L -model ϕ of G . Below left: Intervals $\phi^*(a), \phi^*(b), \phi^*(c)$: for $x \in \{a, b, c\}$, $\phi^*(x)$ is the shortest interval in $L_O \setminus P$ that contains $(L_O \setminus \phi(x))$ and every interval $\bigcup \phi(I)$ which has a non-empty intersection with $(L_O \setminus \phi(x))$. Below right: schematic view of ϕ^* with the components I_1, \dots, I_5 . Our second key step should output \mathcal{H} containing H such that $V(H) = \{a, b, c\}$ and $E(H) = \{ab, bc\}$.

► **Theorem 3.4.**

1. *The problem of recognizing medusa graphs is NP-complete.*
2. *The problem of recognizing medusa graphs parameterized by the number k of components in the tree part $G[V_T]$ of the input graph is FPT.*
3. *The problem of recognizing Helly medusa graphs is polynomial time solvable.*

Proof. The statements of the theorem follow from Lemma 3.3, from the fact that the Helly Cliques problem is NP-complete [3,12] and can be solved in time $2^{O(k \log k)} \text{poly}(n)$ [12], and from the fact that Helly circular-arc graphs recognition can be solved in linear-time [19]. ◀

4 Lollipop Graphs

In this section we derive a polynomial time algorithm for recognizing L -graphs, where L is the graph which consists of the clique K_3 and an edge attached to one vertex of K_3 (L is called a *lollipop* and L -graphs are called *lollipop graphs*). Since there are known polynomial-time algorithms recognizing $K_{1,3}$ -graphs [10] and K_3 -graphs [17, 20], we assume that an input graph $G = (V, E)$ is not an L^* -graph for any proper minor L^* of L . Hence our goal is to test whether G is a saturated L -graph.

We fix a plane drawing of L which consists of the circle L_O and the stick L_S attached to L_O in the point P (see Figure 4.1 for an illustration). Then we treat an L -model ϕ of G as the intersection model of G in which every set $\phi(v)$ forms an arcwise connected subset of the drawing of L . We call the arcs contained in $L_O \setminus P$ as *intervals* and we introduce *left-right* orders of the points in $L_O \setminus P$ (consistent with the clockwise order) and in $L_S \setminus P$.

Let C be a clique of G . An L -model ϕ of G is C -centered if $C = \{v \in V : P \in \phi(v)\}$ and G is C -centered if G admits a C -centered L -model. For example, the model ϕ shown in Figure 4.1 is $\{a, b, c\}$ -centered.

Our approach consists of three key steps. The first step is summarized as follows.

► **Theorem 4.1.** *There is a polynomial time algorithm that, given a graph G , either decides that G is a saturated L -graph, or outputs a set of cliques \mathcal{C} of G such that, if G is a saturated L -graph then G is a C -centered L -graph for some $C \in \mathcal{C}$.*

Given the above theorem it remains to efficiently decide whether G is a C -centered graph for some fixed clique C of G . Let \mathcal{I} denote the set of components of $G[V \setminus C]$. Note that for every $I \in \mathcal{I}$ the set I induces an interval graph in G if G is C -centered.

Suppose ϕ is a C -centered model of G . To describe our second step, for every $c \in C$ let $\phi^*(c)$ denote the shortest interval in $L_O \setminus P$ containing the interval $L_O \setminus \phi(c)$ and every interval $\bigcup \phi(I)$ whenever $L_O \setminus \phi(c)$ intersects $\bigcup \phi(I)$, for $I \in \mathcal{I}$. Let H_ϕ be an interval graph with the vertex set $C' = \{c \in C : \phi^*(c) \neq \emptyset\}$ and with the intersection model given by the intervals $\{\phi^*(c) : c \in C'\}$.

Let $C' \subseteq C$ and let H be an interval graph on the vertex set C' . A C -centered L -model ϕ of G is (C, H) -centered if $H_\phi = H$ and G is (C, H) -centered if G admits a (C, H) -centered model. For example, the model ϕ from Figure 4.1 is $(\{a, b, c\}, H)$ -centered, where H is such that $V(H) = \{a, b, c\}$ and $E(H) = \{ab, bc\}$. Our second step comes down to the following.

► **Theorem 4.2.** *There is a polynomial time algorithm that, given a graph G and a clique $C \subseteq V(G)$, outputs a set of interval graphs \mathcal{H} such that, if G is a C -centered L -graph, then G is a (C, H) -centered L -graph for some $H \in \mathcal{H}$.*

Our final step, which is the most technical and probably most difficult, can be summarized as follows:

► **Theorem 4.3.** *There is a polynomial time algorithm that, given a graph G , a clique $C \subseteq V(G)$, and an interval graph H on a subset C' of C , decides whether G is a (C, H) -centered graph.*

The algorithm from Theorem 4.3 exploits dynamic programming along the PQ-tree of H to test whether there is a partition $(\mathcal{J}, \mathcal{J}')$ of the components of \mathcal{I} together with a total ordering \prec of \mathcal{J} such that there is a (C, H) -centered model ϕ of G that places the interval graphs of \mathcal{J} on $L_O \setminus P$ in the order \prec and the interval graphs of \mathcal{J}' on the stick L_S .

4.1 Sketch of the proof of Theorem 4.1 (the first key step)

We refer to the full version of this paper for the full proof.

Let G be a graph which is neither $K_{1,3}$ -graph nor K_3 -graph. Let $V_C \cup V_T$ be the partition of $V(G)$ into the circle part V_C and the tree part V_T of G , and let \mathcal{T} be the set of all maximal cliques of the chordal graphs $G[T \cup N_C(T)]$, where T runs over the components of $G[V_T]$ (see Section 3). The algorithm for Theorem 4.1 works as follows. For every $C^* \in \mathcal{T}$:

- output C^* and denote by \mathcal{I}^* the components of $G[V \setminus C^*]$,
- for every component $I \in \mathcal{I}^*$ let $C' = \{v \in C^* : v \text{ has a neighbour in } I\}$,
 - accept G as saturated L -graph if $G[I \cup C']$ admits a circular arc model with C' Helly (we use a poly-time algorithm for the Helly Cliques problem with $k = 1$) and $G[V \setminus I]$ admits an interval model with C^* as the leftmost maximal clique,
 - if $G[I]$ is an interval graph, then for every maximal clique D of $G[I]$ output the clique $C(D) = C' \cup \{v \in D : C' \subseteq N(v)\}$.

Now we give a sketch of the proof that this algorithm is correct. Assuming G is a saturated L -graph, we first argue G admits an L -model ϕ such that:

- there is a clique C^* from \mathcal{T} such that the set $\bigcap \phi(C^*)$ is contained in $L_S \setminus P$ and $\bigcap \phi(C^*)$ is as close to P as possible,

- for every component $I \in \mathcal{I}^*$ we have either $\bigcup \phi(I) \subseteq L_O$ or $\bigcup \phi(I) \subseteq (L_S \setminus P)$.

As we show in the full version of this paper, such properties are satisfied by so-called *saturated* L -models of G . Given such ϕ , we fall into one of the following cases:

- P is not covered by $\bigcup \phi(I)$ for any $I \in \mathcal{I}^*$. Then ϕ can be easily turned into C^* -centered L -model of G (since the algorithm adds C^* to \mathcal{C} , Theorem 4.1 is satisfied).
- P is covered by $\bigcup \phi(I)$ for some $I \in \mathcal{I}^*$. The algorithm accepts G if I is the only component on L_O (which possibly induces a circular-arc graph and covers the whole circle L_O). Otherwise, if two components from \mathcal{I}^* are represented on L_O , then all the components in \mathcal{I}^* induce interval graphs in G . In this case we argue that ϕ can be turned into a $C(D)$ -centered model, where D is a maximal clique in $G[I]$ with $\bigcap \phi(D)$ next to P (since the algorithm adds $C(D)$ to \mathcal{C} , Theorem 4.1 is satisfied).

Given the previous, the algorithm for Theorem 4.1 either accepts G , or outputs \mathcal{C} of size at most $O(n^2)$ (\mathcal{T} has size $O(n)$ and for every $C^* \in \mathcal{T}$ the total number of maximal cliques in the interval components of $G[V \setminus C^*]$ is $O(n)$).

4.2 Sketch of the proof of Theorem 4.2 (the second key step)

We refer to the full version of this paper for the full proof.

Consider a graph G and a clique C of G . Let \mathcal{I} be the set of components of $G[V \setminus C]$. For now, consider a C -centered model ϕ . Then every $I \in \mathcal{I}$ induces an interval graph and $\{\phi(v) \mid v \in I\}$ is an interval model for $G[I]$. We partition $I \in \mathcal{I}$ according to the intersection with L_O and L_S in the model ϕ :

$$\mathcal{I}_O^\phi = \left\{ I \in \mathcal{I} : \bigcup \phi(I) \subseteq L_O \right\} \quad \text{and} \quad \mathcal{I}_S^\phi = \left\{ I \in \mathcal{I} : \bigcup \phi(I) \subseteq L_S \right\}.$$

For $I \in \mathcal{I}$, let $C(I) = \{c \in C : c \text{ is not adjacent to some vertex in } I\}$. Let ϕ be a C -centered model of G . Note that the graph H_ϕ , defined in Section 4, can be equivalently defined such that

$$C' = V(H_\phi) = \bigcup_{I \in \mathcal{I}_O^\phi} C(I) \quad \text{and} \quad E(H_\phi) = \{\{c, c'\} : \text{there is } I \in \mathcal{I}_O^\phi \text{ such that } c, c' \in C(I)\}.$$

Similarly, the model ϕ^* of H_ϕ can be equivalently defined such that for every $c \in C'$ the set $\phi^*(c)$ is the shortest interval that contains $\bigcup \phi(I)$ for every $I \in \mathcal{I}_O^\phi$ such that $c \in C(I)$ – see Figure 4.1 for an illustration. Note that every non-minimal sector of ϕ^* contains at least one component from \mathcal{I}_O^ϕ and every $I \in \mathcal{I}_O^\phi$ occupies a sector of ϕ^* with the clique set $C(I)$.

We distinguish the interval graphs $I \in \mathcal{I}$ as follows:

- $I \in \mathcal{I}$ is called *ambiguous* if $G[C \cup I]$ has an interval model with C as its left-most clique (hence, I might be placed on the stick); otherwise I is called *circle*.
- An ambiguous component $I \in \mathcal{I}$ is *simple* if $N(u) \cap C = N(v) \cap C$ for all $u, v \in I$.

We denote the sets of the circle, ambiguous, ambiguous simple, and ambiguous non-simple components by \mathcal{I}_c , \mathcal{I}_a , \mathcal{I}_{as} , and \mathcal{I}_{ans} , respectively. Clearly, $\mathcal{I}_c \subseteq \mathcal{I}_O^\phi$, for every C -centered model ϕ . Moreover, we show that for every $I \in \mathcal{I}_c$ and every C -centered model ϕ of G the component I occupies a maximal sector of ϕ^* and no other component occupies this sector – see the circle component I_2 in Figure 4.1. We additionally assume that $C(I) \neq C(J)$ for every two simple components $I \neq J$ since otherwise we may simply consider the input without J .

8:10 Recognizing H -Graphs – Beyond Circular-Arc Graphs

We order \mathcal{I} based on their subset relation on $\{C(I) : I \in \mathcal{I}\}$ and their classification into \mathcal{I}_a , \mathcal{I}_{as} , and \mathcal{I}_{ans} . We define \subset_t for $I, J \in \mathcal{I}$ as

$$I \subset_t J \text{ if } \begin{aligned} & (C(I) \subsetneq C(J)) \text{ or} \\ & (C(I) = C(J) \text{ and } I \in \mathcal{I}_c \text{ and } J \in \mathcal{I}_a) \text{ or} \\ & (C(I) = C(J) \text{ and } I \in \mathcal{I}_{ans} \text{ and } J \in \mathcal{I}_{as}). \end{aligned}$$

Then $(\mathcal{I}, \subseteq_t)$ is a poset, where \subseteq_t is the reflexive closure of \subset_t .

▷ **Claim 4.4.** For every C -centered model ϕ we have $\mathcal{I}_S^\phi \subseteq \mathcal{I}_a$, the components from \mathcal{I}_S^ϕ form a chain in $(\mathcal{I}, \subseteq_t)$, and the left-right order of the components on L_S coincides with $(\mathcal{I}_S^\phi, \subseteq_t)$.

Let $\mathcal{D} \subseteq \mathcal{I}$ be defined as follows. For every $J \in \mathcal{I}_a$,

(D1) if $C(I) \subseteq C(J)$ for some $I \in \mathcal{I}_c$, add J to \mathcal{D} .

(D2) if $J \in \mathcal{I}$ is such that the downset of J has the width at least 4 in $(\mathcal{I}, \subseteq_t)$, add J to \mathcal{D} . We show that the components from \mathcal{D} need to be represented on the stick in any C -centered model of G .

Since every maximal sector of ϕ^* contains a component from \mathcal{A}^ϕ and the components from \mathcal{A}^ϕ occupy maximal sectors of ϕ^* , the set \mathcal{A}^ϕ determines the set of all maximal cliques of H_ϕ . Hence, the interval graph H_ϕ is uniquely determined by the antichain \mathcal{A}^ϕ containing the maximal components from $(\mathcal{I}_O^\phi, \subseteq_t)$. Therefore, we refer to H_ϕ as to $H(\mathcal{A}^\phi)$.

Let $\mathcal{I}' = \mathcal{I} \setminus \mathcal{D}$. Let \mathcal{A} be the set of the maximal components in $(\mathcal{I}', \subseteq_t)$. Let ϕ be a C -centered L -model of G . Since \mathcal{A} forms an antichain and \mathcal{I}_S^ϕ forms a chain, their intersection contains at most one element. If $\mathcal{A} \cap \mathcal{I}_S^\phi = \emptyset$, then $\mathcal{A}^\phi = \mathcal{A}$ and we output $H(\mathcal{A})$ as a candidate. If $|\mathcal{A} \cap \mathcal{I}_S^\phi| = \{A\}$, then $\mathcal{A}^\phi = (\mathcal{A} \setminus \{A\}) \cup \mathcal{A}_3$, where \mathcal{A}_3 is an antichain in the downset of A . Hence we try all of the $O(n)$ maximal components in $A \in \mathcal{A}$ combined with all antichains in $DS(A)$. Since $A \notin \mathcal{D}$, its downset has the width at most 3 and there are $O(n^3)$ antichains in $DS(A)$ which can be enumerated in polynomial time.

4.3 Sketch of the proof of Theorem 4.3 (the third key step)

We refer to the full version of this paper for the full proof.

First, we reduce the input instance G, C, H , so as there is no component $I \in \mathcal{I}$ such that $C(I)$ is the clique set of a minimal sector of an interval model of H (the set of cliques of minimal sectors of an interval model of H is independent on the model of H).

Let \mathbb{T} be the PQ-tree of H , let $V(\mathbb{T})$ be the nodes of \mathbb{T} , and let \mathbb{R} be the root of \mathbb{T} . For $\mathbb{N} \in V(\mathbb{T})$, let $\mathcal{L}(\mathbb{N})$ denote the set of all leaves of \mathbb{T} which descend \mathbb{N} in \mathbb{T} . We set $\mathcal{L} = \mathcal{L}(\mathbb{R})$. For $\mathbb{L} \in \mathcal{L}$ let $C(\mathbb{L})$ denote the clique of H represented by \mathbb{L} . We refer to the full version of this paper for more about PQ-trees.

Our main task is to find a partition $(\mathcal{J}, \mathcal{J}')$ of $\mathcal{I}' = \mathcal{I} \setminus \mathcal{D}$ together with a total ordering \prec of \mathcal{J} such that there is a model ϕ that places the interval graphs of \mathcal{J} on the circle L_O in the order \prec and the interval graphs of \mathcal{J}' on the stick L_S .

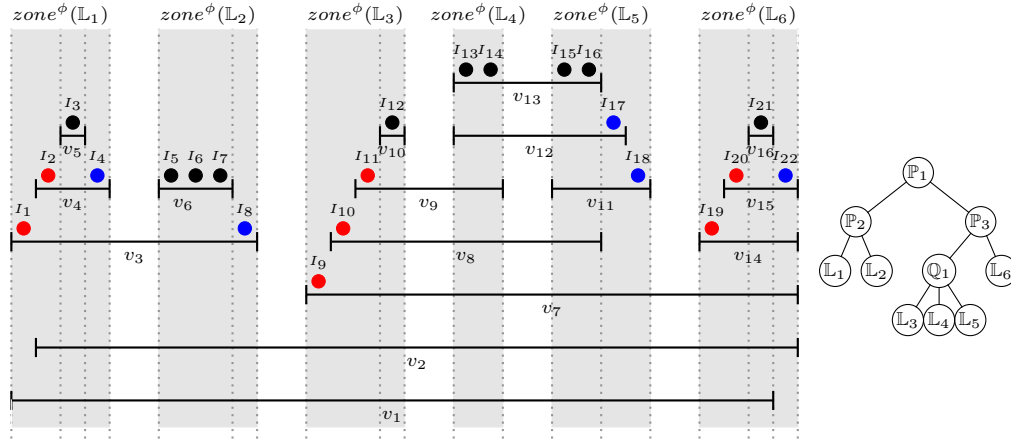
The easy part is to check whether $\mathcal{J}' \subseteq \mathcal{I}'$ can be placed on the stick. We need to check whether $(\mathcal{J}' \cup \mathcal{D})$ forms a chain in the poset $(\mathcal{I}_a, \sqsubseteq)$, with \sqsubseteq being the reflexive closure of the following binary relation \sqsubset , defined for distinct $I, J \in \mathcal{I}_a$:

$$I \sqsubset J \text{ if } C(I) \subseteq C^s(J), \text{ where} \\ C^s(J) = \{c \in C : c \text{ is not adjacent to every vertex in } J\}.$$

Now, let us consider how to test whether $\mathcal{J} \subseteq \mathcal{I}'$ together with an ordering \prec allows a model that places the interval graphs of \mathcal{J} in the order of \prec on the circle. We will denote such an ordering (\mathcal{J}, \prec) as a *good order* for \mathbb{R} . Our final dynamic program then determines the

good orderings in a bottom up fashion. Hence we will define a *good order* (\mathcal{K}, \prec) with respect to some node \mathbb{N} of \mathbb{T} and with respect to some left and right *borders*, (B_L, B_R) , which are cliques in H . Since this definition is technical (see the full version of this paper), we describe some properties of good orderings that occur in $(\mathcal{I}_O^\phi, \prec_\phi)$, where ϕ is a (C, H) -centered model of G and $(\mathcal{I}_O^\phi, \prec_\phi)$ is the left-right order of the components from \mathcal{I}_O^ϕ on $L_O \setminus P$.

Let ϕ be a (C, H) -centered model of G . For $\mathbb{L} \in \mathcal{L}$ let $\text{inner}(\mathbb{L}) = \{I \in \mathcal{I}' : C(I) = C(\mathbb{L})\}$ and let $\text{inner}_O^\phi(\mathbb{L}) = \text{inner}(\mathbb{L}) \cap \mathcal{I}_O^\phi$, and $\text{inner}_S^\phi(\mathbb{L}) = \text{inner}(\mathbb{L}) \cap \mathcal{I}_S^\phi$. Additionally, assume the maximal cliques of H appear in ϕ^* in the order $C(\mathbb{L}_1), \dots, C(\mathbb{L}_n)$. For every $\mathbb{L}_i \in \mathcal{L}$ we define the *left zone* $\text{zone}_L^\phi(\mathbb{L}_i)$ of \mathbb{L}_i as an interval of $(\mathcal{I}_O^\phi, \prec_\phi)$ including the components from the sectors of ϕ^* contained strictly between the maximal sector $S(\mathbb{L}_i)$ with the clique set $C(\mathbb{L}_i)$ and the minimal sector of ϕ^* preceding $S(\mathbb{L}_i)$ in ϕ^* . We define the *right zone* $\text{zone}_R^\phi(\mathbb{L}_i)$ of \mathbb{L}_i analogously and we set $\text{zone}^\phi(\mathbb{L}_i) = \text{zone}_L^\phi(\mathbb{L}_i) \cup \text{inner}_O^\phi(\mathbb{L}_i) \cup \text{zone}_R^\phi(\mathbb{L}_i)$. For a non-leaf node \mathbb{N} , let $\text{zone}^\phi(\mathbb{N}) = \bigcup_{\mathbb{L} \in \mathcal{L}(\mathbb{N})} \text{zone}^\phi(\mathbb{L})$. Figure 4.2 shows the zones for some example model ϕ^* . Components from the sets $\text{zone}_L^\phi(\mathbb{L}_i)$, $\text{inner}_O^\phi(\mathbb{L}_i)$, and $\text{zone}_R^\phi(\mathbb{L}_i)$ are illustrated as red, black, and blue dots, respectively. We have, for example, $\text{zone}_L^\phi(\mathbb{L}_3) = \{I_9, I_{10}, I_{11}\}$, $\text{zone}_R^\phi(\mathbb{L}_2) = \{I_8\}$, and $\text{inner}_O^\phi(\mathbb{L}_2) = \{I_5, I_6, I_7\}$. We have $\text{zone}^\phi(\mathbb{Q}_1) = \{I_9, \dots, I_{18}\}$ as $\mathcal{L}(\mathbb{Q}_1) = \{\mathbb{L}_3, \mathbb{L}_4, \mathbb{L}_5\}$.



■ **Figure 4.2** Zones in ϕ^* .

- Among others, we show that for every $i \in [n]$ (below we assume $C(\mathbb{L}_0) = C(\mathbb{L}_{n+1}) = \emptyset$):
 - the set $\text{zone}_L^\phi(\mathbb{L}_i)$ forms a chain in $(\mathcal{I}_a, \sqsubseteq)$, $(\text{zone}_L^\phi(\mathbb{L}_i), \sqsubseteq)$ is equal to $(\text{zone}_L^\phi(\mathbb{L}_i), \prec_\phi)$ and $\text{zone}_L^\phi(\mathbb{L}_i)$ respects the border $C(\mathbb{L}_{i-1}) \cap C(\mathbb{L}_i)$, which means $C(\mathbb{L}_{i-1}) \cap C(\mathbb{L}_i) \sqsubseteq \text{zone}_L^\phi(\mathbb{L}_i)$,
 - the set $\text{zone}_R^\phi(\mathbb{L}_i)$ forms a chain in $(\mathcal{I}_a, \sqsubseteq)$, we have $(\text{zone}_R^\phi(\mathbb{L}_i), \sqsubseteq)$ is equal to the reverse of $(\text{zone}_R^\phi(\mathbb{L}_i), \prec_\phi)$ and $\text{zone}_R^\phi(\mathbb{L}_i)$ respects the border $C(\mathbb{L}_i) \cap C(\mathbb{L}_{i+1})$, which means $C(\mathbb{L}_i) \cap C(\mathbb{L}_{i+1}) \sqsubseteq \text{zone}_R^\phi(\mathbb{L}_i)$.

The next step is to define when (\mathcal{K}, \prec) is a *good order* for \mathbb{N} and borders (B_L, B_R) . Roughly speaking, (\mathcal{K}, \prec) is defined such that it allows to derive an admissible order $\mathbb{L}_1, \dots, \mathbb{L}_m$ of the leaves from $\mathcal{L}(\mathbb{N})$ and to define the zones for \mathbb{L}_i in (\mathcal{K}, \prec) for all $i \in [m]$. Moreover, it is required that the left zone of \mathbb{L}_1 respects the border B_L , the right zone of \mathbb{L}_m respects the border B_R , and the right zone of \mathbb{L}_i and the left zone of \mathbb{L}_{i+1} respect the border $C(\mathbb{L}_i) \cap C(\mathbb{L}_{i+1})$ for $i \in [m-1]$. For example, in Figure 4.2 $(\text{zone}^\phi(\mathbb{L}_3), \prec_\phi)$ is a good order for \mathbb{L}_3 and the borders $(\{v_1, v_2\}, \{v_1, v_2, v_7, v_8, v_9\})$, $(\text{zone}^\phi(\mathbb{Q}_1), \prec_\phi) = (I_9, \dots, I_{18})$ is a good order for \mathbb{Q}_1 and for the borders $(\{v_1, v_2\}, \{v_1, v_2, v_7\})$. Finally, the full definition of good orders allows us to prove Theorem 4.6, which boils down the problem of searching for a (C, H) -centered model of G to the problem of testing whether there is a “good triple” for the set \mathcal{I}' and the node \mathbb{R} .

► **Definition 4.5.** Let \mathbb{N} be a node of \mathbb{T} and $\mathcal{I}^* \subseteq \mathcal{I}'$. A triple $(\mathcal{J}, \mathcal{J}', \prec)$ is good for the set \mathcal{I}^* , the node \mathbb{N} , and the borders (B_L, B_R) , if:

- $\{\mathcal{J}, \mathcal{J}'\}$ is a partition of \mathcal{I}^* .
- (\mathcal{J}, \prec) is a good order for the node \mathbb{N} and the borders (B_L, B_R) ,
- $\mathcal{J}' \cup \mathcal{D}$ is a chain in $(\mathcal{I}_a, \sqsubseteq)$.

If $(B_L, B_R) = (\emptyset, \emptyset)$, we simply say $(\mathcal{J}, \mathcal{J}', \prec)$ is good for the set \mathcal{I}^* and the node \mathbb{N} .

With the notion of a good triple we obtain the following theorem that characterizes all (C, H) -centered models of G .

► **Theorem 4.6** (see the full version of this paper for the full proof). Let G be a graph.

1. For every (C, H) -centered model ϕ of G the triple $(\mathcal{I}_O^\phi, \mathcal{I}_S^\phi \setminus \mathcal{D}, \prec_\phi)$ is good for \mathcal{I}' and the node \mathbb{R} .
2. For every triple $(\mathcal{J}, \mathcal{J}', \prec)$ that is good for \mathcal{I}' and the node \mathbb{R} , there is a (C, H) -centered model ϕ of G such that $(\mathcal{J}, \prec) = (\mathcal{I}_O^\phi, \prec_\phi)$ and $\mathcal{J}' \cup \mathcal{D} = \mathcal{I}_S^\phi$.

Our algorithm needs to test whether there exists a good triple for the set $\mathcal{I}' = \text{comp}(\mathbb{R})$. Roughly speaking, this technical part is done as follows: first, we carefully define the sets $\text{comp}(\mathbb{N}) \subseteq \mathcal{I}'$ for every node $\mathbb{N} \in V(\mathbb{T})$ and then we compute good triples¹ for every set $\text{comp}(\mathbb{N})$ using dynamic programming over \mathbb{T} .

5 Butterfly-Graphs

Here we sketch an approach to proving Theorem 1.1. The main task is to show NP-hardness of recognizing butterfly-graphs, where a butterfly is the graph consisting of two K_3 's joined on one vertex.

► **Theorem 5.1.** *BUTTERFLY-GRAPH RECOGNITION is NP-complete.*

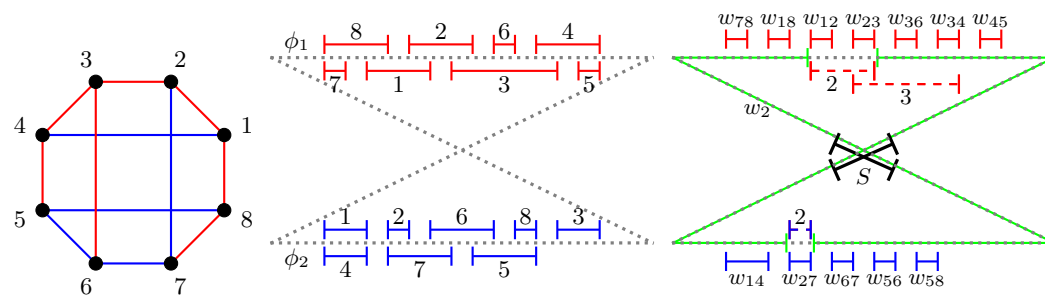
It is easy to see NP-membership [10]. To show NP-hardness, we reduce from the BIPARTITE 2-TRACK; that is to decide whether a given bipartite graph G is 2-track. A graph G is 2-track if there are sets E_1, E_2 whose union is $E(G)$ such that $(V(G), E_1)$ and $(V(G), E_2)$ are interval graphs. Gonçalves & Ochem proved NP-hardness of this problem [15].

Construction: For a given bipartite graph G we construct a graph G' that is a butterfly-graph if and only if G is 2-track. Let S be a star $K_{1,4}$ where every edge is subdivided once. The vertex set $V(G')$ consists of $V(S)$, a vertex w_v for every vertex $v \in V(G)$, an edge-vertex w_{uv} for every edge $uv \in E(G)$ and $V(S)$. The edge set $E(G')$ consists of $\binom{V(G)}{2}$, $E(S)$ and the edges of making w_{uv} adjacent to $w \in V(G) \setminus \{u, v\}$ for every edge $uv \in E(G)$. Finally, we add every edge between $V(G)$ and $V(S)$.

Given G the graph G' can be constructed in polynomial time. It remains to show that G is 2-track if and only if G' is a butterfly-graph. For the formal proof we refer to the full version of this paper. Here we only sketch how we construct a butterfly-model of G' provided G is a bipartite 2-track (witnessed by interval models ϕ_1 and ϕ_2 of $(V(G), E_1)$ and $(V(G), E_2)$). A butterfly-model of G' is obtained as follows (see Figure 5.1 for an illustration):

- we embed ϕ_1 and ϕ_2 into a subdivision H^ϕ of the butterfly as shown in Figure 5.1,
- we represent the vertices of S in the center of H^ϕ ,
- for $v \in V(G)$ we represent w_v by the set $V(H^\phi) \setminus (\phi_1(v) \cup \phi_2(v))$ (see green set w_2)
- for $uv \in E(G)$, we represent w_{uv} by the set $\phi_i(u) \cap \phi_i(v)$ if $uv \in E_i$ (see red set w_{23}).

¹ Since these sets might have exponential size, the algorithm calculates only their “fingerprints”.



■ **Figure 5.1** A bipartite graph G (to the left) as a 2-track (red and blue edges) witnessed by interval models ϕ_1 and ϕ_2 (in the middle), and a *butterfly*-model of G' (to the right).

6 Conclusions

The question whether for a fixed graph H the class of H -graphs can be recognized in polynomial time was posed by Biro, Hujter, and Tuza over 30 years ago [5]. The main results of our work show that the boundary between polynomial and NP-hard cases of H -graphs recognition lies somewhere between unicyclic graphs H , strictly above the class of circular-arc graphs. The research carried out so far reveals connections between the H -graphs recognition problems for unicyclic graphs H and certain problems related to the Helly property of circular-arc graphs. The latter problems are now intensively studied [12] and the positive results achieved so far allow us to state the following conjecture.

► **Conjecture 6.1.** *The recognition of H -graphs is polynomial-time solvable if and only if H is a unicyclic graph or H is a tree.*

In particular, we believe that the techniques introduced in our work, suitably extended, can be used to devise polynomial algorithms for the cases where H consists of a cycle and some edges attached to it. The situation might be different when we allow to have trees attached to the cycle in H . The difficulty might be caused by the lack of a data structure maintaining all representations of a T -graph, where T is a tree different than a path (counterparts of PQ-trees for interval graphs).

References

- 1 Deniz Ağaoğlu Çağırıcı and Petr Hliněný. Efficient isomorphism for S_d -graphs and T -graphs. *Algorithmica*, 2022. doi:10.1007/s00453-022-01033-8.
- 2 Deniz Ağaoğlu Çağırıcı and Petr Hliněný. Isomorphism testing for T -graphs in FPT. *WALCOM: Algorithms and Computation*, pages 239–250, 2022.
- 3 Deniz Ağaoğlu Çağırıcı and Peter Zeman. Recognition and isomorphism of proper U -graphs in FPT-time. *CoRR*, abs/2206.13372, 2022. doi:10.48550/arXiv.2206.13372.
- 4 Vikraman Arvind, Roman Nedela, Iliia Ponomarenko, and Peter Zeman. Testing isomorphism of chordal graphs of bounded leafage is fixed-parameter tractable (extended abstract). In Michael A. Bekos and Michael Kaufmann, editors, *Graph-Theoretic Concepts in Computer Science - 48th International Workshop, WG 2022, Tübingen, Germany, June 22-24, 2022, Revised Selected Papers*, volume 13453 of *Lecture Notes in Computer Science*, pages 29–42. Springer, 2022. doi:10.1007/978-3-031-15914-5_3.
- 5 Miklós Biró, Mihály Hujter, and Zsolt Tuza. Precoloring extension. i. interval graphs. *Discret. Math.*, 100(1-3):267–279, 1992. doi:10.1016/0012-365X(92)90646-W.

- 6 K. Booth and G. Lueker. Testing for the consecutive ones property, interval graphs, and graph planarity using PQ -tree algorithms. *J. Comput. System Sci.*, 13(3):335–379, 1976. doi:10.1016/S0022-0000(76)80045-1.
- 7 K. S. Booth. Pq -tree algorithms, November 1975. URL: <https://www.osti.gov/biblio/7189564>.
- 8 Steven Chaplick, Fedor V. Fomin, Petr A. Golovach, Dusan Knop, and Peter Zeman. Kernelization of graph hamiltonicity: Proper H -graphs. *SIAM J. Discret. Math.*, 35(2):840–892, 2021. doi:10.1137/19M1299001.
- 9 Steven Chaplick, Petr A. Golovach, Tim A. Hartmann, and Dusan Knop. Recognizing proper tree-graphs. In Yixin Cao and Marcin Pilipczuk, editors, *15th International Symposium on Parameterized and Exact Computation, IPEC 2020, December 14-18, 2020, Hong Kong, China (Virtual Conference)*, volume 180 of *LIPICs*, pages 8:1–8:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICs.IPEC.2020.8.
- 10 Steven Chaplick, Martin Töpfer, Jan Voborník, and Peter Zeman. On H -topological intersection graphs. *Algorithmica*, 83(11):3281–3318, 2021. doi:10.1007/s00453-021-00846-3.
- 11 Steven Chaplick and Peter Zeman. Combinatorial problems on h -graphs. *Electron. Notes Discret. Math.*, 61:223–229, 2017. doi:10.1016/j.endm.2017.06.042.
- 12 Jan Derbisz and Tomasz Krawczyk. Circular-arc graphs and the Helly property.
- 13 Fedor V. Fomin, Petr A. Golovach, and Jean-Florent Raymond. On the tractability of optimization problems on H -graphs. In Yossi Azar, Hannah Bast, and Grzegorz Herman, editors, *26th Annual European Symposium on Algorithms, ESA 2018, August 20-22, 2018, Helsinki, Finland*, volume 112 of *LIPICs*, pages 30:1–30:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPICs.ESA.2018.30.
- 14 Fănică Gavril. The intersection graphs of subtrees in trees are exactly the chordal graphs. *Journal of Combinatorial Theory, Series B*, 16(1):47–56, 1974. doi:10.1016/0095-8956(74)90094-X.
- 15 Daniel Gonçalves and Pascal Ochem. On star and caterpillar arboricity. *Discrete Mathematics*, 309(11):3694–3702, 2009. doi:10.1016/j.disc.2008.01.041.
- 16 Lars Jaffke, O-joung Kwon, and Jan Arne Telle. Mim-width II. the feedback vertex set problem. *Algorithmica*, 82(1):118–145, 2020. doi:10.1007/s00453-019-00607-3.
- 17 Haim Kaplan and Yahav Nussbaum. A simpler linear-time recognition of circular-arc graphs. *Algorithmica*, 61(3):694–737, 2011. doi:10.1007/s00453-010-9432-y.
- 18 Pavel Klavík, Jan Kratochvíl, Yota Otachi, and Toshiki Saitoh. Extending partial representations of subclasses of chordal graphs. *Theor. Comput. Sci.*, 576:85–101, 2015. doi:10.1016/j.tcs.2015.02.007.
- 19 Min Chih Lin and Jayme L. Szwarcfiter. Characterizations and linear time recognition of Helly circular-arc graphs. In *Computing and combinatorics*, volume 4112 of *Lecture Notes in Comput. Sci.*, pages 73–82. Springer, Berlin, 2006. doi:10.1007/11809678_10.
- 20 Ross M. McConnell. Linear-time recognition of circular-arc graphs. *Algorithmica*, 37(2):93–147, 2003. doi:10.1007/s00453-003-1032-7.
- 21 Donald J. Rose, Robert Endre Tarjan, and George S. Lueker. Algorithmic aspects of vertex elimination on graphs. *SIAM J. Comput.*, 5(2):266–283, 1976. doi:10.1137/0205021.
- 22 Alan Tucker. An efficient test for circular-arc graphs. *SIAM J. Comput.*, 9(1):1–24, 1980. doi:10.1137/0209001.