Journal of Computer and System Sciences 123 (2022) 103–121

Contents lists available at ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

Complexity of stability ☆

Fabian Frei [a,*], Edith Hemaspaandra [b,1], Jörg Rothe [c,2]

[a] Department of Computer Science, ETH Zürich, Universitätstrasse 6, 8092 Zürich, Switzerland
[b] Department of Computer Science, Rochester Institute of Technology, Rochester, NY 14623, USA
[c] Institut für Informatik, Heinrich-Heine-Universität Düsseldorf, 40225 Düsseldorf, Germany

A R T I C L E I N F O

Article history:
Received 16 January 2021
Received in revised form 4 June 2021
Accepted 12 July 2021
Available online 28 July 2021

Keywords:
Stability
Colorability
Vertex cover
Satisfiability
Difference polynomial time
Parallel access to NP

A B S T R A C T

Graph parameters such as the clique number and the chromatic number are central in many areas, ranging from computer networks to linguistics to computational neuroscience to social networks. In particular, the chromatic number of a graph can be applied in solving practical tasks as diverse as pattern matching, scheduling jobs to machines, allocating registers in compiler optimization, and even solving Sudoku puzzles. Typically, however, the underlying graphs are subject to (often minor) changes. To make these applications of graph parameters robust, it is important to know which graphs are stable in the sense that adding or deleting single edges or vertices does not change them. We initiate the study of stability of graphs in terms of their computational complexity. We show for various central graph parameters that deciding the stability of a given graph is complete for Θ_2^{p}, a well-known complexity class in the second level of the polynomial hierarchy.

1. Introduction

In this first section, we motivate our research topic, introduce the necessary notions and notation, and provide an overview of both the related work and our contribution.

1.1. Motivation

Informally stated, a graph is *stable with respect to some graph parameter* (such as the chromatic number) if some type of small perturbation of the graph (a local modification such as adding an edge or deleting a vertex) does not change the parameter. Other graph parameters we consider are the clique number, the independence number, and the vertex cover number. This notion of stability formalizes the robustness of graphs for these parameters, which is important in many applications. Typical applications of the chromatic number, for instance, include coloring algorithms for complex networks such as social, economic, biological, and information networks (see, e.g., Jackson's book on social and economic networks [28] or Khor's work on applying graph coloring to biological networks [30]). In particular, social networks can

☆ A preliminary version of parts of this paper appeared in the Proceedings of the 31st International Symposium on Algorithms and Computation (ISAAC) [16].

* Corresponding author.
E-mail addresses: fabian.frei@inf.ethz.ch (F. Frei), eh@cs.rit.edu (E. Hemaspaandra), rothe@hhu.de (J. Rothe).

1 Research done in part while on sabbatical at Heinrich-Heine-Universität Düsseldorf and supported in part by NSF grant DUE-1819546 and a Renewed Research Stay grant from the Alexander von Humboldt Foundation.
2 Research supported by DFG grants RO 1202/14-2 and RO 1202/21-1.

https://doi.org/10.1016/j.jcss.2021.07.001
0022-0000/

be colored to find roles [15] or to study human behavior in small controlled groups [29,10]. In various applied areas of computer science, graph coloring has also been used for register allocation in compiler optimization [7], pattern matching and pattern mining [39], and scheduling tasks [31]. To ensure that these applications of graph parameters are robust, graphs need to be stable for them with respect to certain operations. Recognizing instability in advance is desirable as it affords us the opportunity to amend the situation or at least take precautions for the case of a sudden change. We initiate a systematic study of stability of graphs in terms of their computational complexity and present some tools to stabilize specific parts of a graph.

1.2. Notions and notation

In this subsection, we define the core notions used in this paper and fix our notation.

1.2.1. Complexity classes

We begin with the relevant complexity classes. Besides P, NP, and coNP, these are DP, coDP, and Θ_2^{p}. The class DP, introduced by Papadimitriou and Yannakakis [35], is the second level of the Boolean hierarchy over NP; that is,

$$\mathrm{DP} = \mathrm{NP} \wedge \mathrm{coNP} = \{L_1 \cap L_2 \mid L_1 \in \mathrm{NP} \wedge L_2 \in \mathrm{coNP}\}$$

is the set of all intersections of NP languages with coNP languages. Equivalently, it can be seen as the *differences* of NP languages, whence the name. An example of a trivially DP-complete language is Sat-UnSat = Sat × UnSat, where UnSat is the set of all unsatisfiable CNF-formulas. The complement class coDP contains exactly the unions of NP languages with coNP languages.

The class Θ_2^{p}, whose name is due to Wagner [41], belongs to the second level of the polynomial hierarchy; it can be defined as $\Theta_2^{\mathrm{p}} = \mathrm{P}^{\mathrm{NP}[\mathcal{O}(\log n)]}$, which is the class of problems that can be solved in polynomial time by an algorithm with access to an oracle that decides arbitrary instances for an NP-complete problem—with one instance per call and each such query taking constant time—restricted to a logarithmic number of queries. (Without the last restriction, we would get the class $\Delta_2^{\mathrm{p}} = \mathrm{P}^{\mathrm{NP}}$.) Results due to Hemachandra [23, Theorem 4.10] usefully characterize Θ_2^{p} as $\mathrm{P}_{\mathrm{tt}}^{\mathrm{p}}$, the class of languages that are polynomial-time truth-table reducible to NP. By definition, this is the same as $\mathrm{P}_{\|}^{\mathrm{NP}}$, the class of languages that are polynomial-time recognizable with *unlimited parallel* access to an NP oracle. *Unlimited* means that an algorithm witnessing the membership of a problem in $\mathrm{P}_{\|}^{\mathrm{NP}}$ can query the oracle on as many instances of an NP-complete problem as it wants—which due the polynomial running-time means at most polynomially many—while *parallel* means that all queries need to be sent simultaneously. The characterization of Θ_2^{p} as $\mathrm{P}^{\mathrm{NP}[\mathcal{O}(\log n)]}$, in contrast, allows the logarithmically many queries to be *adaptive*; that is, they can be sent interactively, with one depending on the oracle's answers to the previous ones. Membership proofs for Θ_2^{p} are usually easy; we will see a simple example of how to give one at the beginning of Section 3.

Note that the definitions immediately yield the inclusions

$$\mathrm{NP} \cup \mathrm{coNP} \subseteq \mathrm{DP} \subseteq \Theta_2^{\mathrm{p}} \subseteq \Delta_2^{\mathrm{p}}.$$

1.2.2. Graphs and graph numbers

Throughout this paper graphs are simple. Let $\mathcal{G}$ be the set of all (simple) graphs and $\mathbb{N}$ the set of natural numbers including zero. For any set M, we denote its *cardinality* or *size* by $\|M\|$. A map $\xi : \mathcal{G} \to \mathbb{N}$ is called a *graph number*. In this paper, we examine the prominent graph numbers α, β, χ, and ω, which give the size of a maximum independent set, the size of a minimum vertex cover, the size of a minimum coloring (i.e., the minimum number of colors allowing for a proper vertex coloring), and the size a maximum clique, respectively.

Let V, E, and $\overline{E}$ be the functions that map a graph G to its vertex set $V(G)$, its edge set $E(G)$, and its set of *nonedges* $\overline{E}(G) = \{\{u, v\} \mid u, v \in V(G) \wedge u \neq v\} - E(G)$, respectively.

Let G and H be graphs. We denote by $G \cup H$ the disjoint union and by $G + H$ the *join*, which is $G \cup H$ with all *join edges*—i.e., the edges $\{v, w\} \in V(G) \times V(H)$—added to it.[3]

For $v \in V(G)$, $e \in E(G)$, and $e' \in \overline{E}(G)$, we denote by $G - v$, $G - e$, and $G + e'$ the graphs that result from G by deleting v, deleting e, and adding e', respectively.

For any $k \in \mathbb{N}$, we denote by I_k and K_k the empty (i.e., edgeless) and the complete graph on k vertices, respectively. The graph $I_0 = K_0$ without any vertices is called the *null graph*. A vertex v is *universal* with respect to a graph G if it is adjacent to all vertices $V(G) - \{v\}$.

1.2.3. Stability

Let G be a graph. An edge $e \in E(G)$ is called *stable* with respect to a graph number ξ (or ξ*-stable*, for short) if $\xi(G) = \xi(G - e)$, that is, deleting e leaves ξ unchanged. Otherwise (that is, if the deletion of e does change ξ), e is called ξ*-critical*. For a vertex $v \in V(G)$ instead of an edge $e \in E(G)$, stability and criticality are defined in the same way.

[3] We adopt the notation $G + H$ for the join from Harary's classical textbook on graph theory [20, p. 21].

A graph is called ξ-stable if all of its edges are ξ-stable. A graph whose vertices—rather than edges – are all ξ-stable is called ξ-vertex-stable. Analogously, a graph is called ξ-critical and ξ-vertex-critical if all its edges and vertices, respectively, are ξ-critical. Note that each edge and vertex is either stable or critical, whereas a graph might be neither. An unspecified ξ defaults to the chromatic number χ.

A traditional term for stability with respect to adding edges and vertices—rather than deleting them—is unfrozenness.[4] Specifically, a nonedge $e \in \overline{E}(G)$ is called *unfrozen* if adding it to the graph G leaves χ unchanged, and *frozen* otherwise. All of these notions extend naturally to vertices (where we can freely choose to which existing vertices a new vertex is adjacent, implying an exponential number of possibilities), to entire graphs, and to any graph number ξ, as just seen for stability and criticality.

We call a graph *two-way stable* if it is both stable and unfrozen. Again, this notion is understood with respect to the chromatic number and modifying edges by default; namely, a graph is two-way stable if neither deleting nor adding a single edge changes the chromatic number. As before, we have the analogous set of notions with respect to vertices and any graph number ξ.

Prefixing a natural number $k \in \mathbb{N}$ to any of these notions additionally requires the respective graph number to be exactly k. For example, a graph G is *k-critical* if and only if $\chi(G) = k$ and $\chi(G - e) \neq k$ for every $e \in E(G)$.

The notion of stability can be naturally applied to Boolean formulas as well. We call a formula Φ in conjunctive normal form *stable* if deleting an arbitrary clause C does not change its satisfiability status—that is, if it either is satisfiable (and of course stays so upon deletion of a clause) or if it and all its 1-clause-deleted subformulas $\Phi - C$ are unsatisfiable. We remark that the unfrozen formulas are exactly the unsatisfiable ones since adding an empty clause renders any formula unsatisfiable.

1.2.4. Stability problems

We denote by CNF the set of formulas in conjunctive normal form and by 3CNF, 4CNF, and 6CNF the set of CNF-formulas with *exactly* 3, 4, and 6 literals per clause, respectively.[5] The sets SAT and 3SAT contain the satisfiable, UNSAT and 3UNSAT the unsatisfiable formulas from CNF and 3CNF, respectively. Let $\text{StableUnSat} = \{\Phi \in \text{UnSat} \mid (\Phi - C) \in \text{UnSat}$ for every clause C of $\Phi\}$ be the set of stably unsatisfiable formulas. The set $\text{StableCNF} = \text{Sat} \cup \text{StableUnSat}$ consists of the stable CNF-formulas. Intersecting with 3CNF yields the classes STABLE3UNSAT and STABLE3CNF and so on.

Let STABILITY be the set of stable graphs and UNFROZENNESS the set of unfrozen graphs, both with respect to the default graph number χ. The set of two-way stable graphs is $\text{TwoWayStability} = \text{Stability} \cap \text{Unfrozenness}$. Once more, these definitions extend naturally. For example, 4-VERTEXSTABILITY is the set of (with respect to the default χ) 4-vertex-stable graphs and β-TWOWAYSTABILITY consists of the graphs for which the vertex-cover number β remains unchanged upon deletion or addition of an edge.

1.2.5. AND *functions and* OR *functions*

Following Chang and Kadin [9], we say that a language $L \subseteq \Sigma^*$ has AND_2 if there is a polynomial-time computable function $f : \Sigma^* \times \Sigma^* \to \Sigma^*$ such that for all $x_1, x_2 \in \Sigma^*$, we have

$$x_1 \in L \wedge x_2 \in L \iff f(x_1, x_2) \in L.$$

If this is the case, we call f an AND_2 function for L. If there even is a polynomial-time computable function $f : \bigcup_{k=0}^{\infty}(\Sigma^*)^k \to \Sigma^*$ such that for every $k \in \mathbb{N}$ and for all $x_1, \ldots, x_k \in \Sigma^*$ we have

$$x_1 \in L \wedge \cdots \wedge x_k \in L \iff f(x_1, \ldots, x_k) \in L,$$

then we say that L has AND_ω. Replacing $\wedge$ with $\vee$, we get the analogous notions OR_2 and OR_ω. Note that a language has an AND_2 function if and only if its complement has an OR_2 function, with the analogous statement holding for AND_ω and OR_ω.

1.3. Related work

Many interesting problems are suspected to be complete for either DP or Θ_2^{p}. While membership is usually trivial in these cases, matching lower bounds are rare and hard to prove. For example, Woeginger [44] observes that determining

[4] The notion of instance parts being either frozen or unfrozen has originally been introduced to the field of computational complexity in analogy to the physical process of freezing [32,33].

The sudden shift from P to NP-hardness that can be observed when transitioning from 2SAT to 3SAT by allowing a larger and larger percentage of clauses of length 3 rather than 2, for example, mimics the phase transition from liquid to solid, with the former granting much higher degrees of freedom to the substance's constituents than the latter. Based on this general intuition, Beacham and Culberson [2] then more formally defined the notion of unfrozenness with regard to an arbitrary graph property that is downward monotone (meaning that a graph keeps the property when edges are deleted); they call a graph unfrozen if it also keeps the property when an arbitrary new edge is added. We naturally extend this notion to arbitrary graph numbers, which are not necessarily monotone.

[5] In the literature, these set names are often prefixed by an E, emphasizing the exactness. This is notably not the case for a paper by Cai and Meyer [6] that contains a construction crucially relying on this restriction. We will build upon this construction later on and are thus bound to the same constraint.

whether a graph has a wonderfully stable partition is in Θ_2^{p}, and leaves it as an open problem to settle the exact complexity. Wagner, who introduced the class name Θ_2^{p} [40], provided a number of hardness results for variants of standard problems such as Satisfiability, Clique and Colorability, which are designed to be complete for DP or Θ_2^{p}. For example, he proves the DP-completeness of

$$\textsc{ExactClique} = \{(G, k) \in \mathcal{G} \times \mathbb{N} \mid \omega(G) = k\} \quad [40, \text{Theorem 6.1.1}]$$

and the Θ_2^{p}-completeness[6] of

$$\textsc{OddClique} = \{G \in \mathcal{G} \mid \omega(G) \text{ is odd}\} \quad [40, \text{Theorem 6.1.2}].$$

He obtains the analogous results for Colorability [40, Theorem 6.3], Vertex Cover [40, Corollary 6.4], and Independent Set [40, Corollary 6.4] instead of Clique and points out [40, second-to-last paragraph] that his proof techniques also yield the Θ_2^{p}-completeness of the equality version of all of these problems—for example,

$$\textsc{EqualClique} = \{(G, H) \in \mathcal{G}^2 \mid \omega(G) = \omega(H)\}.$$

The same holds true for the comparison versions[7] such as

$$\textsc{CompareClique} = \{(G, H) \in \mathcal{G}^2 \mid \omega(G) \leq \omega(H)\}.$$

The DP-completeness of ExactColorability has been extended to the subproblem of recognizing graphs with chromatic number 4 [36]. Furthermore, a few election problems have been proved to be Θ_2^{p}-complete by Hemaspaandra et al. [24,25], by Rothe et al. [37], and Hemaspaandra et al. [26]. Weishaupt and Rothe [42] provide a systematic study of graph classes (including trees, bipartite graphs, and co-graphs) for which various stability problems become polynomial-time solvable.

In general, establishing lower bounds proved to be difficult for many natural DP-complete and particularly Θ_2^{p}-complete problems. Consequently, hardness results remained rather rare in the area of criticality and stability, despite the great attention that these natural notions have garnered from graph theorists ever since the seminal paper by Dirac [13] from 1952; see for example the classical textbooks by Harary [20, chapters 10 and 12] and Bollobás [3, chapter IV]—the latter having a precursor dedicated exclusively to extremal graph theory [4, chapters I and V]—and countless papers over the decades, of which we cite some selected examples from early to recent ones [14,21,1,43,18,22,12,27,11]. A pioneering complexity result by Papadimitriou and Wolfe [34, Theorem 1] establishes the DP-completeness of MinimalUnSat. (They call a formula minimally unsatisfiable if deleting an arbitrary clause renders it satisfiable, that is, if it is critical.) They also proved that determining, given a graph G and a $k \in \mathbb{N}$, whether G is k-ω-vertex-critical is a DP-complete problem [34, Theorem 4]. Later, Cai and Meyer [6] showed the DP-completeness of k-VertexCriticality (which they call Minimal-k-Uncolorability) for all $k \geq 3$. Burjons et al. [5] recently extended this result to the more difficult case of edge deletion, showing that k-Criticality is DP-complete for all $k \geq 3$ [5, Theorem 8]. They also provided the first Θ_2^{p}-hardness result for a criticality problem, namely for β-VertexCriticality [5, Theorem 15].

Note the drop in difficulty down to DP when fixing the graph number. This emerges as a general pattern, as evidenced by our results outlined in the contribution section below.

Stability, in contrast to criticality, has been sorely neglected by the computational complexity community, which is surprising in light of its apparent practical relevance—for example in the design of infrastructure, where stability is a most desirable property. As a small exception to this, Beacham and Culberson [2] proved a comparably easy variant of Unfrozenness, namely $\{(G, k) \mid \chi(G) \leq k$ and G is unfrozen$\}$, to be NP-complete.

1.4. Contribution

We choose four of the most prominent graph problems— Colorability, Vertex Cover, Independent Set, and Clique—to analyze the complexity of stability. We prove all of them to be Θ_2^{p}-complete for the default case of edge deletion. For unfrozenness—that is, stability with respect to edge addition—we prove the same, with the one exception of Colorability. For this problem, we prove that the existence of a construction with a few simple properties would be sufficient to prove Θ_2^{p}-completeness. Finally, we introduce the notion of two-way stability—stability with respect to both deleting and adding edges—and prove again Θ_2^{p}-completeness for all four problems. Table 1 provides an overview of these results, showcasing surprising contrasts between some of the problems.

We also derive several other useful results with broad appeal on their own, among these being the coDP-completeness of Stable3CNF [Theorem 12], the DP-completeness of k-Stability and k-VertexStability for all $k \geq 4$ [Theorem 17], general criteria for proving DP-hardness [Lemmas 28 and 29], and finally constructions such as the edge-stabilizing gadget

[6] Note that Wagner originally derived his results with respect to the more restricted form of polynomial-time reducibility via Boolean formulas, indicated by the *bf* in the class name. He later proved the resulting notions to be equivalent, however; that is, we have $\mathrm{P}_{\mathrm{bf}}^{\mathrm{p}} = \Theta_2^{\mathrm{p}}$ [41].

[7] Spakowski and Vogel explicitly proved the Θ_2^{p}-completeness of CompareVertexCover [38, Theorem 12], CompareClique and CompareIndependentSet [38, Theorem 13]. For other cases, see Lemma 29 and Theorem 26.

Table 1
An overview of our results regarding the complexity of different stability problems. See Section 7 for the results on Clique and Independent Set; almost all of them follow in analogy to the ones for Vertex Cover [Proposition 34], with α-VertexStability and ω-VertexStability being the exception [Proposition 35].

With respect to this base problem and graph number:	Stability		Unfrozenness		Two-Way Stability	
	Edge	Vertex	Edge	Vertex	Edge	Vertex
Vertex Cover, β	[Theorem 22] Θ_2^{p}-complete	[Theorem 20] P	[Theorem 24] Θ_2^{p}-complete	[Theorem 23] P	[Theorem 33] Θ_2^{p}-complete	[Theorem 30] P
Independent Set, α and Clique, ω	Θ_2^{p}-complete	Θ_2^{p}-complete	Θ_2^{p}-complete	P	Θ_2^{p}-complete	P
Colorability, χ	Θ_2^{p}-complete [Theorem 5]	Θ_2^{p}-complete [Theorem 6]	? [Theorem 25]	P [Theorem 23]	Θ_2^{p}-complete [Theorem 31]	P [Theorem 30]

[Lemma 18] that yields an AND_ω function for Stability [Corollary 19] and has potential applications in various contexts such as reoptimization and general graph theory. For example, it enhances the time-honored Hajós construction [19], which shows how we can build arbitrarily complex critical graphs, by allowing us to construct graphs with any given number of stable edges.

2. Basic observations

We begin with a few very basic and useful observations that will be used implicitly and, where appropriate, explicitly throughout the paper.

Observation 1. *The deletion of an edge or of a vertex either decreases the chromatic number by exactly one or leaves it unchanged.*

Proof. It is clear that deleting an edge or vertex cannot increase the chromatic number. To see that this cannot decrease it by more than one, it suffices to note that introducing a new edge or vertex can increase it by at most one since we can assign one new, unique color to the new vertex or to one of the two vertices of the inserted edge. □

Observation 2. *Let $e = \{u, v\}$ be a critical edge. Then u and v are critical as well.*

Proof. Since $G - u$ and $G - v$ are subgraphs of $G - e$, both $\chi(G - u)$ and $\chi(G - v)$ are at most $\chi(G - e)$, which is less than $\chi(G)$ because e is critical. Thus u and v are critical. □

Observation 3. *Let v be a stable vertex. Then all edges incident to v are stable.*

Proof. This follows immediately from the contrapositive of Observation 2. □

Observation 4. *Let G be a graph. A vertex $v \in V(G)$ is critical if and only if there is an optimal coloring of G that assigns v a color with which no other vertex is colored.*

Proof. Given a critical $v \in V(G)$, consider an arbitrary optimal coloring of $G - v$. Since v is critical, it uses one fewer color than the optimal colorings of G. We therefore obtain an optimal coloring of G by assigning v a new color. The converse is immediate. □

3. Stability and Vertex-Stability for Colorability

We will prove Θ_2^{p}-completeness for both Stability and VertexStability. On a very high level, this structure of this section can be summarized as follows: In general, only the lower bounds are hard to prove. Theorem 12 establishes the coDP-completeness of Stable3CNF, which is reduced to VertexStability in Theorem 16. Knowing VertexStability to be coDP-hard, we can now apply Corollary 8 to elevate this to Θ_2^{p}-hardness. This hardness result in turn transfers to Stability via the reduction from Lemma 7. We will not prove these results in this order, however, in an effort to keep the distance to our main goal minimal by finishing the more manageable parts first and avoiding having too many loose ends at a time. The section will conclude with Theorem 17 and Lemma 18, whose proofs mostly stand on their own.

We now formally state our two main goals for this section.

Theorem 5. *Determining whether a graph is stable is Θ_2^{p}-complete.*

Theorem 6. *Determining whether a graph is vertex-stable is Θ_2^{p}-complete.*

As is typical, the upper bounds are immediate: Recalling that $\Theta_2^p = P_{\parallel}^{NP}$, we can determine the chromatic numbers of a graph and all its 1-vertex-deleted and 1-edge-deleted subgraphs with a polynomial number of parallel queries to an oracle for the standard, NP-complete colorability problem $\{(G,k) \in \mathcal{G} \times \mathbb{N} \mid \chi(G) \leq k\}$. Specifically, the queries (G,k), $(G-e,k)$, and $(G-v,k)$ for every $e \in E(G)$, every $v \in V(G)$, and every $k \in \{0,\dots,\|V(G)\|\}$ suffice to find out whether G is stable and whether it is vertex-stable. We mention that we could also have relied on the definition $\Theta_2^p = P^{NP[\mathcal{O}(\log n)]}$ for proving this upper bound by finding the chromatic numbers via a binary search with a logarithmic number of adaptive queries. To prove the matching lower bounds, we first note that the lower bound for Theorem 6 implies the lower bound for Theorem 5.

Lemma 7. VertexStability *polynomial-time many-one reduces to* Stability.

Proof. We show that G is vertex-stable if and only if $G+G$ is stable. Assume that G is vertex-stable. Then $G+G$ is vertex-stable too as an immediate consequence of the general equation $\chi(G_1+G_2) = \chi(G_1)+\chi(G_2)$, which holds for arbitrary graphs G_1 and G_2 since the join edges force the vertices of G_1 and G_2 to use disjoint sets of colors. Hence $G+G$ is stable by Observation 3. For the converse, suppose that G is not vertex-stable. Then there is a vertex v with $\chi(G-v) = \chi(G)-1$. We will color of the graph $G+G$ from which the edge between the two copies of v has been deleted with $2(\chi(G)-1)+1 = \chi(G+G)-1$ colors, proving that $G+G$ is not stable. Fix any optimal coloring of $G-v$. We now color both copies of $G-v$ in $G+G$ according to it, but using two disjoint sets of $\chi(G)-1$ colors. Finally, we assign one additional new color to the two vertices corresponding to v. □

It remains to establish the lower bound of Theorem 6, that is, to prove that determining whether a graph is vertex-stable is Θ_2^p-hard. Proving Θ_2^p-hardness is not easy. However, we will now argue that it suffices to show that VertexStability is coDP-hard.

Chang and Kadin [8, Theorem 7.2] show that a problem is Θ_2^p-hard if it is DP-hard and has an OR_ω function. Observing that Θ_2^p is closed under complementation, we obtain the following corollary.

Corollary 8. *If a* coDP*-hard problem has an* AND_ω *function, then it is* Θ_2^p*-hard.*

We first note that VertexStability has an AND_ω function.

Theorem 9. *The join is an* AND_ω *function for* VertexStability *and* Unfrozenness.

Proof. Let $G_1,\dots,G_n$ be a finite number of graphs. Consider $G_1+\cdots+G_n$. For every $i \in \{1,\dots,n\}$, the join edges force the vertices $V(G_i)$ to have colors that are different from the colors of all remaining vertices. This implies $\chi(G_1+\cdots+G_n) = \chi(G_1)+\cdots+\chi(G_n)$. Moreover, vertex deletion and edge addition commute with joining: For every $v \in V(G_1+\cdots+G_n) = V(G_1)\cup\cdots\cup V(G_n)$, there is an i such that

$$(G_1+\cdots+G_n)-v = G_1+\cdots+G_{i-1}+(G_i-v)+G_{i+1}+\cdots+G_n.$$

Analogously, for every nonedge $e \in \overline{E}(G_1+\cdots+G_n) = \overline{E}(G_1)\cup\cdots\cup\overline{E}(G_n)$, there is an i such that

$$(G_1+\cdots+G_n)+e = G_1+\cdots+G_{i-1}+(G_i+e)+G_{i+1}+\cdots+G_n.$$

The claim of the theorem follows immediately. □

Now, Theorem 6 follows from Corollary 8 and the coDP-hardness of VertexStability stated in the following lemma.

Lemma 10. *Determining whether a graph is vertex-stable is* coDP*-hard.*

To prove Lemma 10, we show in Theorem 12 that Stable3CNF = 3Sat ∪ Stable3UnSat is coDP-complete and then reduce it to VertexStability in Theorem 16. We will use the following lemma twice.

Lemma 11. *There is a polynomial-time many-one reduction from* Sat *to* 3Sat *converting a* CNF*-formula* Φ *into a* 3CNF*-formula* Ψ *such that* Φ *is stable if and only if* Ψ *is stable.*

Proof. The standard clause-size reducing reduction maps a CNF-formula $\Phi = C_1\wedge\cdots\wedge C_m$ to $\Psi' = F_1\wedge\cdots\wedge F_m$ by splitting any clause $C = (\ell_1\vee\cdots\vee\ell_k)$ with k literals for a $k\geq 3$ into the k clauses of the subformula

$$F = (\ell_1\vee y_1)\wedge(\overline{y}_1\vee\ell_2\vee y_2)\wedge\cdots\wedge(\overline{y}_{k-2}\vee\ell_{k-1}\vee y_{k-1})\wedge(\overline{y}_{k-1}\vee\ell_k)$$

with fresh variables y_i that do not occur elsewhere.

We prove that Φ is stable if and only if Ψ' is. We already know that Φ is satisfiable exactly if Ψ' is. Thus it suffices to show that Φ is satisfiable after deleting C if and only if Ψ' is satisfiable after deleting some clause in F. Let α be a satisfying assignment for $\Phi - C$. We then obtain a satisfying assignment β for $\Psi' - (\ell_1 \vee y_1)$ by setting $\beta(y_1) = \cdots = \beta(y_{k-1}) = 0$ and the remaining variables as usual. For the converse, we simply observe that the restriction of an assignment satisfying Ψ' with an arbitrary clause from F deleted satisfies $\Phi - C$.

We now transform Ψ' into a formula $\Psi \in$ 3CNF that is stable exactly if Ψ' is. We do this by substituting for any two-literal clause $(\ell_1 \vee \ell_2)$ the subformula $(\ell_1 \vee \ell_2 \vee z) \wedge (\ell_1 \vee \ell_2 \vee \overline{z})$ and for any one-literal clause (ℓ_1) the subformula $(\ell_1 \vee z_1 \vee z_2) \wedge (\ell_1 \vee z_1 \vee \overline{z}_2) \wedge (\ell_1 \vee \overline{z}_1 \vee z_2) \wedge (\ell_1 \vee \overline{z}_1 \vee \overline{z}_2)$, where z, z_1, and z_2 are, for each substitution, new variables that do not occur anywhere else. It is now straightforward to check that Ψ has all the desired properties. $\square$

Theorem 12. Stable3CNF *is* coDP*-complete.*

The proof of Theorem 12 is based on a simple corollary to the following observation by Chang and Kadin: If a set is NP-hard, coNP-hard, and it has an OR_2 function, then it is DP-hard [8, Lemma 5].

Corollary 13. *If a set is* NP*-hard,* coNP*-hard, and it has an* AND_2 *function, then it is* coDP*-hard.*

The proof of the corollary is immediate since an AND_2 function for one language is an OR_2 function for its complement and vice versa. We can now start with the proof of Theorem 12.

Proof of Theorem 12. It is immediate that Stable3CNF is in coDP. By Corollary 13, it now suffices to show that Stable3CNF is coNP-hard, NP-hard, and that it has an OR_2 function.

coNP-hardness. It is easy to see that the function

$$\begin{aligned} f: \Phi \mapsto \Phi &\wedge (x \vee y \vee z) \wedge (x \vee y \vee \overline{z}) \wedge (x \vee \overline{y} \vee z) \wedge (x \vee \overline{y} \vee \overline{z}) \\ &\wedge (\overline{x} \vee y \vee z) \wedge (\overline{x} \vee y \vee \overline{z}) \wedge (\overline{x} \vee \overline{y} \vee z) \wedge (\overline{x} \vee \overline{y} \vee \overline{z}), \end{aligned}$$

where x, y, and z are fresh variables not occurring in Φ, reduces 3UnSat to Stable3CNF.

NP-hardness. We give a reduction from 3Sat to Stable4CNF; composing it with the reduction from Lemma 11 yields the desired reduction to Stable3CNF. Given a 3CNF-formula $\Phi = C_1 \wedge \cdots \wedge C_m$ over $X = \{x_1, \ldots, x_n\}$, map it to the 4CNF-formula

$$\begin{aligned} \Psi = &(C_1 \vee y) \wedge (C_1' \vee y') \wedge (C_1'' \vee y'') \wedge \cdots \wedge \\ &(C_m \vee y) \wedge (C_m' \vee y') \wedge (C_m'' \vee y'') \wedge \\ &(\overline{y} \vee \overline{y}' \vee \overline{y}'' \vee z) \wedge (\overline{y} \vee \overline{y}' \vee \overline{y}'' \vee \overline{z}), \end{aligned}$$

where the clauses C_i' and C_i'' are just like the clauses C_i but with a new copy of variables $X' = \{x_1', \ldots, x_n'\}$ and $X'' = \{x_1'', \ldots, x_n''\}$ instead of X, respectively, and y, y', y'', and z being four fresh variables as well. Deleting the clause $(\overline{y} \vee \overline{y}' \vee \overline{y}'' \vee \overline{z})$ renders Ψ trivially satisfiable; any assignment that sets y, y', y'', and z to 1 will do. Thus Ψ is stable if and only if it is satisfiable. It remains to prove the equisatisfiability of Φ and Ψ.

First assume that Φ has a satisfying assignment $\sigma : X \to \{0,1\}$. Then Ψ is satisfied by any assignment τ with $\tau(x_i) = \tau(x_i') = \tau(x_i'') = \sigma(x_i)$ for $i \in \{1, \ldots, n\}$ and $\tau(y) = 0$. Now assume that Ψ has a satisfying assignment τ. The last two clauses $(\overline{y} \vee \overline{y}' \vee \overline{y}'' \vee z)$ and $(\overline{y} \vee \overline{y}' \vee \overline{y}'' \vee \overline{z})$ of Ψ guarantee that $\tau(y) = 0$, $\tau(y') = 0$ or $\tau(y'') = 0$. In the first case, Φ is satisfied by $\sigma : x_i \mapsto \tau(x_i)$, in the second case by $\sigma' : x_i \mapsto \tau(x_i')$, and in the third case by $\sigma'' : x_i \mapsto \tau(x_i'')$.

OR_2. In their proof of DP-completeness, Papadimitriou and Wolfe [34, Lemma 3 plus corollary] implicitly gave a simple AND_2 function for both MinimalUnSat and Minimal3UnSat (the sets of unsatisfiable formulas that become satisfiable after deleting any clause). We make use of the same construction.

Let $\Phi = C_1 \wedge \cdots \wedge C_m$ and $\Phi' = C_1' \wedge \cdots \wedge C_{m'}'$ be two given 3CNF-formulas. Without loss of generality, Φ and Φ' have disjoint variable sets. Let

$$\Psi = \bigwedge_{1 \le i \le m, 1 \le j \le m'} (C_i \vee C_j').$$

Note that Ψ is in 6CNF and equivalent to $\Phi \vee \Phi'$. We will show that $\Psi \in$ Stable6CNF if and only if $\Phi \in$ Stable3CNF or $\Phi' \in$ Stable3CNF. Setting the clause length of Ψ to exactly 3 by applying Lemma 11 then yields the desired OR_2-reduction.

First assume that neither Φ nor Φ' is in Stable3CNF. Then we have $\Phi, \Phi' \notin$ 3Sat and there are $\hat{\imath} \in \{1, \ldots, m\}$ and $\hat{\jmath} \in \{1, \ldots, m'\}$ and assignments σ and σ' such that σ satisfies $\Phi - C_{\hat{\imath}}$ and σ' satisfies $\Phi' - C_{\hat{\jmath}}'$. Then we have

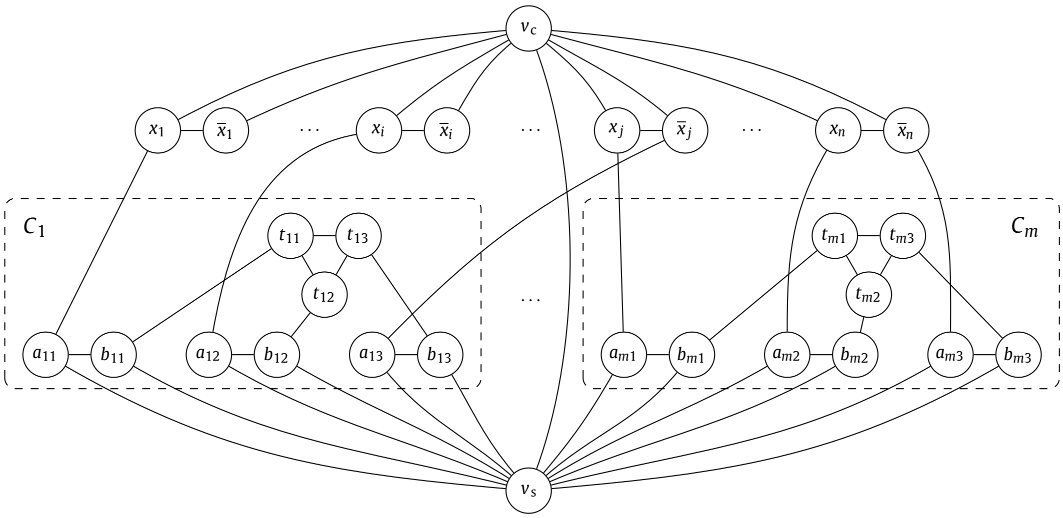


Fig. 1. The graph G_Φ for a 3CNF-formula with $C_1 = x_1 \vee x_i \vee \bar{x}_j$ and $C_m = x_j \vee x_n \vee \bar{x}_n$. The construction is due to Cai and Meyer [6], the figure to Burjons et al. [5, Figure 1, slightly modified].

$\Psi \notin$ SAT and (σ, σ') satisfies $(C_i \vee C'_j)$ for all $(i, j) \neq (\hat{\imath}, \hat{\jmath})$. It follows that $\Psi - (C_{\hat{\imath}} \vee C'_{\hat{\jmath}})$ is satisfiable, and thus $\Psi \notin$ STABLE6CNF.

Now assume that $\Psi \notin$ STABLE6CNF. Then $\Psi \notin$ SAT, and hence $\Phi, \Phi' \notin$ 3SAT. There are indices $\hat{\imath}$ and $\hat{\jmath}$ such that $\Psi - (C_{\hat{\imath}} \vee C'_{\hat{\jmath}})$ is satisfiable, say by assignment τ. This τ satisfies $(C_i \vee C'_j)$ for all $(i, j) \neq (\hat{\imath}, \hat{\jmath})$. In particular, τ satisfies $(C_i \vee \Phi')$ for all $i \neq \hat{\imath}$ and $(\Phi \vee C'_j)$ for all $j \neq \hat{\jmath}$. Since $\Phi, \Phi' \notin$ 3SAT, this implies that τ satisfies C_i for all $i \neq \hat{\imath}$ and C'_j for all $j \neq \hat{\jmath}$. It follows that $\Phi, \Phi' \notin$ STABLE3CNF.

This concludes the proof that STABLE3CNF is coDP-complete. □

All that is left to do is to reduce STABLE3CNF to VERTEXSTABILITY. First, we consider the known reduction from MINIMAL-3UNSAT to VERTEXMINIMAL3UNCOLORABILITY by Cai and Meyer [6].

It maps a formula Φ over the variable set $\{x_1, \dots, x_n\}$ with m 3-clauses $C_1, \dots, C_m$ to the graph G_Φ constructed as follows. For every clause C_i, start with a triangle on three new vertices t_{i1}, t_{i2}, and t_{i3}, add three disjoint edges $\{a_{i1}, b_{i1}\}$, $\{a_{i2}, b_{i2}\}$, and $\{a_{i3}, b_{i3}\}$, and then insert the edges $\{b_{i1}, t_{i1}\}$, $\{b_{i2}, t_{i2}\}$, and $\{b_{i3}, t_{i3}\}$. For every variable x_j, add an isolated edge $\{x_j, \bar{x}_j\}$. For every $i \in \{1, \dots, m\}$ and every $k \in \{1, 2, 3\}$, connect a_{ik} to the vertex representing the kth literal of C_i. Finally, add a vertex v_s connected to the vertices a_{ik} and b_{ik}, for every $i \in \{1, \dots, m\}$ and every $k \in \{1, 2, 3\}$, and add a vertex v_c connected to the vertices x_j and $\bar{x}_j$ for all $j \in \{1, \dots, n\}$. See Figure 1 [5, Figure 1, slightly modified] for an illustration of the full construction, combining the single steps described in the original paper [6].

It comes as no surprise that this reduction does not work for us since, for example, $G_\Phi - v_s$ is always 3-colorable, and thus G_Φ is never stable if Φ is not satisfiable. However, careful checking reveals the following important property of G_Φ.

Lemma 14. *A* 3CNF-*formula Φ is not stable if and only if $\chi(G_\Phi) > \chi(G_\Phi - t_{i1})$ for at least one $i \in \{1, \dots, m\}$.*

Proof of Lemma 14. As stated by Cai and Meyer [6, Lemma 2.2], Φ is satisfiable if and only if G_Φ is 3-colorable. Note that a Φ that is not stable is not satisfiable. Therefore, it suffices to check that $\Phi - C_i$ is satisfiable if and only if $G_\Phi - t_{i1}$ is 3-colorable. The mentioned paper proves this implication "by picture" [6, Figure 2.12] and states the converse in the second-to-last paragraph of its second section. □

What we need now is a way to enhance the construction such that the deletion of a vertex other than $t_{11}, \dots, t_{m1}$, for example v_s, does not decrease the chromatic number. We achieve this by the following lemma.

Lemma 15. *Let G be a graph and $v \in V(G)$. Let $\widehat{G}$ be the graph that results from replicating v; that is, $V(\widehat{G}) = V(G) \cup \{v'\}$ and $E(\widehat{G}) = E(G) \cup \{\{v', w\} \mid \{v, w\} \in E(G)\}$. Then $\chi(G) = \chi(\widehat{G}) = \chi(\widehat{G} - v) = \chi(\widehat{G} - v')$.*

Proof. The only nontrivial part is to show that $\chi(\widehat{G}) \leq \chi(G)$. To see this, we start with an arbitrary optimal valid vertex coloring of G and then color v' with the same color as v. □

Lemma 15 is simple and yet very powerful in our context. It allows us to select a set of vertices whose removal will not influence the chromatic number, and thus will not influence whether or not the graph is vertex-stable. We can use this to obtain the desired reduction.

Theorem 16. Stable3CNF *polynomial-time many-one reduces to* VertexStability.

Proof. Given a 3CNF-formula Φ, map it to $r(G_\Phi)$, where G_Φ is the graph from the reduction by Cai and Meyer [6] and r denotes the replication of all vertices other than $t_{11}, \dots, t_{m1}$.

If Φ is not in Stable3CNF, then we have $\chi(G_\Phi) > \chi(G_\Phi - t_{i1})$ for some $i \in \{1, \dots, m\}$ by Lemma 14. Furthermore, a repeated application of Lemma 15 yields $\chi(r(G_\Phi)) = \chi(G_\Phi)$ and $\chi(r(G_\Phi) - t_{i1}) = \chi(r(G_\Phi - t_{i1})) = \chi(G_\Phi - t_{i1})$. Thus $r(G_\Phi)$ is not vertex-stable. For the converse, suppose that $r(G_\Phi)$ is not vertex-stable. Let $v \in V(r(G_\Phi))$ be a vertex such that $\chi(r(G_\Phi)) > \chi(r(G_\Phi) - v)$. From Lemma 15, we can see that $v = t_{i1}$ for some $i \in \{1, \dots, m\}$. By Lemma 14, this implies that Φ is not stable. $\square$

This completes the proof of Theorem 6— stating that VertexStability is Θ_2^{p}-complete—which in turn implies Theorem 5, the Θ_2^{p}-completeness of Stability, by Lemma 7. Now we briefly turn to some DP-complete problems. Recall that by prefixing a number k to the name of a stability property we additionally require the graph number to be exactly k.

Theorem 17. *The problems* k-Stability *and* k-VertexStability *are* NP-*complete for* $k = 3$ *and* DP-*complete for* $k \geq 4$.

Proof. The membership proofs are immediate. For the lower bound we use that Exact-k-Colorability (the class of all graphs whose chromatic number is not merely at most, but exactly k) is NP-complete for $k = 3$ and DP-complete for $k \geq 4$; see [36]. It suffices to check that mapping G to $G \cup G$ reduces Exact-k-Colorability to k-Stability and k-VertexStability. Indeed, for any two graphs H and H', we have $\chi(H \cup H') = \max\{\chi(H), \chi(H')\}$, implying that $G \cup G$ is stable and vertex-stable with $\chi(G) = \chi(G \cup G)$. $\square$

In the previous proof, we used the disjoint union of a graph with itself to render it stable without changing its chromatic number. Using a far more complicated construction, we can also ensure the stability of an arbitrary set of edges of a graph while keeping track of how exactly this changes the chromatic number. This immediately yields an explicit AND_ω function for Stability, which we will formally state in Corollary 19. Moreover, this stabilizing construction is likely to have applications in reoptimization, the recently introduced neighborly-help model [5], and graph theory in general. We now state the result in the following lemma.

Lemma 18. *There is a polynomial-time algorithm that, given any graph G plus a nonempty subset $S \subseteq E(G)$ of its edges, adds a fixed gadget to the graph and then substitutes for every $e \in S$ some gadget that depends on G and e, yielding a graph $\widehat{G}$ with the following properties:*

1. $\chi(\widehat{G}) = \chi(G) + 2$.
2. *All edges in $E(\widehat{G}) - (E(G) - S)$ are stable.*
3. *Each one of the remaining edges in $E(G) - S$ is stable in $\widehat{G}$ exactly if it is stable in G.*

Proof. Let a graph G and a subset $S \subseteq E(G)$ of edges in it be given. We first describe the construction of $\widehat{G}$ in detail. Figure 2 exemplifies the full construction for a simple graph G and an S that contains exactly one edge $e = \{v_1, v_2\}$.

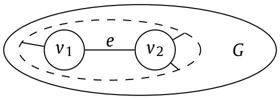
We begin by adding a cycle on four new vertices w_1', w_2', w_1'', w_2'' and joining it to G. Then we complete the procedure described in the following paragraph for each edge $e \in S$.

We add in disjoint union a copy G_e' of the original graph G. We distinguish the vertices of G_e' from the ones of G by adding a prime and the subindex e to them. In the example, where the edge $e = \{v_1, v_2\}$ is to be stabilized, there will thus be two adjacent vertices $v_{1,e}', v_{2,e}' \in V(G_e')$, as shown in Figure 2. Now we join two new vertices to G_e'. One of them we merge with v_1; the other one we call u_e' and connect it to v_2. We then replicate u_e' and every $v' \in V(G_e')$, marking the replicas with a second prime. Excluding u'', these replicas constitute another, empty copy of G, which we denote G_e''. Finally, we delete the edge $e = \{v_1, v_2\}$.

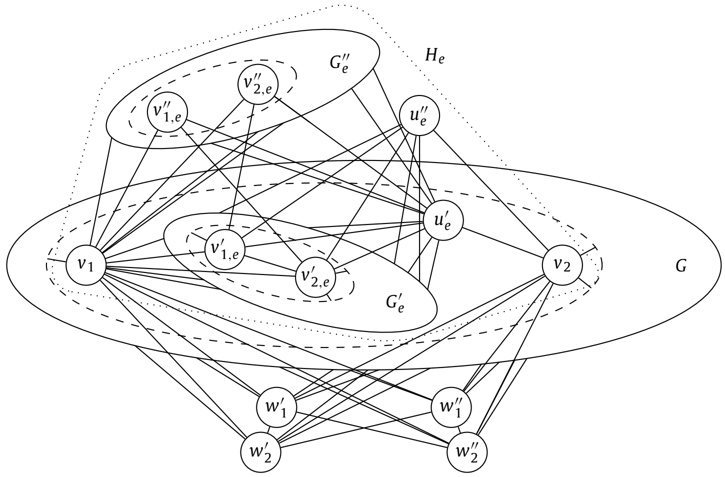
This completes the construction; we call the resulting graph $\widehat{G}$. For each edge $e = \{v_1, v_2\} \in S$, we denote the induced subgraph of $\widehat{G}$ on the vertices $V(G_e') \cup V(G_e'') \cup \{u_e', u_e'', v_1, v_2\}$ by H_e. We now examine the induced subgraph H_e as a gadget that depends on G and substitutes e.

In the following paragraph, we prove an essential property of the graph H_e, namely that it behaves exactly like the deleted edge e as far as $(\chi(G) + 1)$-colorability is concerned, whereas it acts like a nonedge with regards to $(\chi(G) + 2)$-colorability.

Assume that $\chi(G) + 1$ colors are available. First, let v_1 and v_2 be colored by two different ones of them. We can then extend this to a $(\chi(G) + 1)$-coloring of H_e in the following way. We assign to u_e' and u_e'' the color of v_1, choose an arbitrary coloring of G using the remaining $\chi(G)$ colors, and then assign it to both G_e' and G_e''. Now, pick instead an arbitrary color



(a) An example of a simple given graph G with only a single edge $e = \{v_1, v_2\}$ to be stabilized. The vertices outside the dashed ellipse and edge parts leading to them are omitted.



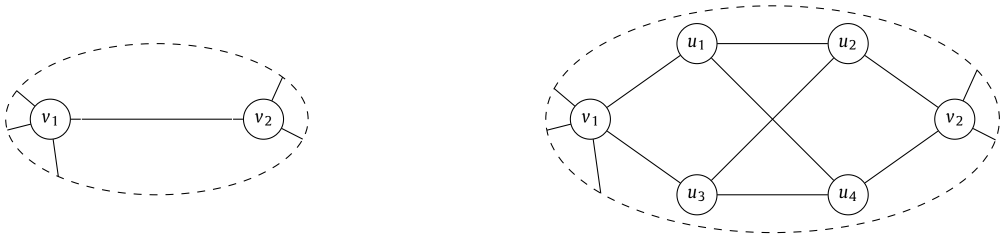
(b) The graph $\widehat{G}$, satisfying all properties stated in Lemma 18. In the simple case of our example that S is only a singleton $\{e\}$, the construction contains three partly modified copies of G, with the same vertices and edge parts as in Figure 2a omitted from the picture.

Fig. 2. Example of the construction of Lemma 18, rendering arbitrary subsets of edges stable.

and let both v_1 and v_2 be colored by it. While this can be extended to a $(\chi(G)+2)$-coloring of H_e immediately—assign a second color to u'_e and u''_e and then color G'_e and G''_e with the remaining $\chi(G)$ colors—it is impossible to extend it to a $(\chi(G)+1)$-coloring of H_e as we show now. Seeking contradiction, assume there were such a coloring. It must assign to u''_e a color different from the one color assigned to both v_1 and v_2. Since the vertices of G'_e are all adjacent to both v_1 and u''_e, they must be colored with the remaining $\chi(G)-1$ colors, yielding the desired contradiction to $\chi(G'_e) = \chi(G)$.

Returning to the entire graph, it remains to show that $\widehat{G}$ has the three stated properties.

1. Let a $\chi(G)$-coloring of G be given. We describe how to extend it to a $(\chi(G)+2)$-coloring of $\widehat{G}$. We assign one of the two new colors to w'_1 and w''_1 and the other one to w'_2 and w''_2. Now, for every $e = \{v_1, v_2\} \in S$, we color the gadget H_e as follows. To both u'_e and u''_e we assign the color of v_1. Now color both G'_e and G''_e according to the initially given $\chi(G)$-coloring of G with one modification, namely swapping out the two colors assigned to v_1 and v_2, wherever they occur, for the two new colors. We can check that this yields a valid $(\chi(G)+2)$-coloring of $\widehat{G}$, proving that $\chi(\widehat{G}) \le \chi(G)+2$.
   For the reverse inequality, assume by contradiction that $\widehat{G}$ has a $(\chi(G)+1)$-coloring. We observe two properties of this coloring. On the one hand, it uses at most $\chi(G)-1$ colors for the vertices $V(G)$ since they are all adjacent to the 2-clique $\{w'_1, w'_2\}$. On the other hand, the restriction to $V(H_e)$, for any $e \in S$, is a $\chi(G)+1$ coloring of H_e, which implies—by the edge-like behavior of H_e under this circumstance proved above—that v_1 and v_2 are assigned different colors. Combining these two insights, we see that the restriction of our coloring to $V(G)$ is a $(\chi(G)-1)$-coloring of G, yielding the desired contradiction.
2. First note that all edges in S are deleted during the described construction of $\widehat{G}$, implying $E(\widehat{G}) - E(G) = E(\widehat{G}) - (E(G) - S)$. We show the vertices of $V(\widehat{G}) - V(G)$ to be stable. Observe that they can be partitioned into the single-primed and the double-primed ones; the latter were constructed as the replicas of the former, and no edges were added after this. We recall from Lemma 15 that replicating a vertex renders it and its replica stable because they have exactly the same neighborhood but are not adjacent. Each edge e in $E(\widehat{G}) - E(G)$ is adjacent to one of the stable vertices in $V(\widehat{G}) - V(G)$ and therefore stable itself by Observation 3.
3. Note that the described construction does not commute with the deletion of an arbitrary edge $e \in E(G)$. (That is, denoting the construction by $f : G \mapsto \widehat{G}$, we do not have $f(G-e) = f(G) - e$.) This stands in contrast to the situation with the other stabilizing constructions in this paper—where commutativity of the construction immediately yields the corresponding property—necessitating an independent proof in this case.
   Let $e' \in E(G) - S$. Assume first that e' is critical in G. Then there is a $(\chi(G)-1)$-coloring of $G - e'$. We can extend it to a $(\chi(G)+1)$-coloring of $\widehat{G} - e'$ by assigning one of the two new colors to w'_1 and w'_2, the other one to w'_2 and w''_2, and then use the already proven fact that, for each $e \in S$, the graph H_e can be colored with $\chi(G)+1$ colors whenever two different colors are prescribed for the two endpoints of e. This proves $\chi(\widehat{G} - e') \ne \chi(\widehat{G})$ and thus the criticality of e' in $\widehat{G}$.
   Now we start with the assumption that e' is critical in $\widehat{G}$. By the first property of Lemma 18 we have $\chi(\widehat{G} - e') = \chi(G)+1$. Pick an arbitrary $(\chi(G)+1)$-coloring of $\widehat{G} - e'$. Since the induced 4-cycle on $\{w'_1, w'_2, w''_1, w''_2\}$ has chromatic number 2 and is joined to the induced subgraph on $V(G)$, the induced coloring on $G - e'$ uses at most $\chi(G)-1$ of these colors, implying $\chi(G - e') \le \chi(G)-1$ and thus the criticality of e' in G. We deduce via the contrapositive that e' is stable in G if and only if it is stable in $\widehat{G}$.



(a) Example of the relevant section of G.

(b) The same section in G' after the substitution.

Fig. 3. Illustration of the substitution of an edge $\{v_1, v_2\}$ by the gadget mentioned in Lemma 21. We remark in passing that the gadget used here is the smallest one with the desired properties.

This concludes the proof of Lemma 18. $\square$

Note that this construction allows us to reduce the problem of deciding whether in a given selection of edges all of them are stable to Stability by stabilizing all other edges. Moreover, it yields the following AND_ω function for Stability, which is stated in the following corollary,

Corollary 19. *Mapping k graphs $G_1, \ldots, G_k$ to $G_1 + \cdots + G_k$ with all join edges stabilized using the construction from Lemma 18 is an AND_ω function for* Stability.

Proof. We know that $\chi(G_1 + \cdots + G_k) = \chi(G_1) + \cdots + \chi(G_k)$. This implies that, for any $i \in \{1, \ldots, k\}$, an edge $e \in E(G_i)$ is stable in G_i exactly if it is stable in $G_1 + \cdots + G_k$. The graph with all join edges stabilized is thus stable exactly if all graphs $G_1, \ldots, G_k$ are. $\square$

Note that the more complicated formulation of Lemma 18 that allows for the stabilization of arbitrary subsets of edges rather than just a single chosen edge is crucial for the derivation of Corollary 19. For since the construction of Lemma 18 more than doubles the number of vertices—even if S is only a singleton—at most a logarithmic number of iterated applications are possible in polynomial time.

4. Stability and Vertex-Stability for Vertex Cover

We will now examine the complexity of stability with respect to the vertex-cover number β. First, we note that β-VertexStability is trivially in P as it consists of the empty graphs.

Theorem 20. *Only the empty graphs are β-vertex-stable.*

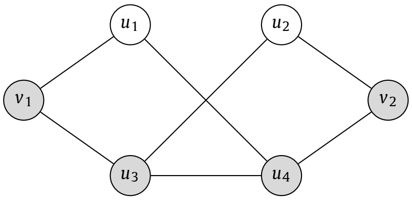
Proof. Let G be a graph. If G is empty, it is β-vertex-stable since the empty set is a minimum vertex cover for both G and $G - v$ for every $v \in V(G)$. If G has an edge $\{u, v\}$, every vertex cover contains either u or v or both. Consider any optimal vertex cover X of G and assume, without loss of generality, that $v \in X$. Then v is a critical vertex since $X - \{v\}$ is a vertex cover of size $\|X\| - 1$ of $G - v$. $\square$

Turning to the smaller change of deleting only an edge instead of a vertex, the situation changes radically. We will prove with Theorem 22 that determining whether a graph is β-stable is Θ_2^{P}-complete. An important ingredient to the proof is the following analogue of Lemma 15, which shows how to β-stabilize an arbitrary edge of a given graph.

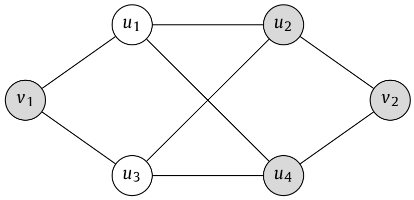
Lemma 21. *Let G be a graph and $\{v_1, v_2\} \in E(G)$ one of its edges. Create from G a new graph G' by replacing the edge $\{v_1, v_2\}$ by the gadget that consists of four new vertices u_1, u_2, u_3, and u_4 with edges $\{u_1, u_2\}$, $\{u_2, u_3\}$, $\{u_3, u_4\}$, and $\{u_4, u_1\}$ (i.e., a new rectangle) and additionally the edges $\{v_1, u_1\}$, $\{v_1, u_3\}$, $\{v_2, u_2\}$, and $\{v_2, u_4\}$. (This gadget is displayed in Figure 3b.) Then we have $\beta(G') = \beta(G) + 2$, all edges of the gadget are stable in G', and the remaining edges are stable in G' if and only if they are stable in G.*

Proof. Let X be a vertex cover for G. Due to $\{v_1, v_2\} \in E(G)$, we have $v_1 \in X$ or $v_2 \in X$. We obtain a vertex cover for G' by adding u_2 and u_4 to X in the first case and u_1 and u_3 in the second case. This shows $\beta(G') \le \beta(G) + 2$. For the reverse inequality observe that out of the four vertices $\{u_1, u_2, u_3, u_4\}$ inducing a 4-cycle every vertex cover for G' contains at least two and removing all of them leaves us with a vertex cover for G.

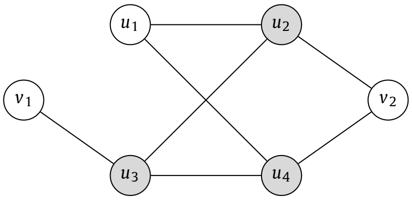
Now, let e be an arbitrary edge of G other than $\{v_1, v_2\}$. We can check that the argument above still holds true for $G - e$ and $G' - e$ instead of G and G', and hence $\beta(G' - e) = \beta(G - e) + 2$. It follows that $\beta(G) - \beta(G - e) = \beta(G') - \beta(G' - e)$; that is, e is stable in G exactly if it is stable in G'.



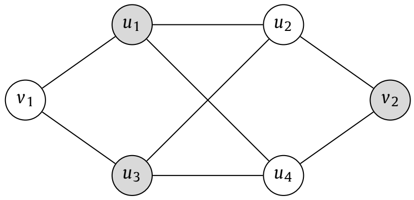
(a) A vertex cover X for $G' - \{u_1, u_2\}$. Only the gadget part is shown.



(b) Removing u_3 and adding u_2 yields a vertex cover of the same size for G'.



(c) A vertex cover Y for $G' - \{v_1, u_1\}$.



(d) A vertex cover of the same size for G'.

Fig. 4. An illustration of the proof of Lemma 21: All edges of the gadget are β-stable in G'.

Finally, we prove β-stability for all the gadget edges. Let e be such an edge. It suffices to show that, for every vertex cover of $G' - e$, there is a vertex cover of the same size for G'. We show this for $e = \{u_1, u_2\}$ and $e = \{v_1, u_1\}$; the remaining cases follow immediately by symmetry. The following argumentation is illustrated in Figure 4. Let X be a vertex cover of $G' - \{u_1, u_2\}$. If X contains u_1 or u_2, it is a vertex cover for G' as well. Otherwise, we have $\{v_1, u_3, u_4, v_2\} \subseteq X$ since the edges incident to u_1 and u_2 need to be covered—see Figure 4a—and $\{u_2\} \cup X - \{u_3\}$ is another vertex cover for G'; see Figure 4b. Analogously, let Y be a vertex cover of $G' - \{v_1, u_1\}$. If Y contains v_1 or u_1, then it is a vertex cover of G' as well. Otherwise, Y contains the vertices $\{u_2, u_3, u_4\}$ since they are neighbors of either v_1 or u_1; see Figure 4c. Then, $\{u_1, v_2\} \cup Y - \{u_2, u_4\}$ is a vertex cover for G' that is either—see Figure 4d—of the same size as Y or, if $v_2 \in Y$, smaller by one. $\square$

Theorem 22. *Determining whether a graph is β-stable is Θ_2^{p}-complete.*

Proof. We reduce from $\{(G, H) \in \mathcal{G}^2 \mid \beta(G) < \beta(H)\}$, which is Θ_2^{p}-hard [38, Theorem 12]. (Note that this language is essentially the complement of CompareVertexCover and that Θ_2^{p} is closed under taking the complement.) Let G and H be given graphs. Replace each edge $e \in E(G)$ by a copy of the stabilizing gadget described in Lemma 21. Call the resulting graph G'. Clearly, we have $\|V(G')\| = \|V(G)\| + 4\|E(G)\|$. By Lemma 21, G' is β-stable and $\beta(G') = \beta(G) + 2\|E(G)\|$. Moreover, let $H' = H \cup K_2$. The edge in K_2 ensures that H' is not β-stable. Moreover, we have $\beta(H') = \beta(H) + 1$ and $\|V(H')\| = \|V(H)\| + 2$.

Now, let $G'' = G'$, just for consistent notation, and $H'' = H' \cup K_{2\|E(G)\|}$. Since $\beta(K_n) = n - 1$ for $n \geq 1$, this implies $\beta(G'') - \beta(G) = 2\|E(G)\| = \beta(H'') - \beta(H)$ whenever $\|E(G)\| \geq 1$, which we can assume without loss of generality by handling the trivial case of an empty graph G separately. We finish the construction by adding isolated vertices to either G'' or H'' such that we achieve an equal number of vertices without changing the vertex cover number; that is, we let

$$G''' = G'' \cup I_{\max\{0, \|V(H'')\| - \|V(G'')\|\}} \text{ and}$$
$$H''' = H'' \cup I_{\max\{0, \|V(G'')\| - \|V(H'')\|\}}.$$

Let $c = \|V(G''')\| = \|V(H''')\|$ and $d = \beta(G''') - \beta(G) = \beta(H''') - \beta(H)$. Note that G''' is β-stable since we stabilized G' with the gadget substitutions and then only added isolated vertices but no more edges. Moreover, H''' is not β-stable due to the β-critical edge of K_2.

Let S be the join $G''' + H'''$ with all join edges stabilized, again by the gadget substitution described in Lemma 21. It is easy to see from the proof of Lemma 21 that the gadget as a whole behaves just like the edge it replaces, in the sense that an optimal vertex cover of the whole graph contains, without loss of generality, either v_1 or v_2 or both. Therefore, an optimal vertex cover of S consists of either an optimal vertex cover of G''' and all vertices of H''' or of an optimal vertex cover of H''' and all vertices of G''' plus, in both cases, a constant number k of vertices for covering the gadget edges—namely two for each former join edge, that is, $k = 2 \cdot \|V(G''')\| \cdot \|V(H''')\|$. In the first case, we obtain an optimal vertex cover for S of size $\beta(G''') + c + k = \beta(G) + d + c + k$, in the second case one of size $\beta(H''') + c + k = \beta(H) + d + c + k$.

Assume first that $\beta(G) < \beta(H)$. It follows that $\beta(G''') < \beta(H''')$ and thus any optimal vertex cover for S consists of all vertices $V(H''')$, an optimal vertex cover for G''', and k vertices for the gadgets. Since we ensured that G''' is β-stable, S is β-stable. Now, assume that $\beta(G) \geq \beta(H)$. Then there is an optimal vertex cover that consists of all vertices of G''', an optimal vertex cover of H''', and again k vertices due to the gadgets. Since H''' not β-stable, as pointed out above, S is not β-stable either. We conclude that S is β-stable exactly if $\beta(G) < \beta(H)$, thus proving that β-stability is Θ_2^{p}-hard and therefore Θ_2^{p}-complete. □

5. Unfrozenness

We begin with the observation that both for Colorability and for Vertex Cover adding a vertex is too generous a modification to be interesting.

Theorem 23. *There is no vertex-unfrozen graph and only one β-vertex-unfrozen graph, namely the null graph (i.e., the graph with the empty vertex set).*

Proof. Adding a universal vertex increases both χ and β by exactly one, with the exception of the null graph K_0, for which $\chi(K_0) = \beta(K_0) = 0$ but $\chi(K_1) = 1 \neq \beta(K_1) = 0$. □

Both problems are far more interesting in the default setting, that is, for adding edges. The Θ_2^{p}-completeness of deciding whether a given graph is β-unfrozen can be obtained by a method similar to the one we used to establish Theorem 22.

Theorem 24. *Determining whether a graph is β-unfrozen is Θ_2^{p}-complete.*

Proof. The upper bound is again immediate. For hardness, we reduce from CompareVertexCover $= \{(G, H) \in \mathcal{G}^2 \mid \beta(G) \leq \beta(H)\}$—known to be Θ_2^{p}-hard [38, Theorem 12]— to β-Unfrozenness.

Let G and H be the two given graphs. We show how to construct in polynomial time from G and H a graph J that is β-unfrozen if and only if $\beta(G) \leq \beta(H)$. Denote $g = \|V(G)\|$ and $h = \|V(H)\|$. Assume without loss of generality that $g > 1$. Let $G' = (G \cup I_h) + (G \cup I_h)$. Note that G' is β-unfrozen, with $\beta(G') = \beta(G) + g + h$, and that $\|V(G')\| = 2(g + h)$. Let $H' = (H + K_{g+h}) \cup I_g$ (i.e., join a $(g + h)$-clique to H and then add g isolated vertices). Note that H' is not β-unfrozen—due to the β-frozen edges that can be added between any two of the $g \geq 2$ vertices of I_g—with $\beta(H') = \beta(H + K_{g+h}) = \beta(H) + g + h$ and that $\|V(H')\| = (h + (g + h)) + g = 2(g + h)$. Let $c = 2(g + h)$. We conclude that G' is β-unfrozen, that $\beta(G') \leq \beta(H') \iff \beta(G) \leq \beta(H)$, that $\|V(G')\| = \|V(H')\| = c$, and that H' is β-frozen. Now, let $J = G' + H'$. Clearly, J can be constructed in polynomial time. We will prove that J is β-unfrozen if and only if $\beta(G) \leq \beta(H)$. For both directions, note that due to $\|V(G')\| = \|V(H')\| = c$, we have $\beta(J) = \min\{\beta(G'), \beta(H')\} + c$.

First, assume that $\beta(G) \leq \beta(H)$. Then $\beta(G') \leq \beta(H')$ and thus $\beta(J) = \beta(G') + c$. We prove that all nonedges $e \in \overline{E}(J)$ are β-unfrozen in J. Such an e is either adjacent to two vertices of G' or to two vertices of H'. For the first case, note that $\beta(G' + e) = \beta(G')$ since G' is β-unfrozen. Thus we have $\beta(J + e) = \min\{\beta(G' + e), \beta(H')\} + c = \beta(J)$. In the second case we have $\beta(J + e) = \min\{\beta(G'), \beta(H' + e)\} + c = \beta(G') + c = \beta(J)$, where the second equality follows from $\beta(G') \leq \beta(H') \leq \beta(H' + e)$. Thus J is β-unfrozen if $\beta(G) \leq \beta(H)$.

For the converse, assume that $\beta(G) > \beta(H)$, implying $\beta(G') > \beta(H')$ and thus $\beta(J) = \beta(H') + c$. Since H' is β-frozen, there is a β-frozen nonedge e that can be added to H', yielding $\beta(H' + e) = \beta(H') + 1 \leq \beta(G)$. Hence, we have $\beta(J + e) = \min\{\beta(G'), \beta(H' + e)\} + c = \beta(H') + 1 + c > \beta(J)$. This shows that e is β-frozen for J as well, concluding the proof. □

Now, we would like to show the analogous result that Unfrozenness is Θ_2^{p}-complete as well. This turns out to be a very difficult task, however. There are many clues suggesting the hardness of Unfrozenness, which exhibits a far richer structure than all of the problems listed in Table 1 as easy. The latter problems are either empty or singletons or consist of all independent sets or all cliques, while Unfrozenness contains large classes of different graphs. We can even produce arbitrarily many new complicated unfrozen graphs using the graph join. There are no clearly identifiable characteristics to these unfrozen graphs to be leveraged. Instead, we give a sufficient condition for the Θ_2^{p}-completeness of Unfrozenness, namely the existence of a polynomial-time computable construction that turns arbitrary graphs into unfrozen ones without changing their chromatic number in an intractable way.

Theorem 25. *Assume that there are polynomial-time computable functions $f : \mathcal{G} \to \mathcal{G}$ and $g : \mathcal{G} \to \mathbb{Z}$ such that for any graph G we have that $f(G)$ is unfrozen and $\chi(f(G)) = \chi(G) + g(G)$. Then* Unfrozenness *is Θ_2^{p}-complete.*

Proof. As usual, membership in Θ_2^{p} is immediate. For the lower bound, assume the existence of functions f and g as described in the theorem. We reduce from $\{(G, H) \mid \chi(G) \leq \chi(H)\}$, which is Θ_2^{p}-hard, as to be expected from this type of comparison problem. This hardness will be formally stated in Theorem 26 only later on because its proof both relies on and well illustrates the application of a general criterion we are going formulate in Lemma 29.

Now, let two graphs G and H be given. Our intermediate goal is to construct two graphs G'' and H'' such that only the latter is unfrozen and

$$\chi(G) \leq \chi(H) \iff \chi(G'') < \chi(H'').$$

Let $G' = G + I_2$ and $H' = f(H)$. Note that H' is unfrozen with $\chi(H') = \chi(H) + g(H)$, while G' is not unfrozen—due to the frozen nonedge that can be added to I_2—and $\chi(G') = \chi(G)+1$. Let $G'' = G' + K_{\max\{0,g(H)-1\}}$ and $H'' = H' + K_{1+\max\{1-g(H),0\}}$. Then H'' is unfrozen with

$$\chi(H'') = \chi(H) + g(H) + 1 + \max\{1 - g(H), 0\} = \chi(H) + \max\{1, g(H)\} + 1,$$

while G'' has a frozen nonedge with

$$\chi(G'') = \chi(G) + 1 + \max\{0, g(H) - 1\} = \chi(G) + \max\{1, g(H)\}.$$

Moreover, we have $\chi(G'') - \chi(G) = \max\{1, g(H)\} = \chi(H'') - \chi(H) - 1$ and hence $\chi(G) \leq \chi(H) \iff \chi(G'') < \chi(H'')$, as desired.

Now let $U = G'' \cup H''$ be the disjoint union of the two constructed graphs. We clearly have $\chi(U) = \max\{\chi(G''), \chi(H'')\}$. Moreover, every nonedge between an arbitrary vertex $v \in V(G'')$ and an arbitrary vertex $w \in V(H'')$ is unfrozen: Let an arbitrary optimal coloring of $U = G'' \cup H''$ be given. If it assigns v and w different colors, we are done; otherwise, swap the two distinct colors of v and w for all vertices in $V(G'')$. Recalling that H'' is unfrozen, we can conclude that U is unfrozen if and only if all nonedges $e \in \overline{E}(G'') = \overline{E}(G')$ are unfrozen.

Assume first that $\chi(G) \leq \chi(H)$; that is, $\chi(G'') < \chi(H'')$. For every $e \in \overline{E}(G'')$, we then have

$$\chi(U + e) = \chi((G'' + e) \cup H'') \leq \max\{\chi(G'') + 1, \chi(H'')\} = \chi(H'') = \chi(U).$$

Therefore, U is unfrozen. Assume now that $\chi(G) > \chi(H)$; that is, $\chi(G'') \geq \chi(H'')$. Recall that G'' has a frozen nonedge e'. It follows that $\chi(G'' + e') = \chi(G'') + 1 > \chi(H'')$ and thus

$$\chi(U + e') = \max\{\chi(G'' + e'), \chi(H'')\} = \chi(G'') + 1 > \max\{\chi(G''), \chi(H'')\} = \chi(U).$$

Therefore, e' is also a frozen nonedge of U, which is thus not unfrozen. This concludes the proof. □

Note that an analogue of Lemma 18 for unfreezing instead of stabilizing edges would be sufficient to satisfy the assumption of Theorem 25. However, based on our efforts we suspect that a suitable gadget—if one exists—must be of significantly higher complexity than the one in Figure 2.

It remains to show the Θ_2^{p}-hardness of the problem from which we reduced in the proof of Theorem 25.

Theorem 26. CompareColorability $= \{(G, H) \in \mathcal{G}^2 \mid \chi(G) \leq \chi(H)\}$ *is* Θ_2^{p}*-hard.*

Theorem 26 is proved essentially in the same way as Wagner [40, Theorem 6.3.2] proves the Θ_2^{p}-hardness of OddColorability. As he suggests [40, page 79], it is rather straightforward to translate the hardness result for OddColorability into one for EqualColorability. This holds true for CompareColorability as well. In the remainder of this section, we generalize the method for obtaining these results to yield two sufficient criteria for Θ_2^{p}-hardness, stated as Lemmas 28 and 29 below. We then use the latter lemma to prove Theorem 26.

Our two sufficient criteria for Θ_2^{p}-hardness, Lemmas 28 and 29, are both consequences of Wagner's criterion [40, Theorem 5.2], which we state in Lemma 27. We assume without loss of generality that all problems are encoded over the same finite alphabet Σ.

Lemma 27 *(By Wagner [40, Theorem 5.2]). A problem A is Θ_2^{p}-hard if the following condition is satisfied:*

There are an NP*-complete problem D and a polynomial-time computable function $f\colon \bigcup_{k=1}^{\infty}(\Sigma^*)^{2k} \to \Sigma^*$ such that for every $k \in \mathbb{N} - \{0\}$ and for all $x_1, \ldots, x_{2k} \in \Sigma^*$ with $x_1 \in D \Leftarrow \cdots \Leftarrow x_{2k} \in D$ we have*

$$f(x_1, \ldots, x_{2k}) \in A \iff \|\{x_1, \ldots, x_{2k}\} \cap D\| \text{ is odd}.$$

The following two lemmas are identical, except for the last line, where once we have an equality and once a nonstrict inequality.

Lemma 28. *A problem A is Θ_2^{p}-hard if the following condition is satisfied:*

There are NP*-complete problems D_1 and D_2 and a polynomial-time computable function $g\colon \bigcup_{k=1}^{\infty}(\Sigma^*)^{2k} \to \Sigma^*$ such that for every $k \in \mathbb{N} - \{0\}$ and for all $y_1, \ldots, y_k, z_1, \ldots, z_k \in \Sigma^*$ with $y_1 \in D_1 \Leftarrow \cdots \Leftarrow y_k \in D_1$ and $z_1 \in D_2 \Leftarrow \cdots \Leftarrow z_k \in D_2$ we have*

$$g(y_1, \ldots, y_k, z_1, \ldots, z_k) \in A \iff \|\{y_1, \ldots, y_k\} \cap D_1\| = \|\{z_1, \ldots, z_k\} \cap D_2\|.$$

Lemma 29. *A problem A is Θ_2^{p}-hard if the following condition is satisfied:*

There are NP*-complete problems D_1 and D_2 and a polynomial-time computable function $g\colon \bigcup_{k=1}^{\infty}(\Sigma^*)^{2k} \to \Sigma^*$ such that for every $k \in \mathbb{N} - \{0\}$ and for all $y_1, \ldots, y_k, z_1, \ldots, z_k \in \Sigma^*$ with $y_1 \in D_1 \Leftarrow \cdots \Leftarrow y_k \in D_1$ and $z_1 \in D_2 \Leftarrow \cdots \Leftarrow z_k \in D_2$ we have*

$$g(y_1, \ldots, y_k, z_1, \ldots, z_k) \in A \iff \|\{y_1, \ldots, y_k\} \cap D_1\| \le \|\{z_1, \ldots, z_k\} \cap D_2\|.$$

We point out that Lemma 29 is the corrected version of a slightly flawed lemma statement in the paper by Spakowski and Vogel [38, Lemma 9]. Since said paper does not apply the problematic lemma anywhere, all other results derived in it remain valid.

Proof of Lemmas 28 and 29. It suffices to make the following five observations.

First, Θ_2^{p} is closed under complementation since we can just invert the output of any algorithm witnessing the membership in $\mathrm{P}_{\|}^{\mathrm{NP}}$. Therefore, Lemma 27 remains true if we replace, on the one hand, only the second occurrence of the language A by its complement $\bar{A} = \Sigma^* - A$ and, on the other hand, "odd" by "even."

Second, within the modified version of Lemma 27 resulting from the two replacements just described, the following three conditions are equivalent due to the assumption $x_1 \in D \Leftarrow x_2 \in D \Leftarrow \cdots \Leftarrow x_{2k} \in D$.

1. $\|\{x_1, \ldots, x_{2k}\} \cap D\|$ is even,
2. $\|\{x_1, x_3, \ldots, x_{2k-1}\} \cap D\| = \|\{x_2, x_4 \ldots, x_{2k}\} \cap D\|$, and
3. $\|\{x_1, x_3, \ldots, x_{2k-1}\} \cap D\| \le \|\{x_2, x_4 \ldots, x_{2k}\} \cap D\|$.

Third, given two arbitrary NP-complete problems D_1 and D_2, there are polynomial-time many-one reductions h_1 and h_2 from D to D_1 and D_2, respectively. Letting

$$y_1 = h_1(x_1),\ y_2 = h_1(x_3),\ \ldots,\ h_1(x_{2k-1}) \text{ and}$$
$$z_1 = h_2(x_2),\ z_2 = h_2(x_4),\ \ldots,\ h_2(x_{2k}),$$

we have

$$\|\{x_1, x_3, \ldots, x_{2k-1}\} \cap D\| = \|\{y_1, y_2, \ldots, y_k\} \cap D_1\| \text{ and}$$
$$\|\{x_2, x_4, \ldots, x_{2k}\} \cap D\| = \|\{z_1, z_2, \ldots, z_k\} \cap D_2\|.$$

Fourth, given a polynomial-time computable function $g\colon \bigcup_{k=1}^{\infty} (\Sigma^*)^{2k} \to \Sigma^*$, we obtain another such function f that satisfies

$$f(x_1, \ldots, x_{2k}) \in A \iff g(y_1, \ldots, y_k, z_1, \ldots, z_k) \in A$$

by simply defining $f(x_1, \ldots, x_{2k}) = g(y_1, \ldots, y_k, z_1, \ldots, z_k)$.

Finally, $x_1 \in D \Leftarrow \cdots \Leftarrow x_{2k} \in D$ implies both

$$x_1 \in D \Leftarrow x_3 \in D \Leftarrow \cdots \Leftarrow x_{2k-1} \in D \text{ and } x_2 \in D \Leftarrow x_4 \in D \Leftarrow \cdots \Leftarrow x_{2k} \in D,$$

which in turn implies

$$y_1 \in D_1 \Leftarrow \cdots \Leftarrow y_k \in D_1 \text{ and } z_1 \in D_2 \Leftarrow \cdots \Leftarrow z_k \in D_2. \quad \square$$

Lemma 28 provides for several equality problems the proofs of Θ_2^{p}-hardness (and thus Θ_2^{p}-completeness), which Wagner asserted [40, page 79] without spelling out the straightforward proofs. In particular, EqualIndependentSet, EqualVertexCover, EqualColorability, and EqualClique—which ask whether two graphs have the same graph number α, β, χ, and ω, respectively—and EqualMaxSat—which asks whether two formulas in 3CNF have the same maximal number of simultaneously satisfiable clauses—are all Θ_2^{p}-complete. This is seen by applying Lemma 28 to the proofs of the corresponding theorems by Wagner [40, Theorems 6.1, 6.2, and 6.3].

By applying Lemma 29 instead, we immediately obtain Θ_2^{p}-completeness for the comparison problems CompareIndependentSet, CompareVertexCover, CompareColorability, CompareClique, and CompareMaxSat.

For all but CompareColorability, Θ_2^{p}-hardness was also proved by Spakowski and Vogel [38, Theorems 2, 12 and 13]. We now show how to apply concretely Lemma 29 to obtain the Θ_2^{p}-hardness of CompareColorability.

Proof of Theorem 26. This proof is modeled after the one for Wagner's Theorem 6.3 [40]. We apply Lemma 29 with $A =$ CompareColorability and $D_1 = D_2 =$ 3Sat.

Let $k \in \mathbb{N} - \{0\}$ and let $2k$ formulas $\Phi_1, \ldots, \Phi_k, \Psi_1, \ldots, \Psi_k \in$ 3CNF satisfying

$$\Phi_1 \in 3\textsc{Sat} \Leftarrow \cdots \Leftarrow \Phi_k \in 3\textsc{Sat} \text{ and}$$
$$\Psi_1 \in 3\textsc{Sat} \Leftarrow \cdots \Leftarrow \Psi_k \in 3\textsc{Sat}$$

be given. Denote by h the standard reduction from 3SAT to 3COLORABILITY by Garey et al. [17]; it maps a formula to a graph whose chromatic number is 3 if the formula is satisfiable and 4 otherwise. Moreover, let $G = h(\Phi_1) + \cdots + h(\Phi_k)$ and $H = h(\Psi_1) + \cdots + h(\Psi_k)$, where $+$ denotes the graph join. It follows that

$$\chi(G) = \sum_{i=1}^{k} \chi(h(\Phi_i)) = 4k - \|\{\Phi_1, \ldots, \Phi_k\} \cap 3\mathrm{SAT}\| \text{ and}$$
$$\chi(H) = \sum_{i=1}^{k} \chi(h(\Psi_i)) = 4k - \|\{\Psi_1, \ldots, \Psi_k\} \cap 3\mathrm{SAT}\|.$$

Thus we have $\chi(H) \le \chi(G)$ if and only if

$$\|\{\Phi_1, \ldots, \Phi_k\} \cap 3\mathrm{SAT}\| \le \|\{\Psi_1, \ldots, \Psi_k\} \cap 3\mathrm{SAT}\|.$$

The map $g\colon (\Phi_1, \ldots, \Phi_k, \Psi_1, \ldots, \Psi_k) \mapsto (H, G)$ therefore satisfies all requirements of Lemma 29, which concludes the proof. □

6. Two-Way Stability

A graph is two-way stable if it is stable with respect to both the deletion and addition of an edge. First, we note that the analogous problem with respect to vertices is trivial for both Colorability and Vertex Cover. The following is an immediate consequence of Theorem 23.

Theorem 30. *There is no vertex-two-way-stable graph and only one β-vertex-two-way-stable graph, namely the null graph with the empty vertex set.*

The default case of edge deletion is more interesting. We begin with Colorability.

Theorem 31. *Assume that there are polynomial-time computable functions $f\colon \mathcal{G} \to \mathcal{G}$ and $g\colon \mathcal{G} \to \mathbb{Z}$ such that for any graph G we have that $f(G)$ is unfrozen and $\chi(f(G)) = \chi(G) + g(G)$. Then the problem* TWOWAYSTABILITY *is Θ_2^{p}-complete.*

Proof. We have TWOWAYSTABILITY = STABILITY ∩ UNFROZENNESS. The membership in Θ_2^{p} is immediate. For Θ_2^{p}-hardness, we show that the map $f(G) = G \cup G$ is a reduction from UNFROZENNESS, which is Θ_2^{p}-hard by Theorem 25 under the stated assumptions. First, $G \cup G$ is stable for any given graph G since $\chi(G_1 \cup G_2) = \max\{\chi(G_1), \chi(G_2)\}$ for all graphs $G_1, G_2 \in \mathcal{G}$. We conclude that $G \cup G$ is two-way-stable if and only if it is unfrozen. Moreover, $G \cup G$ is unfrozen if and only if G is unfrozen: A nonedge $e \in \overline{E}(G)$ is unfrozen in G exactly if it is unfrozen in $G \cup G$, again due to $\chi(G_1 \cup G_2) = \max\{\chi(G_1), \chi(G_2)\}$. It remains to examine the nonedges that can be added to $G \cup G$ between the two copies of G. Let $\{v_1, v_2\}$ be such a nonedge. We prove that it is unfrozen. Without loss of generality, assume that G is nonempty, that is, $\chi(G) > 1$. Given an optimal coloring for G, we obtain an optimal coloring for $G \cup G + \{v_1, v_2\}$ by coloring both copies according to the given coloring, just with the colors permuted appropriately for the second copy, that is, such that v_2 receives a color different from the one of v_1. □

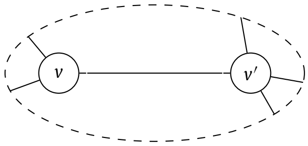
We are able to prove the analogous result for β-TWOWAYSTABILITY via Lemma 32.

Lemma 32. *Let a nonempty graph G and an edge $e = \{v, v'\} \in V(G)$ be given. Construct from G a graph G' by adding to e the gadget consisting of a clique on the new vertex set $Q = \{u_1, u_2, u_3, u_4, u_1', u_2', u_3', u_4'\}$, with the four edges $\{u_i, u_i'\}$ for $i \in \{1, 2, 3, 4\}$ removed and the four edges $\{v, u_1\}$, $\{v, u_2\}$, $\{v', u_3\}$, and $\{v', u_4\}$ added. (This gadget is displayed in Figure 5b.) The graph G' has the following properties.*

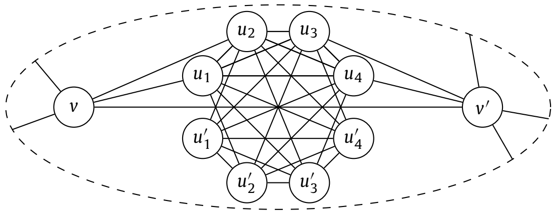
1. *$\beta(G') = \beta(G) + 6$,*
2. *every edge $e' \in E(G) - \{e\}$ is β-stable in G exactly if it is in G',*
3. *all remaining edges of G' are β-stable,*
4. *every nonedge $e' \in \overline{E}(G)$ is β-unfrozen in G exactly if it is in G', and*
5. *all remaining nonedges $e' \in \overline{E}(G') - \overline{E}(G)$ of G' are β-unfrozen.*

Proof. We prove that G' has the required properties.

1. Let X be a vertex cover of G. It must contain v or v'. If $v \in X$, then it follows that $X \cup \{u_2, u_3, u_4, u_2', u_3', u_4'\}$ is a vertex cover of G'; if $v' \in X$, then $X \cup \{u_1, u_2, u_3, u_1', u_2', u_3'\}$ is one. This proves $\beta(G') \le \beta(G) + 6$. To obtain the inverse



(a) An edge to be stabilized.



(b) The same section after adding the stabilization gadget.

Fig. 5. How to stabilize an arbitrary edge $\{v, v'\}$ without introducing new unfrozen edges.

inequality, let X' be a vertex cover of G'. Then $X' - Q$ is a vertex cover of G. Moreover, for any vertex $w \in Q$, we have that, if $w \notin X'$, then X' must contain the entire neighborhood of w, which contains exactly six vertices from Q. It follows that

$$\beta(G) \leq \|X' - Q\| \leq \|X'\| - 6 \leq \beta(G') - 6.$$

2. This is a consequence of the first property since, for every edge $e' \in E(G) - \{e\}$, our construction clearly commutes with deleting e'.
3. Let $e' = e$ or $e' \in E(G') - E(G)$. We need to show that $\beta(G' - e') \geq \beta(G) + 6$. We denote the induced graph of $G' - e'$ on the eight vertices Q by $(G' - e')[Q]$.
   Assume first that $e' \neq e$. This implies that deleting the vertices Q from $G' - e$ yields exactly the original graph G. Therefore, it suffices to show that any vertex cover of $(G' - e')[Q]$ contains at least 6 vertices, which is the same as saying that any independent set of $(G' - e')[Q]$ contains at most 2 vertices, which is in turn equivalent to proving that the complement graph of $(G' - e')[Q]$ contains no clique of size 3. This is obvious since this complement of $(G' - e')[Q]$ consists of the four edges $\{u_1, u_1'\}$, $\{u_2, u_2'\}$, $\{u_3, u_3'\}$, and $\{u_4, u_4'\}$ plus potentially the edge e', which can at most connect two of these otherwise disjoint edges but never complete a triangle.
   Assume now that $e' = e$. Our observation from the previous paragraph is still sufficient in the case of a vertex cover of $G' - e$ that contains v or v' since removing Q still yields a vertex cover of the original graph G in this case. Only if neither v nor v' is part of the considered vertex cover of $G' - e'$, then we have to show that $(G' - e')[Q]$ is guaranteed to contain 7 vertices instead of only 6. This is easy to see since u_1, u_2, u_3, and u_4 are required to cover the edges leading from them to v and v' and three more vertices are necessary to cover the edges of the 4-clique on u_1', u_2', u_3', and u_4'.
4. The argument for the second property is valid for nonedges $e' \in \overline{E}(G)$ as well.
5. Let $e' \in \overline{E}(G') - \overline{E}(G)$ and let X be a vertex cover of G. We show how to obtain a vertex cover for $G' + e'$ by adding six vertices to X. At least one endpoint of e lies in Q, call it w. If $w \in \{u_1, u_1', u_4, u_4'\}$, let $X' = X \cup \{u_1, u_1', u_4, u_4'\}$; otherwise, let $X' = X \cup \{u_2, u_2', u_3, u_3'\}$. Let $X'' = X' \cup \{u_3, u_3', u_4, u_4'\}$ if $v \in X$. Otherwise, we have $v' \in X$ and let $X'' = X' \cup \{u_1, u_1', u_2, u_2'\}$. It is easy to check that X'' is a vertex cover of $G' + e'$ and $\|X''\| = \|X\| + 6$ in all cases.

This concludes the proof. □

An iterated application of Lemma 32 allows us to stabilize an arbitrary set of edges of an arbitrary graph without introducing any new unfrozen nonedges. The Θ_2^{p}-hardness of β-TwoWayStability is now an easy consequence of Lemma 32.

Theorem 33. *The problem β-TwoWayStability is Θ_2^{p}-complete.*

Proof. The upper bound is immediate. We now give a polynomial-time many-one reduction from β-Unfrozenness, which is Θ_2^{p}-hard by Theorem 24, to β-TwoWayStability. For given G, we replace each edge $e \in E(G)$ by the gadget displayed in Figure 5b and call the resulting graph $\widehat{G}$. This is possible in polynomial time because the gadget has constant size. By an iterated application of Lemma 32, all new edges in the resulting graph $\widehat{G}$ are β-stable and each pre-existing edge $e \in E(G)$ is β-unfrozen in $\widehat{G}$ if and only if it was β-unfrozen in G. Thus $\widehat{G}$ is β-two-way-stable if and only if G is β-unfrozen. □

7. Connections between Clique, Vertex Cover, and Independent Set

We conclude our investigations by examining the relations between the three problems of Clique, Vertex Cover, and Independent Set. As is to be expected, they are so closely related that almost all stability results for one of them carry over to the other two in a straightforward way.

Proposition 34. *Let $\overline{G}$ denote the complement graph of G. We have the following equalities.*

1. β-Stability $= \alpha$-Stability $= \{\overline{G} \mid G \in \omega\text{-Unfrozenness}\}$.
2. β-Unfrozenness $= \alpha$-Unfrozenness $= \{\overline{G} \mid G \in \omega\text{-Stability}\}$.
3. β-TwoWayStability $= \alpha$-TwoWayStability $= \{\overline{G} \mid G \in \omega\text{-TwoWayStability}\}$.
4. β-VertexStability $= \{I_n \mid n \in \mathbb{N}\}$.
5. α-VertexStability $= \{\overline{G} \mid G \in \omega\text{-VertexStability}\}$.
6. β-VertexUnfrozenness $= \beta$-VertexTwoWayStability $= \{K_0\}$.
7. α-VertexUnfrozenness $= \alpha$-VertexTwoWayStability $= \omega$-VertexUnfrozenness $= \omega$-VertexTwoWayStability $= \emptyset$.

Proof. For the second equality of the first three items it suffices to note that an independent set of a graph is a clique of its complement graph and vice versa. The first equality of the first three items follows from the fact that, on the one hand, for any graph on n vertices, the complement of a vertex cover of size k is an independent set of size $n-k$ and, on the other hand, adding or deleting edges obviously does not change the number of vertices. For the remaining items, we add or delete vertices, so this argument does not hold anymore. Item 4 is exactly Theorem 20. For item 5, we simply use that a clique is an independent set in the complement graph and vice versa. Item 6 combines Theorems 23 and 30. Item 7 finally follows from the fact that adding an isolated vertex increases α, while adding a universal vertex increases ω. $\square$

An interesting inversion in this pattern occurs for the vertex deletion case. Here, switching from β to α or ω in fact flips the stability problem to the criticality version and vice versa.

Proposition 35. *We have the following equalities.*

1. β-VertexStability $= \alpha$-VertexCriticality $= \{\overline{G} \mid G \in \omega\text{-VertexCriticality}\}$.
2. β-VertexCriticality $= \alpha$-VertexStability $= \{\overline{G} \mid G \in \omega\text{-VertexStability}\}$.

Proof. For the second part of the claim, it suffices to prove that a vertex v of a graph G is α-stable if and only if it is β-critical. For the first part of the claim we then only need to recall that any vertex must be either α-stable or α-critical and likewise either β-stable or β-critical by the definition of these notions. In the fourth step of the following equivalence chain we use that every minimum vertex cover is the complement of a maximum independent set and vice versa.

$$\begin{aligned}
v \text{ is } \alpha\text{-stable} &\iff \alpha(G-v) = \alpha(G)\\
&\iff \|V(G)\| - \alpha(G-v) = \|V(G)\| - \alpha(G)\\
&\iff \|V(G-v)\| - \alpha(G-v) + 1 = \|V(G)\| - \alpha(G)\\
&\iff \beta(G-v) + 1 = \beta(G)\\
&\iff \beta(G-v) \neq \beta(G)\\
&\iff v \text{ is } \beta\text{-critical.} \quad \square
\end{aligned}$$

Using Proposition 35, we directly obtain from the Θ_2^{p}-hardness of β-VertexCriticality [5] the same for α-VertexStability and, by complementing the graphs, ω-VertexStability. We note that these are the only nontrivial results revealed by the connection between stability and criticality.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors are grateful to the anonymous reviewers for their careful reading of our paper, resulting in many useful comments and suggestions.

References

[1] Douglas Bauer, Frank Harary, Juhani Nieminen, Charles L. Suffel, Domination alteration sets in graphs, Discrete Math. 47 (1983) 153–161.
[2] Adam Beacham, Joseph C. Culberson, On the complexity of unfrozen problems, Discrete Appl. Math. 153 (1–3) (2005) 3–24.
[3] Béla Bollobás, Modern Graph Theory, Springer, 1998.
[4] Béla Bollobás, Extremal Graph Theory, London Mathematical Society Monographs, Oxford University Press, London, 1978.
[5] Elisabet Burjons, Fabian Frei, Edith Hemaspaandra, Dennis Komm, David Wehner, Finding optimal solutions with neighborly help, in: Proceedings of the 44th International Symposium on Mathematical Foundations of Computer Science (MFCS 2019), in: Leibniz International Proceedings in Informatics (LIPIcs), vol. 138, Schloss Dagstuhl – Leibniz Center for Informatics, 2019, pp. 78:1–78:14.

[6] Jin-Yi Cai, Gabriele E. Meyer, Graph minimal uncolorability is D^P-complete, SIAM J. Comput. 16 (2) (1987) 259–277.
[7] Gregory J. Chaitin, Register allocation and spilling via graph coloring, in: ACM SIGPLAN Notices – Proceedings of the 1982 SIGPLAN Symposium on Compiler Construction (CC 1982), SIGPLAN Not. 17 (6) (1982) 98–101.
[8] Richard Chang, Jim Kadin, On computing Boolean connectives of characteristic functions, Math. Syst. Theory 28 (3) (1995) 173–198.
[9] Richard Chang, Jim Kadin, The Boolean hierarchy and the polynomial hierarchy: a closer connection, SIAM J. Comput. 25 (2) (1996) 340–354.
[10] Kamalika Chaudhuri, Fan Chung, Mohammad S. Jamall, A network coloring game, in: Proceedings of the 4th International Workshop on Internet and Network Economics (WINE 2008), in: Lecture Notes in Computer Science (LNCS), vol. 5385, Springer, December 2008, pp. 522–530.
[11] Guantao Chen, Guangming Jing, Structural properties of edge-chromatic critical multigraphs, J. Comb. Theory, Ser. B 139 (2019) 128–162.
[12] Wyatt J. Desormeaux, Teresa W. Haynes, Michael A. Henning, Total domination critical and stable graphs upon edge removal, Discrete Appl. Math. 158 (15) (2010) 1587–1592.
[13] Gabriel A. Dirac, Some theorems on abstract graphs, Proc. Lond. Math. Soc. s3-2 (1) (1952) 69–81.
[14] Paul Erdős, Tibor Gallai, On the minimal number of vertices representing the edges of a graph, Magyar Tud. Akad. Mat. Kutató Int. Közl. 6 (1–2) (1961) 181–203.
[15] Martin G. Everett, Steve Borgatti, Role colouring a graph, Math. Soc. Sci. 21 (2) (1991) 183–188.
[16] Fabian Frei, Edith Hemaspaandra, Jörg Rothe, Complexity of stability, in: Yixin Cao, Siu-Wing Cheng, Minming Li (Eds.), Proceedings of the 31st International Symposium on Algorithms and Computation (ISAAC 2020), in: Leibniz International Proceedings in Informatics (LIPIcs), vol. 181, Schloss Dagstuhl – Leibniz Center for Informatics, 2020, pp. 19:1–19:14.
[17] Michael R. Garey, David S. Johnson, Larry J. Stockmeyer, Some simplified NP-complete graph problems, Theor. Comput. Sci. 1 (1976) 237–267.
[18] Georg Gunther, Bert Hartnell, Douglas F. Rall, Graphs whose vertex independence number is unaffected by single edge addition or deletion, Discrete Appl. Math. 46 (1993) 167–172.
[19] György Hajós, Über eine Konstruktion nicht n-färbbarer Graphen, Wiss. Z., Martin-Luther-Univ. Halle-Wittenb. 10 (1961) 116–117.
[20] Frank Harary, Graph Theory, Addison-Wesley, 1969.
[21] Frank Harary, Carsten Thomassen, Anticritical graphs, Math. Proc. Camb. Philos. Soc. 79 (1976) 11–18.
[22] Teresa W. Haynes, Robert C. Brigham, Ronald D. Dutton, Extremal graphs domination insensitive to the removal of k edges, Discrete Appl. Math. 44 (1–3) (1993) 295–304.
[23] Lane A. Hemachandra, The strong exponential hierarchy collapses, J. Comput. Syst. Sci. 39 (3) (1989) 299–322.
[24] Edith Hemaspaandra, Lane A. Hemaspaandra, Jörg Rothe, Exact analysis of Dodgson elections: Lewis Carroll's 1876 voting system is complete for parallel access to NP, J. ACM 44 (6) (1997) 806–825.
[25] Edith Hemaspaandra, Lane A. Hemaspaandra, Jörg Rothe, The complexity of online manipulation of sequential elections, J. Comput. Syst. Sci. 80 (4) (2014) 697–710.
[26] Edith Hemaspaandra, Holger Spakowski, Jörg Vogel, The complexity of Kemeny elections, Theor. Comput. Sci. 349 (3) (2005) 382–391.
[27] Michael A. Henning, Marcin Krzywkowski, Total domination stability in graphs, Discrete Appl. Math. 236 (2018) 246–255.
[28] Matthew O. Jackson, Social and Economic Networks, Princeton University Press, 2008.
[29] Michael Kearns, Siddharth Suri, Nick Montfort, An experimental study of the coloring problem on human subject networks, Science 313 (5788) (2006) 824–827.
[30] Susan Khor, Application of graph coloring to biological networks, IET Syst. Biol. 4 (3) (2010) 185–192.
[31] Frank T. Leighton, A graph coloring algorithm for large scheduling problems, J. Res. Natl. Bur. Stand. 84 (6) (1979) 489–506.
[32] Rémi Monassen, Riccardo Zecchina, Statistical mechanics of the random k-satisfiability model, Bull. Am. Phys. Soc. 56 (2) (1997) 1357–1370.
[33] Rémi Monassen, Riccardo Zecchina, Scott Kirkpatrick, Bart Selman, Lidror Troyansky, Determining computational complexity from characteristic 'phase transitions', Nature 400 (1998) 133–137.
[34] Christos H. Papadimitriou, David Wolfe, The complexity of facets resolved, J. Comput. Syst. Sci. 37 (1) (1988) 2–13.
[35] Christos H. Papadimitriou, Mihalis Yannakakis, The complexity of facets (and some facets of complexity), J. Comput. Syst. Sci. 28 (2) (1984) 244–259.
[36] Jörg Rothe, Exact complexity of exact-four-colorability, Inf. Process. Lett. 87 (1) (2003) 7–12.
[37] Jörg Rothe, Holger Spakowski, Jörg Vogel, Exact complexity of the winner problem for Young elections, Theory Comput. Syst. 36 (4) (2003) 375–386.
[38] Holger Spakowski, Jörg Vogel, θ_2^p-completeness: A classical approach for new results, in: Proceedings of the 20th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2000), in: Lecture Notes in Computer Science (LNCS), vol. 1974, Springer, 2000, pp. 348–360.
[39] Jimeng Sun, Charalampos E. Tsourakakis, Evan Hoke, Christos Faloutsos, Tina Eliassi-Rad, Two heads better than one: pattern discovery in time-evolving multi-aspect data, Data Min. Knowl. Discov. 17 (1) (2008) 111–128.
[40] Klaus W. Wagner, More complicated questions about maxima and minima, and some closures of NP, Theor. Comput. Sci. 51 (1–2) (1987) 53–80.
[41] Klaus W. Wagner, Bounded query classes, SIAM J. Comput. 19 (5) (1990) 833–846.
[42] Robin Weishaupt, Jörg Rothe, Stability of special graph classes, in: Claudio Sacerdoti Coen, Ivano Salvo (Eds.), Proceedings of the 22nd Italian Conference on Theoretical Computer Science (ICTCS 2022), CEUR Workshop Proceedings. CEUR-WS.org, 2022, in press.
[43] Walter Wessel, Criticity with respect to properties and operations in graph theory, in: László Lovász András Hajnal, Vera T. Sós (Eds.), Finite and Infinite Sets. Proceedings of the Sixth Hungarian Combinatorial Colloquium, in: Colloquia Mathematica Societatis Janos Bolyai, vol. 2, North-Holland, 1984, pp. 829–837.
[44] Gerhard J. Woeginger, Core stability in hedonic coalition formation, in: Proceedings of the 39th International Conference on Current Trends in Theory and Practice of Computer Science, in: Lecture Notes in Computer Science (LNCS), vol. 7741, Springer, 2013, pp. 33–50.