Stability, Vertex Stability, and Unfrozenness for Special Graph Classes

Frank Gurski[1] · Jörg Rothe[1] · Robin Weishaupt[1]

Abstract

Frei et al. (J. Comput. Syst. Sci. **123**, 103–121, 2022) show that the stability, vertex stability, and unfrozenness problems with respect to certain graph parameters are complete for $\mathbf{\Theta_2^P}$, the class of problems solvable in polynomial time by parallel access to an NP oracle. They studied the common graph parameters $\boldsymbol{\alpha}$ (the independence number), $\boldsymbol{\beta}$ (the vertex cover number), $\boldsymbol{\omega}$ (the clique number), and $\boldsymbol{\chi}$ (the chromatic number). We complement their approach by providing polynomial-time algorithms solving these problems for special graph classes, namely for graphs with bounded tree-width or bounded clique-width. In order to improve these general time bounds even further, we then focus on trees, forests, bipartite graphs, and co-graphs.

Keywords Graph theory · Stability · Vertex stability · Unfrozenness · Robustness · Tree-width · Clique-width · Colorability · Clique · Independent set · Vertex cover

1 Introduction

Frei et al. [1] comprehensively studied the problem of how stable certain central graph parameters are when a given graph is slightly modified, i.e., under operations such as adding or deleting either edges or vertices. Specifically, they consider the four graph parameters independence number α, vertex cover number β, clique number ω, and chromatic number χ (to be formally defined in Section 2). Given such a graph parameter ξ, they introduce the problems ξ-STABILITY, ξ-VERTEXSTABILITY, and

✉ Robin Weishaupt
robin.weishaupt@hhu.de

Frank Gurski
frank.gurski@hhu.de

Jörg Rothe
rothe@hhu.de

[1] Institut für Informatik, Heinrich-Heine-Universität Düsseldorf, Universitätsstraße 1, 40225 Düsseldorf, Germany

ξ-UNFROZENNESS and show that they are, typically, Θ_2^P-complete,[1] that is, they are complete for the complexity class known as "parallel access to NP," which was introduced by Papadimitriou and Zachos [2] and intensively studied by, e.g., Wagner [3, 4], Gottlob [5], Eiter and Lukasiewicz [6], Lukasiewicz and Malizia [7], Hemaspaandra et al. [8–12], and Rothe et al. [13].[2] In their survey, Hemaspaandra et al. [15] discuss the question of whether Θ_2^P-completeness provides any more insight into the inherent hardness of a problem than NP-hardness, and argue that the answer crucially depends on the computational model used.

Furthermore, Frei et al. [1] prove that some more specific versions of these problems, namely k-χ-STABILITY and k-χ-VERTEXSTABILITY, are NP-complete for $k = 3$ and DP-complete for $k \geq 4$, respectively, where DP was introduced by Papadimitriou and Yannakakis [16] as the class of problems that can be written as the difference of NP problems, a. k. a. the second level of the boolean hierarchy (for more background on this hierarchy, see [17, Section 5.1]).

Overall, the results of Frei et al. [1] indicate that these problems are rather intractable and, assuming P $\neq$ NP (which, of course, is equivalent to P $\neq \Theta_2^P$), there exist no efficient algorithms solving them exactly. Considering the large number of real-world applications for those problems mentioned by Frei et al. [1], these results are rather disappointing and unsatisfying.

This obstacle motivates us to study, from a purely theoretical point of view, whether there are scenarios that allow for efficient solutions to these problems that, assuming P $\neq$ NP, are intractable. Our work is based on the assumption that most of the real-world applications of stability of graph parameters do not use arbitrarily complex graphs but may often be restricted to certain special graph classes. Consequently, our studies show that—despite the completeness results by Frei et al. [1]—there are tractable solutions to these problems when one limits the scope of the problem to a special graph class.

We start by analyzing these problems for graphs with bounded tree-width ($\mathcal{TW}_k$) or bounded clique-width ($\mathcal{CW}_k$), making use of known results for computing the common graph parameters α (the independence number), β (the vertex cover number), ω (the clique number), and χ (the chromatic number) in such graphs. In order to improve these general time bounds even further, we then focus on four classes of special graphs: trees ($\mathcal{T}$), forests ($\mathcal{F}$), bipartite graphs ($\mathcal{B}$), and co-graphs ($\mathcal{C}$). For each such class of graphs, we study twelve different problems: stability, vertex-stability, and unfrozenness for the four graph parameters α, β, ω, and χ.

Note This paper extends a preliminary version that appeared in the proceedings of the *22nd Italian Conference on Theoretical Computer Science* (ICTCS 2021) [18]. In

---

[1] More precisely, Frei et al. [1] prove that ξ-STABILITY, ξ-VERTEXSTABILITY, and ξ-UNFROZENNESS are Θ_2^P-complete for each of these four graph parameters, except β-VERTEXSTABILITY, which is in P, and χ-UNFROZENNESS, whose complexity is unknown.

[2] Θ_2^P is contained in the second level of the polynomial hierarchy and contains the problems that can be solved in polynomial time by an algorithm that accesses its NP oracle in parallel (i.e., it first computes all its queries and then asks them all at once and accepts its input depending on the answer vector). Alternatively, $\Theta_2^P = \mathrm{P}^{\mathrm{NP}[\mathcal{O}(\log n)]}$ (this equality is due to Hemaspaandra [14]) can be viewed as the class of problems solvable in polynomial time via *adaptively* accessing its NP oracle (i.e., computing the next query depending on the answer to the previous query) logarithmically often in the input size.

particular, Sections 4.1 and 4.2 about graphs of bounded tree-width and clique-width are new to the present version, which also contains many more details (e.g., specific running times), arguments, and discussion.

2 Preliminaries

We follow the notation of Frei et al. [1] and briefly collect the relevant notions here (referring to their paper [1] for further discussion). Let $\mathcal{G}$ be the set of all undirected, simple graphs without loops. For $G \in \mathcal{G}$, we denote by $V(G)$ its vertex set and by $E(G)$ its edge set; by $\overline{G}$ its complementary graph with $V(\overline{G}) = V(G)$ and $E(\overline{G}) = \{\{v, w\} \mid v, w \in V(G) \land v \neq w \land \{v, w\} \notin E(G)\}$. For $v \in V(G)$, $e \in E(G)$, and $e' \in E(\overline{G})$, let $G - v$, $G - e$, and $G + e'$, respectively, denote the graphs that result from G by deleting v (and its incident edges), deleting e, and adding e' (between any two vertices of G). Unless stated otherwise, we always use m to refer to the number of edges, $\overline{m}$ to refer to the number of nonedges of a graph, and we use n to refer to its number of vertices, i.e., $m = |E(G)|$, $\overline{m} = |E(\overline{G})|$, and $n = |V(G)|$.

A *graph parameter* is a map $\xi : \mathcal{G} \to \mathbb{N}$. As Frei et al. [1], we focus on the following four prominent graph parameters:

- the *independence number* α, i.e., the size of a maximum independent set—a subset of the vertices such that no two of them are connected by an edge,
- the *vertex cover number* β, i.e., the size of a minimum vertex cover—a subset of the vertices that contains at least one vertex of each edge,
- the *clique number* ω, i.e., the size of a maximum clique—a subset of the vertices such that any two of them are connected by an edge, and
- the *chromatic number* χ, i.e., the minimum number of colors needed to color the vertices of a graph so that no two adjacent vertices have the same color.

For a graph parameter ξ, an edge $e \in E(G)$ is said to be ξ-*stable* if $\xi(G) = \xi(G-e)$, i.e., $\xi(G)$ remains unchanged after e is deleted from G. Otherwise (i.e., if $\xi(G)$ is changed by deleting e), e is said to be ξ-*critical*. Stability and criticality are defined analogously for a vertex $v \in V(G)$ instead of an edge $e \in E(G)$.

A graph is said to be ξ-*stable* if all its edges are ξ-stable. A graph whose vertices (instead of edges) are all ξ-stable is said to be ξ-*vertex-stable*, and ξ-*criticality* and ξ-*vertex-criticality* are defined analogously. Obviously, each edge and each vertex is either stable or critical, yet a graph might be neither.

Traditionally, the analogous terms for stability or vertex stability when an edge or a vertex (together with an arbitrary neighborhood) is *added* rather than deleted are *unfrozenness* and *vertex unfrozenness*: They too indicate that a graph parameter does not change by this operation. And if, however, a graph parameter *changes* when an edge or vertex is *added* (not deleted), the notions analogous to criticality and vertex criticality are simply termed *frozenness* and *vertex frozenness*. Again, each edge and each vertex is either unfrozen or frozen, but a graph might be neither.

For any given graph parameter ξ, define the problem ξ-STABILITY to be the set of all ξ-stable graphs; and analogously so for the sets (problems) ξ-VERTEXSTABILITY, ξ-VERTEXCRITICALITY, ξ-UNFROZENNESS, ξ-FROZENNESS, and

ξ-VERTEXUNFROZENNESS. These are the decision problems studied by Frei et al. [1] for general graphs in terms of their computational complexity. We will study them when restricted to the special graph classes mentioned in the introduction, and to be formally defined in Section 4.

A graph class $\mathcal{J} \subseteq \mathcal{G}$ is *closed under (induced) subgraphs* if for every $G \in \mathcal{J}$, it holds that all (induced) subgraphs H of G satisfy $H \in \mathcal{J}$.

Perfect graphs were originally introduced by Berge [19] in 1963 (for more background, see also the book by Golumbic [20]): A graph G is called *perfect* if for all induced subgraphs H of G, we have $\chi(H) = \omega(H)$. Note that G is also an induced subgraph of itself.

We assume familiarity with the standard complexity classes P and NP. For more background on computational complexity (e.g., regarding the complexity classes DP and Θ_2^P mentioned in the introduction; note that $P \subseteq NP \subseteq DP \subseteq \Theta_2^P$ by definition), we refer to the textbooks by Papadimitriou [21] and Rothe [17].

For the sake of self-containment, we here state a well known result of Gallai [22], which is used to obtain several of our results.

Lemma 2.1 (Gallai [22]) *For every graph $G \in \mathcal{G}$, it holds that*

$$n = \alpha(G) + \beta(G).$$

3 General Stability and Unfrozenness Results

In this section, we provide general results that hold for specific graph classes satisfying special requirements. These results can be used to easily show that some stability or unfrozenness problems are tractable for various graph classes. As is common, we say a problem (a function) is *tractable* if it can be solved (computed) deterministically in polynomial time.

Theorem 3.1 *Let $\mathcal{J} \subseteq \mathcal{G}$ be a graph class closed under induced subgraphs, and let ξ be a tractable graph parameter for $\mathcal{J}$; say, $\xi(G)$ can be determined in time $\mathcal{O}(n^c)$ for $G \in \mathcal{J}$ and $c \in \mathbb{N}$. Then ξ-VERTEXSTABILITY can be solved in time $\mathcal{O}(n^{c+1})$ for $G \in \mathcal{J}$.*

Proof Let $G \in \mathcal{J}$ and compute $\xi(G)$. For all $v \in V(G)$, we have $G - v \in \mathcal{J}$, since $\mathcal{J}$ is closed under induced subgraphs. Hence, for all $v \in V(G)$, we can compute $\xi(G-v)$ efficiently and compare it to $\xi(G)$. If there is no vertex such that the values differ, G is ξ-vertex-stable. Thus, for all $G \in \mathcal{J}$, ξ-VERTEXSTABILITY can be solved in time $\mathcal{O}(n \cdot n^c)$.   □

Since every graph class that is closed under subgraphs is also closed under induced subgraphs, Corollary 3.2 is a simple consequence of the previous theorem. Slightly abusing common notation, we write "for all graphs in a class $\mathcal{J}$, some graph problem Π is in P" to mean that $\Pi_{|\mathcal{J}}$ is in P, where $\Pi_{|\mathcal{J}}$ denotes the graph problem Π restricted to graphs from the class $\mathcal{J}$.

Corollary 3.2 *Let $\mathcal{J} \subseteq \mathcal{G}$ be a graph class closed under subgraphs and ξ a tractable graph parameter for $\mathcal{J}$. Then, for all graphs in $\mathcal{J}$, ξ-VERTEXSTABILITY $\in$ P.*

The first theorem made a statement related to vertex stability about graph classes closed under induced subgraphs. Theorem 3.3 is related to (edge) stability instead. The easy proof of Theorem 3.3 is analogous and thus omitted (see also the technical report [23]).

Theorem 3.3 *Let $\mathcal{J} \subseteq \mathcal{G}$ be a graph class closed under subgraphs, and let ξ be a tractable graph parameter for $\mathcal{J}$; say, $\xi(G)$ can be determined in time $\mathcal{O}(n^c)$ for $G \in \mathcal{J}$ and $c \in \mathbb{N}$. Then ξ-STABILITY can be solved in time $\mathcal{O}(m \cdot n^c)$ for $G \in \mathcal{J}$.*

Some of the graph classes we study are classes of perfect graphs, which is why we now provide some results on vertex stability for perfect graphs.

Theorem 3.4 *A perfect graph G is ω-vertex-stable if and only if G is χ-vertex-stable.*

Proof Let G be a perfect graph. We have $\chi(G) = \omega(G)$ and for every vertex $v \in V(G)$, it holds that $\chi(G-v) = \omega(G-v)$, as $G-v$ is an induced subgraph of G. Consequently, we have

$$G \in \omega\text{-VertexStability} \Leftrightarrow \forall v \in V(G): \omega(G-v) = \omega(G)$$
$$\Leftrightarrow \forall v \in V(G): \chi(G-v) = \chi(G)$$
$$\Leftrightarrow G \in \chi\text{-VertexStability}.$$

This completes the proof.   □

The next corollary follows immediately from this result.

Corollary 3.5 *Let $\mathcal{J} \subseteq \mathcal{G}$ be a class of perfect graphs. Then, for all graphs in $\mathcal{J}$, we have χ-VertexStability $= \omega$-VertexStability.*

While the above results are related to the concepts of stability and vertex stability, the next two results address the property of unfrozenness.

Theorem 3.6 *Let $\mathcal{J} \subseteq \mathcal{G}$ be a graph class closed under complements and subgraphs. If α is tractable for $\mathcal{J}$, then for all graphs in $\mathcal{J}$, ω-UNFROZENNESS belongs to P.*

Proof Let us assume that we can efficiently compute $\alpha(G)$ for all $G \in \mathcal{J}$. Since $\mathcal{J}$ is closed under subgraphs, applying Theorem 3.3 yields that for all graphs in $\mathcal{J}$, α-STABILITY belongs to P. Let $G \in \mathcal{J}$ be a graph. As $\mathcal{J}$ is closed under complements, $\overline{G}$ also belongs to $\mathcal{J}$. Consequently, we can efficiently compute $\alpha(\overline{G})$. It is known from Frei et al. [1, Proposition 34(1)] that α-STABILITY $= \{\overline{G} \mid G \in \omega\text{-UNFROZENNESS}\}$. Hence, if $\overline{G}$ is α-stable, it immediately follows that G is ω-unfrozen. Therefore, for graphs in $\mathcal{J}$ we can decide efficiently whether they are ω-unfrozen, so for all graphs in $\mathcal{J}$ ω-UNFROZENNESS belongs to P.   □

Note that this theorem makes use of a relation between α-STABILITY and ω-UNFROZENNESS that analogously holds for β-STABILITY and ω-UNFROZENNESS via Lemma 2.1. The next theorem follows by a similar approach but exploits a relation between ω-STABILITY and both α-UNFROZENNESS and β-UNFROZENNESS.

Theorem 3.7 *Let $\mathcal{J} \subseteq \mathcal{G}$ be a graph class closed under complements and subgraphs. If ω is tractable for $\mathcal{J}$, then α-*UNFROZENNESS* and β-*UNFROZENNESS* $\in$ P *for all* $G \in \mathcal{J}$.*

Proof The argumentation is similar to that in the proof of Theorem 3.6. We know that $\mathcal{J}$ is closed under subgraphs and we can efficiently compute $\omega(G)$ for all $G \in \mathcal{J}$. Consequently, by Theorem 3.3 it follows that for all graphs in $\mathcal{J}$, ω-STABILITY belongs to P. Further, it is known from a result by Frei et al. [1, Proposition 34(2)] that

$$\alpha\text{-UNFROZENNESS} = \beta\text{-UNFROZENNESS} = \{\overline{G} \mid G \in \omega\text{-STABILITY}\}.$$

Let $G \in \mathcal{J}$ be a graph. As $\mathcal{J}$ is closed under complements, $\overline{G}$ is in $\mathcal{J}$, too. Now, efficiently decide whether $\overline{G}$ is ω-stable. If that is the case, it immediately follows that G is α-unfrozen and β-unfrozen. Hence, for all graphs in $\mathcal{J}$, both α-UNFROZENNESS and β-UNFROZENNESS, belong to P.   □

4 Tractability Results for Special Graph Classes

Ahead of our results for the individual graph classes, we provide two observations, which we will use multiple times in the upcoming proofs.

Observation 4.1 χ-VERTEXSTABILITY $\subseteq \chi$-STABILITY.

Proof Let G be any graph in χ-VERTEXSTABILITY. That is, G is χ-vertex-stable, i.e., every vertex $v \in V(G)$ is χ-stable. Frei et al. [1, Observation 3] observed that if $e = \{u, v\}$ is an edge in a graph and u or v are χ-stable, then so is e. Hence, it follows immediately that every edge $e \in E(G)$ is χ-stable, since all its incident vertices are χ-stable. If all edges of G are χ-stable, it follows that G is χ-stable and thus G is in χ-STABILITY.   □

Observation 4.2 *Let G be a graph. If an edge $\{u, v\} \in E(G)$ is β-critical, then u and v are β-critical, too.*

We omit the proof of Observation 4.2, as it is basically the same as that given by Frei et al. [1, Observation 2] for the analogous claim about χ-critical edges and vertices.

With these two observations we can now inspect several graph classes. In the following subsections we study the problems ξ-STABILITY, ξ-VERTEXSTABILITY, and ξ-UNFROZENNESS with $\xi \in \{\alpha, \beta, \omega, \chi\}$, restricted to special graph classes. Frei et al. [1] showed that for $\xi \in \{\alpha, \omega, \chi\}$ we have ξ-VERTEXUNFROZENNESS $= \emptyset$ as well as β-VERTEXUNFROZENNESS $= \{(\emptyset, \emptyset)\}$, where $(\emptyset, \emptyset)$ is the *null graph*, i.e., the graph without vertices or edges (which we will not further consider in this paper). This is why we do not study problems related to vertex unfrozenness, as all related questions are already answered.

4.1 Graphs of Bounded Tree-Width

We start with a famous, well-studied class of graphs: those having bounded tree-width. The notion of tree-width was defined in the 1980s by Robertson and Seymour [24] as

stated in Definition 4.1. First, we introduce the following well-known graph-theoretical notions. A graph G is said to be a *tree* if G is connected and has no cycles. Let $\mathcal{T}$ denote the class of all trees. G is said to be a *forest* if there exist vertex-disjoint trees $G_1, \ldots, G_t$ such that $G = G_1 \cup \cdots \cup G_t$. Let $\mathcal{F}$ denote the class of all forests.

Definition 4.1 (tree-width; Robertson and Seymour [24]) A *tree decomposition of a graph G* is a pair $(\mathcal{X}, T)$, where T is a tree and $\mathcal{X} = \{X_u \mid u \in V(T)\}$ is a family of subsets $X_u \subseteq V(G)$, one for each vertex u of T, such that the following three conditions hold true:

1. $\bigcup_{u \in V(T)} X_u = V(G)$.
2. For every edge $\{v_1, v_2\} \in E(G)$, there is some vertex $u \in V(T)$ such that $v_1 \in X_u$ and $v_2 \in X_u$.
3. For every vertex $v \in V(G)$ the subgraph of T induced by the vertices $u \in V(T)$ with $v \in X_u$ is connected.

The *width of a tree-decomposition* $(\mathcal{X} = \{X_u \mid u \in V(T)\}, \ T)$ is

$$\max_{u \in V(T)} |X_u| - 1.$$

The *tree-width of a graph G*, $tw(G)$ for short, is the smallest integer k such that there is a tree-decomposition $(\mathcal{X}, T)$ for G of width k.

We denote the class of all graphs of tree-width at most k by $\mathcal{TW}_k$. Next, we list some useful, well-known facts about graphs of bounded tree-width that can be found in the survey of Bodlaender [25, Theorem 65 and Lemma 11].

Proposition 4.3 (Bodlaender [25]) *$\mathcal{TW}_1$ is equal to the class of all forests, i.e., $\mathcal{TW}_1 = \mathcal{F}$.*

Lemma 4.4 (Bodlaender [25]) *Let G be a graph and H be a subgraph of G. Then, it holds that $tw(H) \leq tw(G)$.*

Lemma 4.5 *Let G be a graph and $e \in E(\overline{G})$. Then, it holds that $tw(G+e) \leq tw(G)+1$.*

Proof We can obtain a tree-decomposition for $G + e$ from a tree-decomposition $(\mathcal{X} = \{X_u \mid u \in V(T)\}, \ T)$ for G by adding one of the two end vertices of e into every set X_u. □

Determining whether the tree-width of some given graph is at most some given value is NP-complete even for bipartite graphs (to be discussed in Section 4.3) and complements of bipartite graphs [26].

Theorem 4.6 (Bodlaender [27]) *There is an algorithm that, given a graph G, computes a tree-decomposition of G of width $k = tw(G)$ in time $2^{\mathcal{O}(k^3)} \cdot n$.*

Thus, for every fixed positive integer k, one can determine in linear time whether the tree-width of a given graph G is at most k, and if so, find a tree-decomposition of width at most k.

Theorem 4.7 (Hliněný et al. [28]) *Determining the size of a largest independent set in a graph G of tree-width at most k, assuming a tree-decomposition of G is given, can be done in time $\mathcal{O}(2^k \cdot n)$.*

By Lemma 2.1, we can compute β from α, which leads to the following results.

Corollary 4.8 *Determining the size of a smallest vertex cover for a graph G of tree-width at most k, assuming a tree-decomposition of G is given, can be done in time $\mathcal{O}(2^k \cdot n)$.*

By storing information about cliques instead of independent sets, the solution given by Hliněný et al. [28, Algorithm 1.2] can be suitably modified to show the next result.

Theorem 4.9 *Determining the size of a largest clique in a graph G of tree-width at most k, assuming a tree-decomposition of G is given, can be done in time $\mathcal{O}(2^k \cdot n)$.*

The following result due to Cygan et al. [29, Theorem 7.10.] gives a different time bound for the chromatic number.

Theorem 4.10 (Cygan et al. [29]) *Determining the chromatic number of a graph G of tree-width at most k, assuming a tree-decomposition of G is given, can be done in time $k^{\mathcal{O}(k)} \cdot n$.*

Next, we give stability, vertex stability, and unfrozenness results for graphs of tree-width at most k. Regarding stability and vertex stability, Corollaries 4.11 and 4.12 follow immediately from Theorems 3.3, 4.7, 4.9, and 4.10, Corollary 4.8, and Lemma 4.4.

Corollary 4.11 *For every graph in $\mathcal{TW}_k$, assuming a tree-decomposition of width at most k is given, ξ-STABILITY can be solved in time $\mathcal{O}(2^k mn)$ for all $\xi \in \{\alpha, \beta, \omega\}$ and χ-STABILITY can be solved in time $\mathcal{O}(k^{\mathcal{O}(k)} mn)$. Thus, for every fixed positive integer k, these problems belong to P.*

Corollary 4.12 *For every graph in $\mathcal{TW}_k$, assuming a tree-decomposition of width at most k is given, ξ-VERTEXSTABILITY can be solved in time $\mathcal{O}(2^k n^2)$ for all $\xi \in \{\alpha, \beta, \omega\}$ and χ-VERTEXSTABILITY can be solved in time $\mathcal{O}(k^{\mathcal{O}(k)} n^2)$. Thus, for every fixed positive integer k, these problems belong to P.*

Our unfrozenness results in the next theorem can also be easily shown.

Theorem 4.13 *For every graph in $\mathcal{TW}_k$, assuming a tree-decomposition of width at most k is given, ξ-UNFROZENNESS can be solved in time $\mathcal{O}(2^k n^3)$ for all $\xi \in \{\alpha, \beta, \omega\}$ and χ-UNFROZENNESS can be solved in time $\mathcal{O}(k^{\mathcal{O}(k)} n^3)$. Thus, for every fixed positive integer k, these problems belong to P.*

Proof Let $G \in \mathcal{TW}_k$ and compute $\xi(G)$ by Theorem 4.7, Corollary 4.8, Theorem 4.9, or Theorem 4.10. For all $e \in E(\overline{G})$, we have $G + e \in \mathcal{TW}_{k+1}$ by Lemma 4.5. Thus, for all $e \in E(\overline{G})$, we can compute $\xi(G + e)$ efficiently and compare it to $\xi(G)$. If there is no $e \in E(\overline{G})$ such that the values differ, G is ξ-unfrozen. □

4.2 Graphs of Bounded Clique-Width

Next, we turn to graphs of bounded clique-width. Clique-width measures the difficulty of decomposing a graph into a special tree structure. The clique-width of graphs has been defined by Courcelle and Olariu [30] as follows.

Definition 4.2 (clique-width; Courcelle and Olariu [30]) The *clique-width of a graph* G is the minimum number of labels needed to define G using the following four operations:

1. Creation of a new vertex v with label a (denoted by $a(v)$).
2. Disjoint union of two labeled vertex-disjoint graphs G and H (denoted by $G \cup H$).
3. Inserting an edge between every vertex with label a and every vertex with label b ($a \neq b$, denoted by $\eta_{a,b}$).
4. Changing label a into label b (denoted by $\rho_{a \to b}$).

We denote the class of all graphs of clique-width at most k by $\mathcal{CW}_k$.

An expression using these four operations and at most k different labels is called a *k-expression*. Thus the *clique-width* of graph G, cw(G) for short, is the least integer k such that there is a k-expression for G. Let graph(X) be the graph defined by k-expression X.

Example 4.1 (k-expression) The expression

$$X = \eta_{2,3}(\rho_{3 \to 2}(\rho_{2 \to 1}(\eta_{2,3}(\eta_{1,2}(1(v_1) \cup 2(v_2)) \cup 3(v_3)))) \cup 3(v_4))$$

is a 3-expression and graph(X) is a path on four vertices, i.e., $P_4 \in \mathcal{CW}_3$.

By this definition, every graph of clique-width at most k can be represented by a tree structure, denoted as *k-expression-tree*. The leaves of the k-expression-tree represent the vertices of the graph and the inner vertices of the k-expression-tree correspond to the operations applied to the subexpressions defined by the subtrees. Using the k-expression-tree numerous hard problems have been shown to be solvable in polynomial time when restricted to graphs of bounded clique-width [31–33].

Note that graphs of bounded clique-width generalize co-graphs, which we will define and study in Section 4.5, denoting by $\mathcal{C}$ the class of co-graphs.

Proposition 4.14 (Courcelle and Olariu [30]) *$\mathcal{CW}_2$ is equal to the class of all co-graphs, i.e., $\mathcal{CW}_2 = \mathcal{C}$.*

Next, we list some further useful, well-known facts about graphs of bounded clique-width.

Theorem 4.15 (Corneil and Rotics [34]) *Every graph of tree-width at most k has clique-width at most $3 \cdot 2^{k-1}$.*

As $\mathcal{CW}_1 = \mathcal{I}$ (the class of empty graphs consisting of only isolated vertices and no edges) and $\mathcal{CW}_2 = \mathcal{C}$ (the class of co-graphs), it follows that for $k \geq 1$, $\mathcal{CW}_k$ is not closed under complementation. By the following result, however, complementing graphs can only double their clique-width.

Lemma 4.16 (Courcelle and Olariu [30]) *For every graph G, $cw(\overline{G}) \leq 2 \cdot cw(G)$, and for a given k-expression X for G, one can compute a $2k$-expression for $\overline{G}$ in time $\mathcal{O}(|X|)$.*

As every complete graph K_n is in $\mathcal{CW}_2$, it follows that for $k \geq 2$, $\mathcal{CW}_k$ is not closed under subgraphs; it is closed, however, under induced subgraphs.

Lemma 4.17 *Let G be a graph and H be an induced subgraph of G. Then $cw(H) \leq cw(G)$.*

Lemma 4.18 (Gurski [35]) *Let G be a graph and $e \in E(G)$ and $e' \in E(\overline{G})$. Then $cw(G - e) \leq cw(G) + 2$ and $cw(G + e') \leq cw(G) + 2$.*

Determining whether the clique-width of some given graph is at most some given value is NP-complete [36].

There are graph classes for which we can compute a clique-width expression in linear time. This has been shown by Courcelle and Olariu [30] for co-graphs which have clique-width at most 2 and by Golumbic and Rotics [37] for distance-hereditary graphs which have clique-width at most 3, and it is also known for several graph classes allowing only a few P_4, such as for $(q, q - 4)$-graphs which have clique-width at most q and for P_4-tidy graphs which have clique-width at most 4, as shown by Courcelle et al. [38].

In order to obtain an expression for general graphs, a useful result of Oum and Seymour [39] allows to approximate clique-width expressions defined by a rank-decomposition, with the drawback of a single-exponential blow-up on the parameter.

Theorem 4.19 (Oum [40]) *There is an algorithm that computes, given a graph G of clique-width k, a clique-width $(8^k - 1)$-expression for G in time $\mathcal{O}(n^3)$.*

Now we turn specifically to the graph parameters studied in this paper.

Theorem 4.20 (Gurski [41]) *Finding the size of a largest independent set in a graph G of clique-width at most k, assuming a clique-width k-expression of G is given, can be done in time $\mathcal{O}(2^{2k} \cdot n)$.*

By Lemma 2.1 we can compute β from α, which leads to the following results.

Corollary 4.21 *Finding the size of a smallest vertex cover of a graph G of clique-width at most k, assuming a clique-width k-expression of G is given, can be done in time $\mathcal{O}(2^{2k} \cdot n)$.*

Theorem 4.22 (Gurski [41]) *Finding the size of a largest clique in a graph G of clique-width at most k, assuming a clique-width k-expression of G is given, can be done in time $\mathcal{O}(2^{2k} \cdot n)$.*

A different time bound is known for the chromatic number.

Theorem 4.23 (Kobler and Rotics [33]) *Finding the chromatic number in a graph G of clique-width at most k, assuming a clique-width k-expression of G is given, can be done in time $\mathcal{O}(2^{3k} \cdot k^2 \cdot n^{2^{2k+1}+1})$.*

Next we give stability, vertex stability, and unfrozenness results for graphs of clique-width at most k.

Theorem 4.24 *For every graph in $\mathcal{CW}_k$, assuming a clique-width k-expression is given, ξ-STABILITY can be solved in time $\mathcal{O}(2^{2k}mn)$ for all $\xi \in \{\alpha, \beta, \omega\}$ and χ-STABILITY can be solved in time $\mathcal{O}(2^{3k}k^2mn^{2^{2k+5}+1})$. Thus, for every fixed positive integer k, these problems belong to* P.

Proof Given a graph $G \in \mathcal{CW}_k$, compute $\xi(G)$ by Theorem 4.20, Corollary 4.21, Theorem 4.22, or Theorem 4.23. For all edges $e \in E(G)$, we have $G - e \in \mathcal{CW}_{k+2}$ by Lemma 4.18. Thus, for all $e \in E(G)$, we can compute $\xi(G - e)$ efficiently and compare it to $\xi(G)$. If there is no $e \in E(G)$ such that the values differ, G is ξ-stable. □

The next result follows from Theorems 3.1, 4.20, 4.22, and 4.23, Corollary 4.21, and Lemma 4.17.

Corollary 4.25 *For every graph in $\mathcal{CW}_k$, assuming a clique-width k-expression is given, ξ-VERTEXSTABILITY can be solved in time $\mathcal{O}(2^{2k}n^2)$ for all $\xi \in \{\alpha, \beta, \omega\}$ and χ-VERTEXSTABILITY can be solved in time $\mathcal{O}(2^{3k}k^2n^{2^{2k+1}+2})$. Thus, for every fixed positive integer k, these problems belong to* P.

Theorem 4.26 *For every graph in $\mathcal{CW}_k$, assuming a clique-width k-expression is given, ξ-UNFROZENNESS can be solved in time $\mathcal{O}(2^{2k}n^3)$ for all $\xi \in \{\alpha, \beta, \omega\}$ and χ-UNFROZENNESS can be solved in time $\mathcal{O}(2^{3k}k^2n^{2^{2k+5}+3})$. Thus, for every fixed positive integer k, these problems belong to* P.

Proof Given a graph $G \in \mathcal{CW}_k$, compute $\xi(G)$ by Theorem 4.20, Corollary 4.21, Theorem 4.22, or Theorem 4.23. For all edges $e \in E(\overline{G})$, we have $G + e \in \mathcal{CW}_{k+2}$ by Lemma 4.18. Thus, for all $e \in E(\overline{G})$, we can compute $\xi(G+e)$ efficiently and compare it to $\xi(G)$. If there is no $e \in E(\overline{G})$ such that the values differ, G is ξ-unfrozen. □

Summing up our results so far, we have shown that even though the twelve problem variants we investigate—STABILITY, VERTEXSTABILITY, and UNFROZENNESS for the four graph parameters α, β, ω, and χ—are hard to solve in general by the results of Frei et al. [1], they can each be solved in polynomial time when restricted to graphs of bounded tree-width or clique-width. However, the specific time bounds obtained by these general results may still be improvable. Therefore, in the following sections we will have a closer look at some other special graph classes which, as already mentioned, are contained in $\mathcal{TW}_k$ and $\mathcal{CW}_k$: trees and forests, and co-graphs. In each case, we will seek to provide better time bounds for the above-mentioned problems than those given in the previous sections. Before doing so, however, we have a closer look at another interesting class of graphs, the bipartite graphs.

4.3 Bipartite Graphs

$G = (V_1 \cup V_2, E)$ is a *bipartite graph* if $V_1 \cap V_2 = \emptyset$ and $E \subseteq \{\{u, v\} \mid u \in V_1 \wedge v \in V_2\}$. Since bipartite graphs in particular contain grid graphs, they do not have bounded

tree-width or bounded clique-width. Denote the class of all bipartite graphs by $\mathcal{B}$. We begin with two simple observations. Observation 4.27 is so evident that we omit providing a proof.

Observation 4.27 *Let G be a bipartite graph. Then $\chi(G) = \omega(G) = 1$ if $E(G) = \emptyset$, and $\chi(G) = \omega(G) = 2$ if $E(G) \neq \emptyset$.*

Consequently, we can efficiently compute χ and ω for all bipartite graphs.

The proof of the next observation provides a method to efficiently compute α and β for bipartite graphs as well, by making use of the algorithm due to Hopcroft and Karp [42] and the result of Kőnig [43].[3]

Observation 4.28 *For each bipartite graph G, we can compute $\alpha(G)$ and $\beta(G)$ in time $\mathcal{O}(n^{2.5})$.*

Furthermore, as the class of bipartite graphs is closed under subgraphs and induced subgraphs, the following corollary follows from Theorems 3.3 and 3.1.

Corollary 4.29 *For every $\xi \in \{\alpha, \beta, \omega, \chi\}$ and all bipartite graphs the problems ξ-STABILITY and ξ-VERTEXSTABILITY are in P. In particular, for bipartite graphs and $\xi \in \{\alpha, \beta\}$, ξ-STABILITY can be solved in time $\mathcal{O}(n^{2.5}m)$ and ξ-VERTEXSTABILITY in time $\mathcal{O}(n^{3.5})$.*

Note that specific time bounds for ξ-STABILITY and ξ-VERTEXSTABILITY with $\xi \in \{\omega, \chi\}$ are implied later on by Propositions 4.31 and 4.33 and by Theorems 4.32 and 4.34.

Next, we discuss approaches for how to decide whether a bipartite graph is stable. If a bipartite graph G has no edges, it is trivial to handle [23]. For bipartite graphs with one edge, we have the following simple result.

Proposition 4.30 *Let G be a bipartite graph with $m = 1$ edge. Then G is neither ξ-stable nor ξ-vertex-stable for $\xi \in \{\alpha, \beta, \omega, \chi\}$.*

Proof Denote G's only edge by $e = \{u, v\}$ for $u, v \in V(G)$. Then G is neither α-stable nor α-vertex-stable, as we have $\alpha(G) = n - 1$, yet $\alpha(G - e) = n$ and $\alpha(G - w) = n - 2$ for $w \in V(G) \setminus \{u, v\}$. Consequently, G is neither β-stable, which follows from a result by Frei et al. [1, Proposition 34], nor β-vertex-stable because of $\beta(G) = 1$ and $\beta(G - u) = 0$. Furthermore, we have $\omega(G) = 2$ as well as $\omega(G - e) = \omega(G - u) = 1$, so G is neither ω-stable nor ω-vertex-stable. Lastly, $\chi(G) = 2$ but $\chi(G - e) = \chi(G - u) = 1$, so G is neither χ-stable nor χ-vertex-stable.   □

We now provide results for bipartite graphs with more than one edge.

Proposition 4.31 *Every bipartite graph G with $m \geq 2$ edges is ω-stable.*

---

[3] Note that Observation 4.28 is an exercise in the book by Cormen et al. [44] (of course, using improved matching algorithms for bipartite graphs such as that of Alt et al. [45] would improve the running time even further).

Proof For all $e \in E(G)$, we have $\omega(G - e) = 2 = \omega(G)$, as $E(G - e) \neq \emptyset$, so G is ω-stable. □

Furthermore, we can also characterize ω-vertex-stability.

Theorem 4.32 *Let G be a bipartite graph with $m \geq 2$ edges. G is ω-vertex-stable if and only if for all $v \in V(G)$, it holds that $\deg(v) < m$.*

Proof Assume that G is ω-vertex-stable. That is, for all $v \in V(G)$, it holds that $\omega(G - v) = 2 = \omega(G)$. If there were one $v \in V(G)$ with $\deg(v) = m$, we would have $\omega(G - v) = 1$ as $E(G - v) = \emptyset$, a contradiction to G's ω-vertex-stability. Conversely, assume that $\deg(v) < m$ for all $v \in V(G)$. Then, for all $v \in V(G)$, it follows that $E(G - v) \neq \emptyset$. Consequently, $\omega(G) = 2 = \omega(G - v)$ and, hence, G is ω-vertex-stable. □

The proof of the following proposition is similar to that of Proposition 4.31.

Proposition 4.33 *Every bipartite graph G with $m \geq 2$ edges is χ-stable.*

Proof Let $e \in E(G)$ be an arbitrary edge of G. Since $E(G - e) \neq \emptyset$, it holds that $\chi(G - e) = 2 = \chi(G)$ and, thus, G is χ-stable. □

Lastly, we can characterize χ-vertex-stability.

Theorem 4.34 *Let G be a bipartite graph with $m \geq 2$ edges. G is χ-vertex-stable if and only if for all $v \in V(G)$, it holds that $\deg(v) < m$.*

Proof Assume G to be χ-vertex-stable. Furthermore, as we assume that $m \geq 2$, it holds that $\chi(G) = 2$. Then there cannot exist a vertex $v \in V(G)$ with $\deg(v) = m$, as such a vertex would be χ-critical, since $\chi(G - v) = 1$ because of $E(G - v) = \emptyset$. For the opposite direction, assume that $\deg(v) < m$ for all vertices $v \in V(G)$. Hence, no matter what vertex $v \in V(G)$ we remove from G, it always holds that $E(G - v) \neq \emptyset$, so $\chi(G - v) = 2 = \chi(G)$ and, thus, G is χ-vertex-stable. □

Besides these (vertex) stability characterizations for bipartite graphs, we now address unfrozenness for them.

Theorem 4.35 *Let G be a bipartite graph. G is χ-unfrozen if and only if G possesses no P_3 as an induced subgraph.*

Proof First, assume that G is χ-unfrozen but contains P_3 as an induced subgraph. Write $V(P_3) = \{v_1, v_2, v_3\}$ and $E(P_3) = \{\{v_1, v_2\}, \{v_2, v_3\}\}$ for the corresponding vertices and edges. Then $e = \{v_1, v_3\} \in E(\overline{G})$ is not an edge in G. However, adding e to G we obtain $\chi(G) = 2 < 3 = \chi(G + e)$, as $P_3 + e$ forms a 3-clique in G, a contradiction to the assumption that G is χ-unfrozen. Conversely, assume that G possesses no P_3 as an induced subgraph but is not χ-unfrozen. Hence, there must exist a nonedge $e = \{u, v\} \in E(\overline{G})$ such that $\chi(G + e) = 3 > 2 = \chi(G)$. Denote the two disjoint vertex sets of G by $V_1 \cup V_2 = V(G)$. Obviously, $u \in V_1$ and $v \in V_2$ cannot be true, since then $\chi(G + e) = 2$ would hold. Therefore, without loss of generality, we may assume that $u, v \in V_1$. Adding e to G must create a cycle of odd length in G, as cycles of even length as well as paths can be colored with two colors. Consequently, $G + e$ possesses a cycle C_n with $n = 2k + 1 \geq 3$ vertices, $k \in \mathbb{N}$, as a subgraph. This implies that G must possess P_3 as an induced subgraph, again a contradiction. □

Slightly modifying (the direction from right to left in) the previous proof yields Corollary 4.36. This time, adding e to G must create a 3-clique in G.

Corollary 4.36 *Let G be a bipartite graph. G is ω-unfrozen if and only if G possesses no P_3 as an induced subgraph.*

Both results show that ω- and χ-UNFROZENNESS can be solved in time $\mathcal{O}(n)$ and thus belong to P for bipartite graphs. For the proof of Theorem 4.38, which establishes the same result for β-UNFROZENNESS, we need the following lemma.

Lemma 4.37 *Let G be a bipartite graph and $u \in V(G)$. Then, $\beta(G-u) = \beta(G)-1$ if and only if there exists some vertex cover $V' \subseteq V(G)$ with $u \in V'$ and $|V'| = \beta(G)$.*

Proof Denote by $V'' \subseteq V(G-u)$ a minimum vertex cover for $G-u$ with $|V''| = \beta(G-u)$ and write $V' = V'' \cup \{u\}$. Then $|V'| = \beta(G-u)+1 = \beta(G)$ and $E(G-u) \cup \{e \in E(G) \mid e \cap \{u\} \neq \emptyset\} = E(G)$. As V'' is a minimum vertex cover for $G-u$, all edges in $E(G-u)$ are covered by V''. All remaining edges in $\{e \in E(G) \mid e \cap \{u\} \neq \emptyset\}$ are covered by u, so that V' covers all edges in $E(G)$ and is a vertex cover of G.

Conversely, let $V' \subseteq V(G)$ with $u \in V'$ be a minimum vertex cover for G, i.e., $|V'| = \beta(G)$. Removing u from G also removes all edges incident to u from $E(G)$; hence, $V' \setminus \{u\}$ is a minimum vertex cover for $G-u$. Consequently, we have $\beta(G-u) = |V'|-1 = \beta(G)-1$.   □

Theorem 4.38 *For all bipartite graphs, β-UNFROZENNESS can be solved in time $\mathcal{O}(n^{4.5})$ and thus belongs to P.*

Proof Let G be a bipartite graph with $V(G) = V_1 \cup V_2$ and $V_1 \cap V_2 = \emptyset$. Then we have $E \subseteq \{\{u,v\} \mid u \in V_1 \wedge v \in V_2\}$ and, according to Observation 4.28, we can compute $\beta(G)$ in time $\mathcal{O}(n^{2.5})$. For any nonedge $e = \{u,v\} \in E(\overline{G})$, either (1) $e \in \{\{u,v\} \mid u \in V_1 \wedge v \in V_2\}$ or (2) $e \in \{\{u,v\} \mid u \in V_i \wedge v \in V_i\}, i \in \{1,2\}$, must hold. We study both cases separately:

Case 1: If $e \in \{\{u,v\} \mid u \in V_1 \wedge v \in V_2\}$, then $G+e$ is a bipartite graph, so we can efficiently compute $\beta(G+e)$ and compare it with $\beta(G)$ to determine whether e is β-unfrozen or β-frozen.

Case 2: Without loss of generality, assume $e \in \{\{u,v\} \mid u \in V_1 \wedge v \in V_1\}$. Then two cases are possible:

Case 2.a: $G+e$ is bipartite by a modified vertex partition $V(G) = V_1' \cup V_2'$. This is possible if and only if $\chi(G+e) = 2$, which can be checked efficiently. In this case, we can compute $\beta(G+e)$ efficiently to determine whether e is β-unfrozen or β-frozen.

Case 2.b: $G+e$ is not a bipartite graph since it contains a cycle of odd length as a subgraph. In this case, we check with Lemma 4.37 for u, and afterwards for v, whether there exists some minimum vertex cover $V' \subseteq V(G)$ for G with $u \in V'$ or $v \in V'$, respectively. If one of these two checks is positive, we know that $\beta(G+e) = \beta(G)$ holds and hence, e is β-unfrozen. Otherwise, $\beta(G+e) = \beta(G)+1$ must hold, so e is β-frozen.

Doing so, we can check every nonedge $e \in E(\overline{G})$ efficiently for β-unfrozenness. Since there are $\mathcal{O}(n^2)$ nonedges and we can compute $\beta(G)$ in time $\mathcal{O}(n^{2.5})$, the running time follows.   □

The proof of Theorem 4.38 allows for every nonedge of a bipartite graph to decide whether or not it is β-unfrozen, so β-FROZENNESS too is in P (with the same running time of $\mathcal{O}(n^{4.5})$) for bipartite graphs. By Lemma 2.1, α-UNFROZENNESS is in P for bipartite graphs as well, again with the same running time of $\mathcal{O}(n^{4.5})$.

Corollary 4.39 *For all bipartite graphs, α-UNFROZENNESS and β-FROZENNESS can be solved in time $\mathcal{O}(n^{4.5})$ and they thus belong to* P.

4.4 Trees and Forests

We now investigate trees and forests. While each tree is a forest, each forest is a bipartite graph and a graph of tree-width 1 (Proposition 4.3). Thus all results of Sections 4.3 and 4.1 also hold for trees and forests, but some running times can be improved as follows.

For every tree T, it holds that $m = n - 1$ (see, e.g., Bollobás [46]). So, we have $\omega(T) = \chi(T) = 2$ if $n > 1$, and $\omega(T) = \chi(T) = 1$ if $n = 1$.

Also, there exist algorithms to determine $\alpha(T)$ for trees T in time $\mathcal{O}(n)$ (see, e.g., [47, Section 10.2]). Then, using Lemma 2.1, we can compute β for trees in time $\mathcal{O}(n)$. Summing up, all four graph parameters α, β, ω, and χ are computable in time $\mathcal{O}(n)$ for trees.

Now, let F with $F = T_1 \cup \cdots \cup T_t$ and $T_i \in \mathcal{T}$, $1 \leq i \leq t$, be a forest. It is easy to check that $\alpha(F) = \sum_{i=1}^{t} \alpha(T_i)$, $\beta(F) = \sum_{i=1}^{t} \beta(T_i)$, $\omega(F) = \max_{1 \leq i \leq t} \omega(T_i)$, and $\chi(F) = \max_{1 \leq i \leq t} \chi(T_i)$. Thus even for forests, all four graph parameters α, β, ω, and χ are computable in time $\mathcal{O}(n)$. Furthermore, the class of forests $\mathcal{F}$ is closed under subgraphs and induced subgraphs. From these observations and Theorems 3.3 and 3.1 we have the following results.

Theorem 4.40 *Let $\xi \in \{\alpha, \beta, \omega, \chi\}$ be a graph parameter. For all forests, the problems ξ-STABILITY and ξ-VERTEXSTABILITY can be solved in time $\mathcal{O}(n^2)$ and thus belong to* P.

For the two parameters ω and χ, we have even better running times for bipartite graphs by Propositions 4.31 and 4.33 and Theorems 4.32 and 4.34, which immediately also hold for the special case of forests.

Corollary 4.41 *Let $\xi \in \{\omega, \chi\}$ be a graph parameter. The problem ξ-STABILITY can be solved in time $\mathcal{O}(1)$ and the problem ξ-VERTEXSTABILITY can be solved in time $\mathcal{O}(n)$ for all forests.*

The next two corollaries follow immediately since each tree is a forest.

Corollary 4.42 *For all trees and for each $\xi \in \{\alpha, \beta, \omega, \chi\}$, the problems ξ-STABILITY and ξ-VERTEXSTABILITY can be solved in time $\mathcal{O}(n^2)$ and thus belong to* P.

Corollary 4.43 *Let $\xi \in \{\omega, \chi\}$ be a graph parameter. The problem ξ-STABILITY can be solved in time $\mathcal{O}(1)$ and the problem ξ-VERTEXSTABILITY can be solved in time $\mathcal{O}(n)$ for all trees.*

We now focus on the unfrozenness problems. All trees and forests with fewer than three vertices are trivial to handle (see the technical report [23] for details). It remains to study trees and forests with at least three vertices. Let P_i denote the path with i vertices.

Proposition 4.44 *Every tree T with $n \geq 3$ vertices is neither ω- nor χ-unfrozen.*

Proof With $n \geq 3$ and $m = n - 1$ it follows that T must contain P_3 as an induced subgraph. Denote the corresponding vertices by v_1, v_2, and v_3, so we have edges $\{v_1, v_2\}$ and $\{v_2, v_3\}$ in T. Being a tree, T does not contain any cycle of length greater than or equal to three, so $e = \{v_3, v_1\}$ cannot be an edge of T. Adding e to T creates the 3-clique with vertices v_1, v_2, v_3 in $T + e$, so we obtain

$$\omega(T) = 2 < 3 = \omega(T + e).$$

Hence, e is ω-frozen and thus T cannot be ω-unfrozen. A similar argument yields that T cannot be χ-unfrozen. □

Based on this result we can deduce whether forests are ω- or χ-unfrozen. As forests without edges are empty graphs, we study forests with at least one edge.

Theorem 4.45 *If $F \in \mathcal{F}$ contains P_2 but no P_3 as induced subgraphs, F is ω- and χ-unfrozen. If F contains P_3 as an induced subgraph, F is neither ω- nor χ-unfrozen.*

Proof We prove both statements separately. If F contains P_2 but no P_3 as an induced subgraph, we have $\omega(F) = \chi(F) = 2$. Let $e = \{u, v\} \in E(\overline{F})$ be a nonedge of F. Both vertices u, v of e satisfy one of two cases: Either the vertex is isolated or part of some P_2 in F. In both cases, adding e to F does not create a 3-clique, such that $\omega(F + e) = \chi(F + e) = 2$ still holds and e is ω- and χ-unfrozen. Since that holds for all nonedges of F, it follows that F is ω- and χ-unfrozen. On the other hand, if F contains P_3 as an induced subgraph, we can follow the same arguments as in the proof of Proposition 4.44 to see that F is neither ω- nor χ-unfrozen. □

Theorem 4.45 allows to decide whether a forest is ω- or χ-unfrozen by considering the vertex degrees.

Theorem 4.46 *Let $\xi \in \{\omega, \chi\}$ be a graph parameter. For all forests, the problem ξ-UNFROZENNESS can be solved in time $\mathcal{O}(n)$ and thus belongs to P.*

Finally, the next corollary follows by Corollary 4.39, $\mathcal{T} \subseteq \mathcal{F} \subseteq \mathcal{B}$, and the fact that for trees and forests we can compute β in time $\mathcal{O}(n)$.

Corollary 4.47 *For all trees and forests, the problems α-UNFROZENNESS and β-UNFROZENNESS as well as the problem β-FROZENNESS can be solved in time $\mathcal{O}(n^3)$ and thus belong to P.*

4.5 Co-Graphs

First of all, we recursively define co-graphs, using a slightly adapted definition by Corneil et al. [48].

Definition 4.3 (co-graph; Corneil et al. [48]) The graph $G = (\{v\}, \emptyset)$ is a *co-graph*. If G_1 and G_2 are vertex-disjoint co-graphs, then $G_1 \cup G_2$ and $G_1 + G_2$ are co-graphs, too.

We denote the class of all co-graphs by $\mathcal{C}$ and use the operators $\cup$ and $+$ as is common (see, e.g., [1]).

Since every co-graph is a graph of clique-width at most 2 (Proposition 4.14) all results of Section 4.2 also hold for co-graphs, but some running times can be improved as follows.

We will use the following result by Corneil et al. [48].

Theorem 4.48 (Corneil et al. [48]) *Co-graphs are (i) closed under complements and (ii) closed under induced subgraphs, but (iii) not closed under subgraphs. Furthermore, G is a co-graph if and only if G possesses no P_4 as an induced subgraph.*

Property (iii) has not been explicitly proven by Corneil et al. [48]. However, C_4 (the cycle on four vertices) is an easy example since C_4 is a co-graph (see Example 4.2 below), and removing one edge yields P_4. Since every co-graph can be constructed by $\cup$ and $+$, we can represent a co-graph by its *co-expression*.

Example 4.2 (co-expression) The co-expression $X = (v_1 \cup v_3) + (v_2 \cup v_4)$ describes the graph $C_4 = (\{v_1, v_2, v_3, v_4\}, \{\{v_1, v_2\}, \{v_2, v_3\}, \{v_3, v_4\}, \{v_4, v_1\}\})$.

Obviously, we can build a binary tree for every co-graph via its co-expression. The tree's leaves correspond to the graph's vertices and the inner vertices of the tree correspond to the expression's operations. For example, the tree in Fig. 1 corresponds to the co-expression from Example 4.2 and, thus, describes a C_4. Such a tree is called a *co-tree*. To prove our results regarding stability, vertex stability, and unfrozenness of co-graphs, we need the following result of Corneil et al. [49].

Theorem 4.49 (Corneil et al. [49]) *For every graph G, in time $\mathcal{O}(n+m)$ we can decide whether G is a co-graph and, if so, provide a corresponding co-tree.*

Combining the previous results with the next one by Corneil et al. [48], we can efficiently determine a co-graph's chromatic number.

Fig. 1 Co-tree for C_4
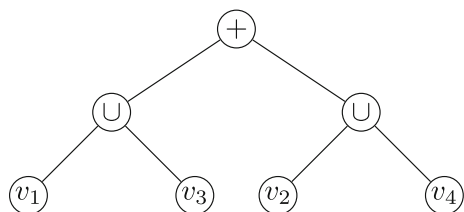
Theorem 4.50 (Corneil et al. [48]) *Let $G \in \mathcal{G}$ be a co-graph and T the corresponding co-tree. For a vertex w of T, denote by $G[w]$ the graph induced by the subtree of T with root w. To every leaf v of T we add as a label $\chi(G[v]) = 1$. For every inner vertex w of T, we add, depending on the inner vertex's type, the following label: (1) If w is a $\cup$-vertex with children v_1 and v_2, $\chi(G[w]) = \max\{\chi(G[v_1]), \chi(G[v_2])\}$, and (2) if w is a $+$-vertex with children v_1 and v_2, $\chi(G[w]) = \chi(G[v_1]) + \chi(G[v_2])$. If r is the root vertex of T, then it holds that $\chi(G[r]) = \chi(G)$.*

A result similar to the previous one for the independence number α was given by Corneil et al. [48] as well.

Remark 4.1 We label all leaves of T with $\alpha(G[v]) = 1$. Each inner vertex w of T with children v_1 and v_2 is labeled by $\alpha(G[w]) = \max\{\alpha(G[v_1]), \alpha(G[v_2])\}$ if w contains the $+$ operation, and by $\alpha(G[w]) = \alpha(G[v_1]) + \alpha(G[v_2])$ if w contains the $\cup$ operation. For the root r of T, it then holds that $\alpha(G[r]) = \alpha(G)$.

Note that the formulas for computing χ and α, which are stated in Theorem 4.50 and Remark 4.1, are not limited to co-graphs but also hold for general graphs; however, we will apply them only to co-graphs.

By Remark 4.1, we can efficiently compute α for co-graphs and thus obtain the following results.

Theorem 4.51 *For every co-graph, χ-VERTEXSTABILITY can be solved in time $\mathcal{O}(n^2)$ and thus is in* P.

Proof Let G be a co-graph. According to Theorem 4.49, we can compute the graph's co-tree T in time $\mathcal{O}(n + m)$. Now, compute $\chi(G)$ according to Theorem 4.50 in time $\mathcal{O}(n)$. Since co-graphs are closed under induced subgraphs, $G - v$ is a co-graph, too, for every $v \in V(G)$. Thus we can compute $\chi(G - v)$ for every $v \in V(G)$ in time $\mathcal{O}(n)$. If there is a vertex $v \in V(G)$ such that $\chi(G - v) < \chi(G)$, we immediately know that G is not χ-vertex-stable. Otherwise, if $\chi(G - v) = \chi(G)$ for all $v \in V(G)$, it directly follows that G is χ-vertex-stable. Consequently, for every co-graph, we can decide whether it is χ-vertex-stable or not in time $\mathcal{O}(n^2 + m) \subseteq \mathcal{O}(n^2)$. □

With a similar proof as for the previous theorem, we obtain the next result.

Theorem 4.52 *For every co-graph, α-VERTEXSTABILITY can be solved in time $\mathcal{O}(n^2)$ and thus is in* P.

Proof Let G be a co-graph. According to Theorem 4.49, we can compute the graph's co-tree T in time $\mathcal{O}(n + m)$. Compute $\alpha(G)$ according to Remark 4.1 in time $\mathcal{O}(n)$. Now, for every $v \in V(G)$, we compute $\alpha(G - v)$ as previously described in time $\mathcal{O}(n)$. If there exists at least one vertex $v \in V(G)$ such that $\alpha(G - v) \neq \alpha(G)$, it follows immediately that G is not α-vertex-stable. Otherwise, G is α-vertex-stable. Hence, α-VERTEXSTABILITY can be solved in time $\mathcal{O}(n^2 + m) \subseteq \mathcal{O}(n^2)$ and thus is in P for co-graphs. □

We can use the same proof as for Theorem 4.52 to obtain the next corollary. However, this time we additionally use Lemma 2.1 to compute β from α for G and all induced subgraphs of G with one vertex removed.

Corollary 4.53 *For every co-graph, β-VERTEXSTABILITY can be solved in time $\mathcal{O}(n^2)$ and thus is in P.*

Now, Corollary 4.54 follows from a result due to Frei et al. [1, Proposition 34(5)] because α-VERTEXSTABILITY $= \{\overline{G} \mid G \in \omega$-VERTEXSTABILITY$\}$ and co-graphs are closed under complements.

Corollary 4.54 *For every co-graph, ω-VERTEXSTABILITY can be solved in time $\mathcal{O}(n^2)$ and thus is in P.*

Proof Let G be a co-graph. Then its complement $\overline{G}$ is a co-graph, too. Hence, we can exploit the fact that $\omega(G) = \alpha(\overline{G})$ and re-use the same idea as in the proof of Theorem 4.52 to decide whether G is ω-vertex-stable. ☐

Next, let us study the edge-related stability problems for co-graphs. To obtain our results, we need the following two auxiliary propositions.

Proposition 4.55 *Let G be a co-graph with $n > 1$ and let $u \in V(G)$ be χ-critical for G. There exist two co-graphs G_1, G_2 such that $G = G_1 \cup G_2$ or $G = G_1 + G_2$. Assuming, without loss of generality, that $u \in V(G_1)$, u is χ-critical for G_1.*

Proof If $G = G_1 \cup G_2$, we have $\chi(G) = \max\{\chi(G_1), \chi(G_2)\}$. Furthermore, if u is χ-critical for G, then $\chi(G - u) = \chi(G) - 1$. As we assume $u \in V(G_1)$, the removal of u from G only affects G_1, i.e., $G - u = (G_1 - u) \cup G_2$. Therefore, $\chi(G) = \chi(G_1) > \chi(G_2)$ must hold, as otherwise the removal of u would not affect $\chi(G)$. Consequently, $\chi(G_1 - u) = \chi(G_1) - 1$ is true, so u is χ-critical for G_1.

If $G = G_1 + G_2$, we have

$$\chi(G - u) = \chi(G) - 1$$
$$\Rightarrow \quad \chi((G_1 - u) + G_2) = \chi(G_1) + \chi(G_2) - 1$$
$$\Rightarrow \quad \chi(G_1 - u) + \chi(G_2) = \chi(G_1) + \chi(G_2) - 1$$
$$\Rightarrow \quad \chi(G_1 - u) = \chi(G_1) - 1,$$

so u is χ-critical for G_1. ☐

Proposition 4.56 *Let G be a co-graph and let $e = \{u, v\} \in E(G)$. If u and v are χ-critical for G, then e is χ-critical for G as well.*

Proof Let G be a co-graph and let $e = \{u, v\} \in E(G)$ be an edge with two χ-critical vertices $u, v \in V(G)$. First, we study the case that $G = G_1 + G_2$, where $u \in V(G_1)$ and $v \in V(G_2)$. Afterwards, we explain how to cover the other cases.

From Proposition 4.55 we know that u is χ-critical for G_1 and v is χ-critical for G_2. As noted by Frei et al. [1, Observation 4],[4] there exists an optimal coloring $c_1 : V(G_1) \to \mathbb{N}$ for G_1 such that for all $\tilde{u} \in V(G_1) \setminus \{u\}$ it holds that $c_1(\tilde{u}) \neq c_1(u)$. In other words, there is a coloring c_1 for G_1 such that u is the only vertex in G_1

---

[4] Which says that a vertex is critical in a graph exactly if there is an optimal coloring of the graph assigning one color to this vertex alone.

of its color. A similar optimal coloring c_2 must exist for G_2 with respect to v. For the combined graph with e removed, i.e., $G - e$, again as noted by Frei et al. [1, Observation 1],[5] it must hold that $\chi(G - e) \in \{\chi(G) - 1, \chi(G)\}$. Consequently, we can re-use c_1 and c_2 from G_1 and G_2, assuming distinct colors sets for c_1 and c_2, to obtain a legal coloring of G with $\chi(G)$ colors. However, we can color u with the same color as v, $c_2(v)$, and thus obtain a legal coloring for $G - e$ with $\chi(G) - 1$ colors. This is possible because

1. u is the only vertex in G_1 colored in $c_1(u)$ by definition of c_1,
2. no vertex $\tilde{u} \in V(G_1) \setminus \{u\}$ is colored with $c_2(v)$, as $c_1(V(G_1)) \cap c_2(V(G_2)) = \emptyset$, and
3. v is the only vertex in G_2 with this color, by definition of c_2, and there is no edge between u and v.

Consequently, after removing e from G, we can color $G - e$ with one color less than before such that $\chi(G - e) = \chi(G) - 1$ holds and e is χ-critical.

Initially, we assumed that $G = G_1 + G_2$ with $u \in V(G_1)$ and $v \in V(G_2)$ holds. If $G = G_1 \cup G_2$, there cannot exist any edge between vertices from G_1 and G_2. Hence, the only cases left are $G = G_1 + G_2$ or $G = G_1 \cup G_2$ with both vertices in either G_1 or G_2. Without loss of generality, assume that both vertices are in G_1. From Proposition 4.55 we know that both vertices are χ-critical for G_1, as they are χ-critical for G. When we can show that e is χ-critical for G_1, it immediately follows that e is also χ-critical for G. That is so because if $G = G_1 + G_2$ and e is χ-critical for G_1, we have $\chi(G_1 - e) = \chi(G_1) - 1$ such that

$$\chi(G - e) = \chi(G_1 - e) + \chi(G_2) = \chi(G_1) - 1 + \chi(G_2) = \chi(G) - 1,$$

so e is χ-critical for G in this case.

If $G = G_1 \cup G_2$, there is one more argument to add. We know that u and v are χ-critical for G and G_1. Consequently, $\chi(G_1) > \chi(G_2)$ must hold, as otherwise u or v cannot be χ-critical for G, since $\chi(G) = \max\{\chi(G_1), \chi(G_2)\}$. But then, it is enough to show that e is χ-critical for G_1, since reducing $\chi(G_1)$ by one via removing e also causes a reduction of $\chi(G)$ by one and, hence, e is χ-critical for G, too.

At some point, we must arrive in the case that one vertex is in G_1 and the other vertex is in G_2 and $G = G_1 + G_2$ holds, since the $+$ operation is the only possibility to add edges between vertices in co-graphs.   □

Having these results, we are now able to provide our stability-related results.

Theorem 4.57 *For co-graphs, χ-*STABILITY *can be solved in time $\mathcal{O}(nm)$ and thus is in* P.

Proof Let G be a co-graph. According to Theorem 4.49, we can compute the graph's co-tree T in time $\mathcal{O}(n + m)$. We can compute $\chi(G)$ according to Theorem 4.50 in time $\mathcal{O}(n)$ and, according to [1, Observation 1], for every edge $e \in E(G)$ and for

---

[5] Which says that deleting an edge or a vertex either decreases the chromatic number by exactly one or leaves it unchanged.

every vertex $v \in V(G)$, it holds that $\chi(G-e), \chi(G-v) \in \{\chi(G)-1, \chi(G)\}$. Thus, for every edge $e \in E(G)$, we proceed as follows to efficiently determine whether e is χ-critical or χ-stable in G: Let $u, v \in V(G)$ be the vertices with $e = \{u, v\}$. Then $G - u$ and $G - v$ are induced subgraphs of $G - e$, and $G - e$ is a subgraph of G. Again according to [1, Observation 1], it must hold that

$$\underbrace{\chi(G-v), \chi(G-u)}_{\in\{\chi(G)-1,\chi(G)\}} \leq \chi(G-e) \leq \chi(G).$$

Hence, if $\chi(G-v) = \chi(G)$ or $\chi(G-u) = \chi(G)$, which we can compute efficiently, it immediately follows that $\chi(G-e) = \chi(G)$. In other words, if u or v is χ-stable, we know that e must be χ-stable, too.[6]

On the other hand, if both u and v are χ-critical, it follows by Proposition 4.56 that e is χ-critical. Since, for every vertex in $V(G)$, we can determine whether it is χ-stable in time $\mathcal{O}(n)$, we can also determine for every edge in $E(G)$ whether it is χ-stable in time $\mathcal{O}(n)$. Consequently, we can decide in time $\mathcal{O}(n + m + nm) = \mathcal{O}(nm)$ whether G is χ-stable, and it follows that χ-STABILITY is in P for co-graphs. $\square$

Next, we study the problem ω-STABILITY for co-graphs. To do so, we need the following lemma.

Lemma 4.58 *Let G be a graph and let $v \in V(G)$ and $e \in E(G)$. Then $\omega(G-v)$ and $\omega(G-e)$ are in $\{\omega(G)-1, \omega(G)\}$.*

Proof First of all, it is obvious that by removing a vertex or an edge from G we cannot increase $\omega(G)$. Hence, $\omega(G-v) \leq \omega(G)$ and $\omega(G-e) \leq \omega(G)$. Furthermore, when we remove v from G, either v is part of all cliques in G of size $\omega(G)$ (so $\omega(G-v) = \omega(G) - 1$) or v is not part of all cliques in G of size $\omega(G)$ (so $\omega(G-v) = \omega(G)$). Generally speaking, by removing a vertex from G, we can either reduce a clique's size by one or leave it as it is. Now, let $e = \{u, v\}$ be an edge of G. Either e is between two vertices of a clique in G of size $\omega(G)$ or not. That is, by removing e we either reduce a largest clique's size by one or leave it unchanged. Therefore, for all $v \in V(G)$ and $e \in E(G)$, we have that $\omega(G - v)$ and $\omega(G - e)$ are in $\{\omega(G) - 1, \omega(G)\}$, as claimed. $\square$

Having these results, we can show the next theorem.

Theorem 4.59 *For co-graphs, ω-STABILITY can be solved in time $\mathcal{O}(n^2 m)$ and thus is in P.*

Proof Let G be a co-graph. According to Theorem 4.49, we can compute the graph's co-tree T in time $\mathcal{O}(n + m)$. By Theorem 4.50 we can compute ω for G and all induced subgraphs in time $\mathcal{O}(n)$. In order to decide whether G is ω-stable, we proceed as follows for every edge $e = \{u, v\}$:

   Case 1: $G = G_1 \cup G_2$ for two co-graphs G_1, G_2, and either $u, v \in V(G_1)$ or $u, v \in V(G_2)$ holds, since there are no edges between G_1 and G_2. Without loss

---

[6] Recall that this has been noted by Frei et al. [1, Observation 3].

of generality, assume that $u, v \in V(G_1)$. As $\omega(G) = \max\{\omega(G_1), \omega(G_2)\}$, we efficiently check whether $\omega(G_2) \geq \omega(G_1)$. In this case, we know that e cannot be ω-critical for G, because even if e were ω-critical for G_1, using Lemma 4.58, we would still have

$$\omega(G - e) = \max\{\omega(G_1 - e), \omega(G_2)\} = \max\{\omega(G_1) - 1, \omega(G_2)\} = \omega(G_2).$$

Otherwise, if $\omega(G_1) > \omega(G_2)$ holds, we study whether e is ω-critical for G_1 by recursively selecting the appropriate case, this time with G_1 as G. This is sufficient because if e is ω-critical for G_1, it is also ω-critical for G.

Case 2: $G = G_1 + G_2$ for two co-graphs G_1, G_2, and $u, v \in V(G_1)$ or $u, v \in V(G_2)$. In this case, it is sufficient to check whether e is ω-critical for the partial graph (i.e., G_1 or G_2) containing u and v. That is so because $\omega(G) = \omega(G_1) + \omega(G_2)$ and so, if e is ω-critical for one of the two partial graphs, e is also ω-critical for G. Once again, we check this by recursively applying the appropriate case for the corresponding partial graph.

Case 3: $G = G_1 + G_2$ for two co-graphs G_1, G_2, and u and v are in different partial graphs. Assume that $u \in V(G_1)$ and $v \in V(G_2)$. For any graph H, we denote by $W(H)$ the set of all maximum cliques in H, i.e., the set of all cliques of size $\omega(H)$. Now, in order for e to be ω-critical, for every clique $V' \in W(G_1)$, it must hold that $u \in V'$, and for every clique $V'' \in W(G_2)$, it must hold that $v \in V''$. For some fixed edge e, we can check both conditions in time $\mathcal{O}(n)$ by $\omega(G_1 - u) = \omega(G_1) - 1$ and $\omega(G_2 - v) = \omega(G_2) - 1$. If this is the case, then all other cliques in G_1 are strictly smaller than all $V' \in W(G_1)$, and all other cliques in G_2 are strictly smaller than all $V'' \in W(G_2)$. Hence, all cliques of size $\omega(G)$ in G are composed as $V' \cup V''$, with $V' \in W(G_1)$, $V'' \in W(G_2)$, $u \in V'$, and $v \in V''$. Removing $e = \{u, v\}$ from G causes $\omega(G)$ to be reduced by one, since every clique of size $\omega(G)$ contains e and afterwards it is missing the edge e in $G - e$. Therefore, only in this case e is ω-critical.

The number of recursive calls is limited by the height of the co-tree T, which is at most n. Every case can be handled in time $\mathcal{O}(n)$, since we only need to compute ω for a constant number of induced subgraphs. Thus, for a given co-graph G, we can determine in time $\mathcal{O}(n + m + mnn) = \mathcal{O}(n^2 m)$ whether G is ω-stable. Consequently, for all co-graphs, ω-STABILITY is in P. $\square$

As we now know that we can efficiently determine whether a given co-graph G is ω-stable, we can exploit the fact that co-graphs are closed under complements to obtain the following corollary.

Corollary 4.60 *For co-graphs, α-STABILITY can be solved in time $\mathcal{O}(n^4)$ and thus is in* P.

Proof Let G be a co-graph. Then $\overline{G}$ is a co-graph and $\alpha(G) = \omega(\overline{G})$. It follows immediately that G is α-stable if, and only if, $\overline{G}$ is ω-stable, which we can determine in time $\mathcal{O}(n^2 \overline{m})$) by Theorem 4.59. Hence, we can determine in time $\mathcal{O}(n^2 \overline{m}) = \mathcal{O}\left(n^2 \left(\frac{n(n-1)}{2} - m\right)\right) \subseteq \mathcal{O}(n^2 n^2) = \mathcal{O}(n^4)$ whether a given co-graph G is α-stable, so α-STABILITY is in P for co-graphs. $\square$

The next corollary follows immediately from Lemma 2.1 and [1, Proposition 34].

Corollary 4.61 *For co-graphs, β-STABILITY can be solved in time $\mathcal{O}(n^4)$ and thus is in* P.

At this point, we have completed our study of stability problems for co-graphs, as all open questions have been answered, and we turn to the unfrozenness problems for co-graphs. The next two corollaries exploit the fact that co-graphs are closed under complements and follow a similar argumentation.

Corollary 4.62 *For co-graphs, α-UNFROZENNESS and β-UNFROZENNESS can be solved in time $\mathcal{O}(n^4)$ and thus are in* P.

Proof Let G be a co-graph. Co-graphs are closed under complements, so $\overline{G}$ is a co-graph as well and we can compute $\overline{G}$ from G and a co-tree for $\overline{G}$ in time $\mathcal{O}(n + m)$. According to Theorem 4.59, we can check in time $\mathcal{O}(n^2\overline{m}) \subseteq \mathcal{O}(n^2n^2) = \mathcal{O}(n^4)$ whether $\overline{G}$ is ω-stable. Using [1, Proposition 34(2)], we immediately know that G is α-unfrozen and β-unfrozen if, and only if, $\overline{G}$ is ω-stable. Hence, both problems, α-UNFROZENNESS and β-UNFROZENNESS, can be solved in time $\mathcal{O}(n^4)$ and thus, for all co-graphs, belong to P. □

Corollary 4.63 *For co-graphs, ω-UNFROZENNESS can be solved in time $\mathcal{O}(n^4)$ and thus is in* P.

Proof Let G be a co-graph. Thus $\overline{G}$ is a co-graph, too, which we can compute efficiently, and from Corollary 4.60 we know that in time $\mathcal{O}(n^4)$ we can decide for $\overline{G}$ whether it is α-stable. Applying [1, Proposition 34(1)], we know that $\overline{G}$ is α-stable exactly if G is ω-unfrozen, which completes the proof. □

Finally, we answer the last remaining open question related to the unfrozenness of co-graphs.

Theorem 4.64 *For co-graphs, χ-UNFROZENNESS can be solved in time $\mathcal{O}(n^3)$ and thus is in* P.

Proof Let G be a co-graph and $e = \{u, v\} \in E(\overline{G})$ be a nonedge of G. Since G has at least two vertices, u and v, we have either $G = G_1 + G_2$ or $G = G_1 \cup G_2$ for two co-graphs G_1 and G_2. We handle both cases separately:

Case 1: If $G = G_1 + G_2$ is true, then e must belong either to $E(\overline{G_1})$ or to $E(\overline{G_2})$, since $\{\{u, v\} \mid u \in V(G_1), v \in V(G_2)\} \subseteq E(G)$ such that $E(\overline{G}) = E(\overline{G_1}) \cup E(\overline{G_2})$. Without loss of generality, assume that $e \in E(\overline{G_1})$. If e is χ-unfrozen for G_1, i.e., $\chi(G_1 + e) = \chi(G_1)$, then e is χ-unfrozen for G, since

$$\chi(G + e) = \chi(G_1 + e) + \chi(G_2) = \chi(G_1) + \chi(G_2) = \chi(G).$$

Conversely, if e is χ-frozen for G_1, i.e., $\chi(G_1 + e) = \chi(G_1) + 1$, then e is χ-frozen for G as well, as

$$\chi(G + e) = \chi(G_1 + e) + \chi(G_2) = \chi(G_1) + 1 + \chi(G_2) = \chi(G) + 1.$$

Hence, it is enough to determine whether e is χ-unfrozen or χ-frozen for G_1 and we can follow the argumentation of this proof recursively for G_1.

Case 2: If $G = G_1 \cup G_2$ is true, e can belong to $E(\overline{G_1}) \cup E(\overline{G_2})$ or $E(\overline{G}) \setminus (E(\overline{G_1}) \cup E(\overline{G_2}))$. Consider the following two subcases:

Case 2.a: If $e \in E(\overline{G_1})$ or $e \in E(\overline{G_2})$, we proceed as follows. Without loss of generality, assume $e \in E(\overline{G_1})$. Since $\chi(G) = \max\{\chi(G_1), \chi(G_2)\}$, an increase of $\chi(G_1)$ affects $\chi(G)$ only if $\chi(G_1) \geq \chi(G_2)$. Otherwise, e is χ-unfrozen for G (but not necessarily for G_1). If $\chi(G_1) \geq \chi(G_2)$, then if e is χ-unfrozen for G_1, e is χ-unfrozen for G, since

$$\chi(G + e) = \max\{\chi(G_1 + e), \chi(G_2)\} = \chi(G_1 + e) = \chi(G_1) = \chi(G).$$

Similarly, if e is χ-frozen for G_1, e is χ-frozen for G, since

$$\chi(G+e) = \max\{\chi(G_1+e), \chi(G_2)\} = \chi(G_1+e) = \chi(G_1)+1 = \chi(G)+1.$$

Consequently, it is enough to determine whether e is χ-unfrozen or χ-frozen for G_1 and we can follow the argumentation of this proof recursively for G_1.

Case 2.b: If $e \in E(\overline{G}) \setminus (E(\overline{G_1}) \cup E(\overline{G_2}))$, then $u \in V(G_1)$ and $v \in V(G_2)$. Now, if $\chi(G_1) = \chi(G_2) = 1$, e is χ-frozen for G, since

$$\chi(G + e) = 1 + 1 = 2 > 1 = \max\{\chi(G_1), \chi(G_2)\} = \chi(G).$$

Conversely, if $\chi(G_1) > 1$ or $\chi(G_2) > 1$, e is χ-unfrozen for G since G_1 and G_2 share no edge but e. Because of that we can arrange the colors for $V(G_1)$ and $V(G_2)$ in such a way that both vertices incident to e have different colors, resulting in $\chi(G + e) = \chi(G)$.

According to Theorem 4.49, we can compute the graph's co-tree T in time $\mathcal{O}(n+m)$. Following these cases, for each of the at most n^2 nonedges $e \in E(\overline{G})$, we can efficiently (namely, in time $\mathcal{O}(n)$) determine whether it is χ-frozen or χ-unfrozen for G, so χ-UNFROZENNESS can be solved in time $\mathcal{O}(n^3)$ and thus, for all co-graphs, belongs to P. □

5 Conclusion

We have studied the stability, vertex stability, and unfrozenness problems introduced by Frei et al. [1] when restricted to special graph classes. In particular, we have studied these three problems for six important graph classes and four central graph parameters: α, β, ω, and χ. While Frei et al. [1] provided hardness results for their problems for general graphs, we showed that restricting to special graph classes allows for efficient solvability. Thus our work provides some baseline for further, more expanding work along this line of research. In particular, these results can be useful for real-world

Table 1 Overview of the running times of algorithms designed for stability (E), vertex stability (V), and unfrozenness with respect to the four graph parameters studied here for graphs G of bounded tree-width and clique-width, where $m = |E(G)|$ and $n = |V(G)|$ and k is the tree-width or clique-width

			Bounded tree-width: $\mathcal{TW}_k$	Bounded clique-width: $\mathcal{CW}_k$
α	Stable	E	$\mathcal{O}(2^k mn)$ C. 4.11	$\mathcal{O}(2^{2k} mn)$ T. 4.24
		V	$\mathcal{O}(2^k n^2)$ C. 4.12	$\mathcal{O}(2^{2k} n^2)$ C. 4.25
	Unfrozen		$\mathcal{O}(2^k n^3)$ T. 4.13	$\mathcal{O}(2^{2k} n^3)$ T. 4.26
β	Stable	E	$\mathcal{O}(2^k mn)$ C. 4.11	$\mathcal{O}(2^{2k} mn)$ T. 4.24
		V	$\mathcal{O}(2^k n^2)$ C. 4.12	$\mathcal{O}(2^{2k} n^2)$ C. 4.25
	Unfrozen		$\mathcal{O}(2^k n^3)$ T. 4.13	$\mathcal{O}(2^{2k} n^3)$ T. 4.26
ω	Stable	E	$\mathcal{O}(2^k mn)$ C. 4.11	$\mathcal{O}(2^{2k} mn)$ T. 4.24
		V	$\mathcal{O}(2^k n^2)$ C. 4.12	$\mathcal{O}(2^{2k} n^2)$ C. 4.25
	Unfrozen		$\mathcal{O}(2^k n^3)$ T. 4.13	$\mathcal{O}(2^{2k} n^3)$ T. 4.26
χ	Stable	E	$\mathcal{O}(k^{\mathcal{O}(k)} mn)$ C. 4.11	$\mathcal{O}(2^{3k} k^2 mn 2^{2k+5}+1)$ T. 4.24
		V	$\mathcal{O}(k^{\mathcal{O}(k)} n^2)$ C. 4.12	$\mathcal{O}(2^{3k} k^2 n^{2^{2k+1}+2})$ C. 4.25
	Unfrozen		$\mathcal{O}(k^{\mathcal{O}(k)} n^3)$ T. 4.13	$\mathcal{O}(2^{3k} k^2 n^{2^{2k+5}+3})$ T. 4.26

Behind each running time we state the number of the theorem (T.) or corollary (C.) establishing this result

applications when knowledge about the stability, vertex stability, or unfrozenness of a graph with respect to a certain graph parameter is required and graphs with such a special structure may typically occur in this application.

Table 2 Overview of the running times of algorithms designed for stability (E), vertex stability (V), and unfrozenness with respect to the four graph parameters studied here for some special graph classes, where $m = |E(G)|$ and $n = |V(G)|$

			Trees: $\mathcal{T}$	Forests: $\mathcal{F}$	Bipartite graphs: $\mathcal{B}$	Co-graphs: $\mathcal{C}$
α	Stable	E	$\mathcal{O}(n^2)$ C. 4.42	$\mathcal{O}(n^2)$ T. 4.40	$\mathcal{O}(n^{2.5} m)$ C. 4.29	$\mathcal{O}(n^4)$ C. 4.60
		V	$\mathcal{O}(n^2)$ C. 4.42	$\mathcal{O}(n^2)$ T. 4.40	$\mathcal{O}(n^{3.5})$ C. 4.29	$\mathcal{O}(n^2)$ T. 4.52
	Unfrozen		$\mathcal{O}(n^3)$ C. 4.47	$\mathcal{O}(n^3)$ C. 4.47	$\mathcal{O}(n^{4.5})$ C. 4.39	$\mathcal{O}(n^4)$ C. 4.62
β	Stable	E	$\mathcal{O}(n^2)$ C. 4.42	$\mathcal{O}(n^2)$ T. 4.40	$\mathcal{O}(n^{2.5} m)$ C. 4.29	$\mathcal{O}(n^4)$ C. 4.61
		V	$\mathcal{O}(n^2)$ C. 4.42	$\mathcal{O}(n^2)$ T. 4.40	$\mathcal{O}(n^{3.5})$ C. 4.29	$\mathcal{O}(n^2)$ C. 4.53
	Unfrozen		$\mathcal{O}(n^3)$ C. 4.47	$\mathcal{O}(n^3)$ C. 4.47	$\mathcal{O}(n^{4.5})$ T. 4.38	$\mathcal{O}(n^4)$ C. 4.62
ω	Stable	E	$\mathcal{O}(1)$ C. 4.43	$\mathcal{O}(1)$ C. 4.41	$\mathcal{O}(1)$ P. 4.31	$\mathcal{O}(n^2 m)$ T. 4.59
		V	$\mathcal{O}(n)$ C. 4.43	$\mathcal{O}(n)$ C. 4.41	$\mathcal{O}(n)$ T. 4.32	$\mathcal{O}(n^2)$ C. 4.54
	Unfrozen		$\mathcal{O}(1)$ P. 4.44	$\mathcal{O}(n)$ T. 4.46	$\mathcal{O}(n)$ C. 4.36	$\mathcal{O}(n^4)$ C. 4.63
χ	Stable	E	$\mathcal{O}(1)$ C. 4.43	$\mathcal{O}(1)$ C. 4.41	$\mathcal{O}(1)$ P. 4.33	$\mathcal{O}(nm)$ T. 4.57
		V	$\mathcal{O}(n)$ C. 4.43	$\mathcal{O}(n)$ C. 4.41	$\mathcal{O}(n)$ T. 4.34	$\mathcal{O}(n^2)$ T. 4.51
	Unfrozen		$\mathcal{O}(1)$ P. 4.44	$\mathcal{O}(n)$ T. 4.46	$\mathcal{O}(n)$ T. 4.35	$\mathcal{O}(n^3)$ T. 4.64

Behind each running time we state the number of the theorem (T.), corollary (C.), or proposition (P.) establishing this result

Table 1 summarizes our results for graphs of bounded tree-width and bounded clique-width. While this already shows that the stability, vertex stability, and unfrozenness problems are solvable in polynomial time for some of the other special graph classes we study (namely, for trees, forests, and co-graphs), we have also investigated these problems for these three classes of graphs as well as for the class of bipartite graphs in detail to obtain better running times. These results are summarized in Table 2.

For future work, we propose to study further special graph classes that are not covered here. Besides the study of stability for other graph classes, one can also study the concept of *cost of stability*:[7] Given a graph, the question is how costly it is to stabilize it. In other words, what is the smallest number of vertices or edges to be added to or removed from the graph such that the resulting graph is stable or unfrozen with respect to some graph parameter. Relatedly, it would make sense to combine these two approaches and study the cost of stability for special graph classes.

Declarations

References

1. Frei, F., Hemaspaandra, E., Rothe, J.: Complexity of stability. J. Comput. Syst. Sci. **123**, 103–121 (2022)
2. Papadimitriou, C., Zachos, S.:Two remarks on the power of counting.In: Proceedings of the 6th GI conference on theoretical computer science, pp. 269–276. Springer, Heidelberg and Berlin, Germany (1983)
3. Wagner, K.: More complicated questions about maxima and minima, and some closures of NP. Theoret. Comput. Sci. **51**(1–2), 53–80 (1987)
4. Wagner, K.: Bounded query classes. SIAM J. Comput. **19**(5), 833–846 (1990)
5. Gottlob, G.: NP trees and Carnap's modal logic. J. ACM **42**(2), 421–457 (1995)

---

[7] Bachrach et al. [50] study a related notion of "cost of stability" for cooperative games.

6. Eiter, T., Lukasiewicz, T.: Default reasoning from conditional knowledge bases: complexity and tractable cases. Artif. Intell. **124**(2), 169–241 (2000)
7. Lukasiewicz, T., Malizia, E.: A novel characterization of the complexity class Θ_k^p based on counting and comparison. Theoret. Comput. Sci. **694**, 21–33 (2017)
8. Hemaspaandra, E., Hemaspaandra, L., Rothe, J.: Exact analysis of Dodgson elections: Lewis Carroll's 1876 voting system is complete for parallel access to NP. J. ACM **44**(6), 806–825 (1997)
9. Hemaspaandra, E., Spakowski, H., Vogel, J.: The complexity of Kemeny elections. Theoret. Comput. Sci. **349**(3), 382–391 (2005)
10. Hemaspaandra, E., Wechsung, G.: The minimization problem for boolean formulas. SIAM J. Comput. **31**(6), 1948–1958 (2002)
11. Hemaspaandra, E., Rothe, J.: Recognizing when greed can approximate maximum independent sets is complete for parallel access to NP. Inf. Process. Lett. **65**(3), 151–156 (1998)
12. Hemaspaandra, E., Rothe, J., Spakowski, H.: Recognizing when heuristics can approximate minimum vertex covers is complete for parallel access to NP. R.A.I.R.O. Theoretical Informatics and Applications **40**(1), 75–91 (2006)
13. Rothe, J., Spakowski, H., Vogel, J.: Exact complexity of the winner problem for Young elections. Theory of Computing Systems **36**(4), 375–386 (2003)
14. Hemachandra, L.: The strong exponential hierarchy collapses. J. Comput. Syst. Sci. **39**(3), 299–322 (1989)
15. Hemaspaandra, E., Hemaspaandra, L., Rothe, J.: Raising NP lower bounds to parallel NP lower bounds. SIGACT News **28**(2), 2–13 (1997)
16. Papadimitriou, C., Yannakakis, M.: The complexity of facets (and some facets of complexity). J. Comput. Syst. Sci. **28**(2), 244–259 (1984)
17. Rothe, J.: Complexity theory and cryptology. An introduction to cryptocomplexity. EATCS Texts in Theoretical Computer Science. Springer, Heidelberg and Berlin, Germany (2005)
18. Weishaupt, R., Rothe, J.: Stability of special graph classes. In: Proceedings of the 22nd Italian conference on theoretical computer science, vol. 3072, pp. 20–12015. CEUR-WS.org, RWTH Aachen, Germany (2021)
19. Berge, C.: Perfect graphs. In: Six papers on graph theory, pp. 1–21. Indian Statistical Institute, Calcutta, India (1963)
20. Golumbic, M.: Algorithmic graph theory and perfect graphs. Academic Press, Cambridge, Massachusetts, USA (1980)
21. Papadimitriou, C.: Computational complexity, 2nd edn. Addison-Wesley, Boston, Massachusetts, USA (1995)
22. Gallai, T.: Über extreme Punkt- und Kantenmengen. Annales Universitatis Scientiarum Budapestinensis de Rolando Eötvös Nominatae. Sectio mathematica **2**, 133–138 (1959)
23. Weishaupt, R., Rothe, J.: Stability of special graph classes. Technical Report arXiv:2106.01496 [cs.CC], arXiv.org (2021)
24. Robertson, N., Seymour, P.D.: Graph minors II. Algorithmic aspects of tree width. Journal of Algorithms **7**, 309–322 (1986)
25. Bodlaender, H.L.: A partial k-arboretum of graphs with bounded treewidth. Theoret. Comput. Sci. **209**, 1–45 (1998)
26. Arnborg, S., Corneil, D.G., Proskurowski, A.: Complexity of finding embeddings in a k-tree. SIAM Journal of Algebraic and Discrete Methods **8**(2), 277–284 (1987)
27. Bodlaender, H.L.: A linear-time algorithm for finding tree-decompositions of small treewidth. SIAM J. Comput. **25**(6), 1305–1317 (1996)
28. Hliněný, P., Oum, S., Seese, D., Gottlob, G.: Width parameters beyond tree-width and their applications. Computer Journal **51**(3), 326–362 (2008)
29. Cygan, M., Fomin, F.V., Kowalik, L., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., Saurabh, S.: Parameterized algorithms. Springer, New York, USA (2015)
30. Courcelle, B., Olariu, S.: Upper bounds to the clique width of graphs. Discret. Appl. Math. **101**, 77–114 (2000)
31. Espelage, W., Gurski, F., Wanke, E.: How to solve NP-hard graph problems on clique-width bounded graphs in polynomial time. In: Proceedings of the 27th International workshop on graph-theoretic concepts in computer science. LNCS, vol. 2204, pp. 117–128. Springer, Heidelberg and Berlin, Germany (2001)

32. Gurski, F., Wanke, E.: Vertex disjoint paths on clique-width bounded graphs (Extended abstract). In: Proceedings of the 13th Latin American symposium on theoretical informatics. LNCS, vol. 2976, pp. 119–128. Springer, Heidelberg and Berlin, Germany (2004)

33. Kobler, D., Rotics, U.: Edge dominating set and colorings on graphs with fixed clique-width. Discret. Appl. Math. **126**(2–3), 197–221 (2003)

34. Corneil, D.G., Rotics, U.: On the relationship between clique-width and treewidth. SIAM J. Comput. **4**, 825–847 (2005)

35. Gurski, F.: The behavior of clique-width under graph operations and graph transformations. Theory of Computing Systems **60**(2), 346–376 (2017)

36. Fellows, M.R., Rosamond, F.A., Rotics, U., Szeider, S.: Clique-width is NP-complete. SIAM J. Discret. Math. **23**(2), 909–939 (2009)

37. Golumbic, M.C., Rotics, U.: On the clique-width of some perfect graph classes. In: Proceedings of the 25th International workshop on graph-theoretic concepts in computer science. LNCS, vol. 1665, pp. 135–147. Springer, Heidelberg and Berlin, Germany (1999)

38. Courcelle, B., Makowsky, J.A., Rotics, U.: Linear time solvable optimization problems on graphs of bounded clique-width. Theory of Computing Systems **33**(2), 125–150 (2000)

39. Oum, S., Seymour, P.D.: Approximating clique-width and branch-width. Journal of Combinatorial Theory, Series B **96**(4), 514–528 (2006)

40. Oum, S.: Approximating rank-width and clique-width quickly. ACM Trans. Algorithms **5**(1), 1–20 (2008)

41. Gurski, F.: A comparison of two approaches for polynomial-time algorithms computing basic graph parameters. Technical Report arXiv:0806.4073 [cs.DS], arXiv.org (2008)

42. Hopcroft, J., Karp, R.: An $n^{5/2}$ algorithm for maximum matching in bipartite graphs. SIAM J. Comput. **2**, 225–231 (1973)

43. Kőnig, D.: Gráfok és mátrixok. Matematikai és Fizikai Lapok **38**, 116–119 (1931)

44. Cormen, T., Leiserson, C., Rivest, R., Stein, C.: Introduction to algorithms, 4th edn. MIT Press, Cambridge, Massachusetts, USA (2022)

45. Alt, H., Blum, N., Mehlhorn, K., Paul, M.: Computing a maximum cardinality matching in a bipartite graph in time $O(n^{1.5}m\log n)$. Inf. Process. Lett. **37**(4), 237–240 (1991)

46. Bollobás, B.: Modern graph theory. Graduate Texts in Mathematics, vol. 184. Springer, Heidelberg and Berlin, Germany (1998)

47. Kleinberg, J., Tardos, É.: Algorithm design. Addison-Wesley Publishing Company, Boston, Massachusetts, USA (2004)

48. Corneil, D., Lerchs, H., Burlingham, L.S.: Complement reducible graphs. Discret. Appl. Math. **3**(3), 163–174 (1981)

49. Corneil, D., Perl, Y., Stewart, L.: A linear recognition algorithm for cographs. SIAM J. Comput. **14**(4), 926–934 (1985)

50. Bachrach, Y., Elkind, E., Malizia, E., Meir, R., Pasechnik, D., Rosenschein, J., Rothe, J., Zuckerman, M.: Bounds on the cost of stabilizing a cooperative game. Journal of Artificial Intelligence Research **63**, 987–1023 (2018)