Complexity of counting subgraphs: Only the boundedness of the vertex-cover number counts

Radu Curticapean
Department of Computer Science,
Saarland University
curticapean@cs.uni-sb.de

Dániel Marx
Institute for Computer Science and Control,
Hungarian Academy of Sciences (MTA SZTAKI)
dmarx@cs.bme.hu

***Abstract*—For a class C of graphs, #Sub(C) is the counting problem that, given a graph H from C and an arbitrary graph G, asks for the number of subgraphs of G isomorphic to H. It is known that if C has bounded vertex-cover number (equivalently, the size of the maximum matching in C is bounded), then #Sub(C) is polynomial-time solvable. We complement this result with a corresponding lower bound: if C is *any* recursively enumerable class of graphs with unbounded vertex-cover number, then #Sub(C) is #W[1]-hard parameterized by the size of H and hence not polynomial-time solvable and not even fixed-parameter tractable, unless FPT is equal to #W[1].**

As a first step of the proof, we show that counting k-matchings in bipartite graphs is #W[1]-hard. Recently, Curticapean [ICALP 2013] proved the #W[1]-hardness of counting k-matchings in general graphs; our result strengthens this statement to bipartite graphs with a considerably simpler proof and even shows that, assuming the Exponential Time Hypothesis (ETH), there is no f(k)*n^o(k/log(k)) time algorithm for counting k-matchings in bipartite graphs for any computable function f. As a consequence, we obtain an independent and somewhat simpler proof of the classical result of Flum and Grohe [SICOMP 2004] stating that counting paths of length k is #W[1]-hard, as well as a similar almost-tight ETH-based lower bound on the exponent.

I. Introduction

Counting the number of solutions is often a considerably more difficult task than deciding whether a solution exists or finding a single solution. A classical example is the case of perfect matchings in bipartite graphs: there are well-known polynomial-time algorithms for finding a perfect matching, but the seminal result of Valiant [1] showed that counting the number of perfect matchings in bipartite graphs is #P-hard, and hence unlikely to be polynomial-time solvable. This phenomenon has been systematically analyzed, for example, in the context of Constraint Satisfaction Problems (CSPs), where dichotomy theorems characterizing the polynomial-time solvable and #P-hard cases [2], [3], [4], [5], [6] show that very restrictive conditions are needed to ensure that not only the decision problem is polynomial-time solvable, but the counting problem is as well.

Research of the second author was supported by the European Research Council (ERC) grant PARAMTIGHT (No. 280152) and OTKA grant NK105645.

Our goal in the present paper is to systematically analyze the tractable cases of counting subgraphs. Counting the number of times a certain pattern appears in a graph is a fundamental theoretical problem that has been explored intensively also on real-world large graphs [7], [8], [9], [10], [11]. Formally, given graphs H and G, the task is to count the number of subgraphs of G that are isomorphic to the pattern graph H; we would like to understand which graphs H make this problem easy or hard. However, we have to be careful how we formulate the framework of our investigations. For every fixed pattern graph H, the number of subgraphs of G isomorphic to H can be determined in polynomial-time by brute force: it suffices to check each of the $|V(G)|^{|V(H)|}$ mappings from the vertices of H to the vertices of G, resulting in a simple polynomial-time algorithm for fixed H. There is a line of research devoted to finding nontrivial improvements over brute-force search for specific patterns [12], [13], [14], [15], [16], [17], [18], [19], [20]. Besides improvements for specific small graphs H, these papers identified structural properties, such as boundedness of treewidth, pathwidth, and vertex-cover number, that can give improvements for some infinite classes $\mathcal{H}$ of graphs H. Our goal is to exhaustively characterize which graph properties are sufficiently strong to guarantee polynomial-time solvability.

The search for graph properties that make counting easy can be studied in the following framework. For every class $\mathcal{H}$ of graphs, $\#\mathsf{Sub}(\mathcal{H})$ is the counting problem where, given a graph $H \in \mathcal{H}$ and arbitrary graph G, the task is to count the number of (not necessarily induced) subgraphs of G isomorphic to H. Rather than asking which fixed graphs H make counting easy (as we have seen, the problem is polynomial-time solvable for every fixed H), we ask which *classes* $\mathcal{H}$ of graphs make $\#\mathsf{Sub}(\mathcal{H})$ polynomial-time solvable. As many of the theoretical results and applications involve counting a small fixed pattern graph H in a large graph G, an equally natural question to ask is whether $\#\mathsf{Sub}(\mathcal{H})$ can be solved in time $f(|V(H)|) \cdot n^{O(1)}$ for some computable function f depending only on the size of H. That is, we ask whether $\#\mathsf{Sub}(\mathcal{H})$ for a particular class $\mathcal{H}$ is *fixed-parameter tractable (FPT)* parameterized by $|V(H)|$.

0272-5428/14 $31.00 © 2014 IEEE
DOI 10.1109/FOCS.2014.22

Main result. The vertex-cover number $\tau(H)$ of a graph H is the minimum size of a set of vertices that contains at least one endpoint of every edge. It is well known that if $\nu(H)$ is the size of a maximum matching in G, then $\nu(H) \leq \tau(H) \leq 2\nu(H)$. If the class $\mathcal{H}$ has bounded vertex-cover number (or equivalently the maximum matching size is bounded), then it follows from a result of Vassilevska Williams and Williams [18] that $\#\mathsf{Sub}(\mathcal{H})$ is FPT and it follows from a result of Kowaluk, Lingas, and Lundell [21] that $\#\mathsf{Sub}(\mathcal{H})$ is actually polynomial-time solvable (we also present a simple self-contained argument for the polynomial-time solvability of $\#\mathsf{Sub}(\mathcal{H})$ in the full version). Our main result complements these algorithms by showing that boundedness of the vertex-cover number is the only property of $\mathcal{H}$ that guarantees tractability of $\#\mathsf{Sub}(\mathcal{H})$.

Theorem I.1. *Let $\mathcal{H}$ be a recursively enumerable class of graphs. If* FPT $\neq$ #W[1]*, the following are equivalent:*

1) $\#\mathsf{Sub}(\mathcal{H})$ *is polynomial-time solvable.*
2) $\#\mathsf{Sub}(\mathcal{H})$ *is FPT parameterized by* $|V(H)|$.
3) $\mathcal{H}$ *has bounded vertex-cover number.*

Let us review some results from the literature that are of similar form as Theorem I.1. A result of Grohe, Schwentick, and Segoufin [22] can be interpreted as characterizing the complexity of finding a vertex-colored graph $H \in \mathcal{H}$ in G; they show that the tractability criterion is the boundedness of the treewidth of $\mathcal{H}$. Grohe [23] considered the problem of deciding if there is a homomorphism from a graph $H \in \mathcal{H}$ to G; here the tractability criterion is the boundedness of the treewidth of the core of H. For the problem of counting homomorphisms, Dalmau and Jonsson [24] showed that it is again the boundedness of the treewidth that matters. Chen, Thurley, and Weyer [25] studied the problem of finding induced subgraphs, which is apparently the most difficult of these problems, as the problem is easy only if the class $\mathcal{H}$ contains only graphs of bounded size. In all of these results, similarly to Theorem I.1, polynomial time and fixed-parameter tractability coincide. An example where polynomial time and FPT is not known to be equivalent is the result of Marx [26], which can be interpreted as characterizing the complexity of finding vertex-colored hypergraphs. For this problem, bounded submodular width is the property that guarantees fixed-parameter tractability, but it is not known if it implies polynomial-time solvability.

Very recently, Jerrum and Meeks [27], [28], [29] studied problems related to counting induced subgraphs isomorphic to a given graph H and counting induced subgraphs satisfying certain fixed properties. As these investigations are in the very different setting of induced subgraphs, they are not directly related to our results.

We remark that there have been investigations of finding and counting subgraphs in a framework when the pattern graph H is arbitrary and the host graph G is restricted to a certain class; some of these results appear in the more general context of evaluating first-order logical sentences [30], [31], [32]. Needless to say, these results are very different from our setting.

Complexity of counting k-matchings. The study of the fixed-parameter tractability of counting problems was initiated by Flum and Grohe [33]. Finding paths and cycles of length k is well known to be fixed-parameter tractable [13], [34], [35], [36], but Flum and Grohe [33] proved the surprising result that counting paths and cycles of length k is #W[1]-hard, and hence unlikely to be fixed-parameter tractable. They raised as an open question whether counting k-matchings (i) in general graphs or (ii) in bipartite graphs is fixed-parameter tractable. Very recently, Curticapean [37] (based on earlier work of Bläser and Curticapean [38]) used quite involved algebraic techniques to answer the first question in the negative by showing that counting k-matchings is #W[1]-hard on general graphs. Our proof of Theorem I.1 is based on a reduction from counting k-matchings. In fact, the proof technique requires the stronger result that counting k-matchings is #W[1]-hard even in bipartite graphs. Therefore, in Section III, we prove this stronger result using a proof that relies only on basic linear algebra (the rank of the Kronecker product of matrices) and is significantly simpler than the proof of Curticapean [37]. Our proof also shows the hardness of the "edge-colorful" variant where the edges of G are colored with k colors and we need to count the k-matchings in G where every edge has a different color. An additional benefit of our proof is that, combined with a lower bound of Marx [39] for subgraph isomorphism, it gives an almost-tight lower bound on the exponent of n. The Exponential Time Hypothesis (ETH) of Impagliazzo, Paturi, and Zane [40] implies that n-variable 3SAT cannot be solved in time $2^{o(n)}$. Our result shows that, assuming ETH, the number of k-matchings in a bipartite graph cannot be counted in time $f(k)n^{o(k/\log k)}$ for any computable function f. There are simple reductions from counting k-matchings to counting paths and cycles of length k, thus our proof gives an independent and somewhat simpler proof of the results of Flum and Grohe [33] on counting paths and cycles, together with almost-tight ETH-based lower bounds on the exponent that were not known previously.

Theorem I.2. *The following problems are* #W[1]*-hard and, assuming ETH, cannot be solved in time $f(k) \cdot n^{o(k/\log k)}$ for any computable function f: Counting (directed) paths or cycles of length k, and counting edge-colorful or uncolored k-matchings in bipartite graphs.*

Proof overview. We proceed the following way for general (not necessarily hereditary) classes $\mathcal{H}$. First, if $\mathcal{H}$ has unbounded treewidth, then the arguments underlying the previous work of Grohe, Schwentick, and Segoufin [22], Grohe [23], Dalmau and Jonsson [24], and Chen, Thurley, and Weyer [25] go through (see Section II-B). Essentially, we need two reductions. First, there is a simple reduction from counting cliques to counting colored grids. If $\mathcal{H}$ has unbounded treewidth, then the Excluded Grid Theorem of Robertson and Seymour [41] shows that the graphs in $\mathcal{H}$ have arbitrary large grid minors. Therefore, we can embed the problem of counting colored grids into $\#\mathsf{Sub}(\mathcal{H})$. As these techniques are fairly standard by now, the main part of our proof is handling the case when $\mathcal{H}$ has bounded treewidth. This is the part where we have to deviate from previous results (where bounded treewidth always implied tractability) and have to use the fact that counting k-matchings is hard.

If $\mathcal{H}$ has bounded treewidth, then a Ramsey argument contained in the full version shows that there are graphs in $\mathcal{H}$ containing large induced matchings. Our goal is to use these large induced matchings to reduce counting k-matchings in bipartite graphs to $\#\mathsf{Sub}(\mathcal{H})$. Suppose that there is a graph $H \in \mathcal{H}$ such that $V(H)$ has a partition (X, Y) where $H[Y]$ is a k-matching. By simple inclusion/exclusion arguments, it is sufficient to prove hardness for the more general problem where we count only those subgraphs of G isomorphic to H that contain certain specified vertices/edges of G. This suggests the following reduction: let us extend G to a graph G' by introducing a copy of $H[X]$ fully connected to every original vertex of G and then consider the problem of counting subgraphs of G' isomorphic to H that contain every vertex and edge of this copy of $H[X]$. As $H[Y]$ is a k-matching (that is, attaching to a $H[X]$ a k-matching in a certain way extends it to H), any k-matching of G can be used to extend the copy of $H[X]$ to a subgraph of G' isomorphic to H. It could seem now that the number of subgraphs of G' isomorphic to H and containing $H[X]$ is exactly the number of k-matchings in G.

Unfortunately, this is not true in general due to a seemingly unlikely problem: if we extend $H[X]$ to a copy of H, then it is not necessarily true that the extension forms a k-matching. That is, it is possible that $V(H)$ has another partition (X', Y') such that $H[X']$ is isomorphic to $H[X]$, but $H[Y']$ is not a k-matching. While this can be perhaps considered counterintuitive, there are very simple examples where this can happen. Consider, for example, the graph H on vertices a, b, c, d, where any two vertices are adjacent, except a and d. Now $X = \{a, b\}$ and $Y = \{c, d\}$ is a partition where $H[Y]$ is an edge. Consider now the partition $X' = \{b, c\}$, $Y' = \{a, d\}$. We have $H[X] \simeq H[X']$, but $H[Y']$ contains two independent vertices.

Our goal is to find graphs $H \in \mathcal{H}$ and partitions (X, Y) where the problem described in the previous paragraph does not occur. We say that $H \in \mathcal{H}$ and a partition (X, Y) is a *k-matching gadget* if $H[Y]$ is a k-matching, and whenever (X', Y') is a partition of $V(H)$ such that $H[X] \simeq H[X']$ and $H[Y']$ satisfies some technical conditions that we can enforce in the reduction (such as $H[Y']$ is bipartite and has no isolated vertices), then $H[Y']$ is also a k-matching. If the class $\mathcal{H}$ has such k-matching gadgets for every $k \geq 1$, then we can reduce counting k-matchings to $\#\mathsf{Sub}(\mathcal{H})$ with a reduction similar to what was sketched in the previous paragraph (Section IV). We prove the existence of k-matching gadgets in $\mathcal{H}$ by a detailed graph-theoretic study, where we first consider bounded-degree graphs (Section V), then move on to graphs that have unbounded degree, but do not contain large subdivided stars (Section VI), and then finally consider graphs where only the treewidth is bounded (Section VII). Together with the hardness proof for classes with unbounded treewidth (Section II-B) and an algorithm for bounded vertex-cover number, this completes the proof of Theorem I.1.

The full version of this paper, containing all omitted proofs, is available at http://arxiv.org/abs/1407.2929

II. Preliminaries

We sometimes write $\#A := |A|$ for sets A. For $\ell \in \mathbb{N}$, let $(x)_\ell := (x)(x-1)\dots(x-\ell+1)$ denote the falling factorial. Graphs are undirected, unweighted and simple, unless stated otherwise. We write $H \simeq H'$ if H and H' are isomorphic.

The graph H is a *minor* of G, written $H \preceq G$, if H can be obtained from G by edge/vertex-deletions and edge-contractions. The contraction of an edge $uv \in E(G)$ identifies $u, v \in V(G)$ to a single vertex adjacent to the union of the neighborhoods of u and v in G.

Definition II.1. A *tree decomposition* of a graph G is a pair $(T, \mathcal{B})$ in which T is a tree and $\mathcal{B} = \{B_t \mid t \in V(T)\}$ is a family of subsets of $V(G)$ such that (i) $\bigcup_{t \in V(T)} B_i = V$, and (ii) for each edge $e = uv \in E(G)$, there exists a $t \in V(T)$ such that both u and v belong to B_t, and (iii) the set of nodes $\{t \in V(T) \mid v \in B_t\}$ forms a connected subtree of T for every $v \in V(G)$.

We call vertices of T *nodes* and their corresponding B_i's *bags*. The *width* of the tree decomposition is the maximum size of a bag in $\mathcal{B}$ minus 1. The *treewidth* of a graph G, denoted by $\mathrm{tw}(G)$, is the minimum width over all possible tree decompositions of G.

In this paper, parameterized problems are problems that ask for some output on input (x, k), where x is an instance and $k \in \mathbb{N}$ is a parameter. A problem is *fixed-parameter tractable* (FPT) if it admits an algorithm with runtime $f(k)n^{\mathcal{O}(1)}$ for a computable function f. For parameterized problems A, B, we write $A \leq^{\mathrm{T}}_{\mathsf{fpt}} B$ if A admits a parameterized Turing reduction to B, i.e., given oracle access for B, we can solve an instance (x, k) to A in time $f(k)n^{\mathcal{O}(1)}$, calling the oracle only on queries (y, k') with $k' \leq g(k)$. Here, both f and g are computable functions. We write $\leq^{\mathrm{T},\ell}_{\mathsf{fpt}}$ if such a reduction exists with $g \in \mathcal{O}(k)$. It is known that if $A \leq^{\mathrm{T}}_{\mathsf{fpt}} B$ and B is FPT, then it follows that A is FPT as well.

For our purposes, we say that a parameterized problem A is #W[1]-hard if there is a reduction $\#\mathsf{Clique} \leq^{\mathrm{T}}_{\mathsf{fpt}} A$, where $\#\mathsf{Clique}$ is the problem of counting k-cliques in a graph G on input (G, k). It is a standard assumption of parameterized complexity theory that $\mathsf{FPT} \neq \#\mathsf{W}[1]$, parallel to the classical assumption that $\mathsf{P} \neq \#\mathsf{P}$.

Definition II.2. Let $\mathcal{H}$ be a class of graphs, and let H, G be graphs. Let $\mathsf{Sub}(H \to G)$ denote the set of all (not necessarily induced) subgraphs $F \subseteq G$ with $F \simeq H$.

The problem $\#\mathsf{Sub}(\mathcal{H})$ asks, given as input a graph $H \in \mathcal{H}$ and an arbitrary graph G, for the number $\#\mathsf{Sub}(H \to G)$. The parameter in this problem is $|V(H)|$.

In $\#\mathsf{Match}$, we are given a bipartite graph G and $k \in \mathbb{N}$ and ask for $\#\mathsf{Sub}(M_k \to G)$, where M_k denotes the matching of size k, i.e., the 1-regular graph on $2k$ vertices with k edges. The parameter in this problem is k.

In the full version, we present a simple self-contained polynomial-time algorithm for determining $\#\mathsf{Sub}(H \to G)$ in time polynomial in $|V(H)|$ and $|V(G)|$ when the vertex-cover number $\tau(H)$ can be assumed to be constant. As already stated in the introduction, more efficient algorithms are known [18], [21].

Theorem II.3. *Let H be a graph on k vertices with vertex-cover number τ and let G be a graph on n vertices. Then we can compute $\#\mathsf{Sub}(H \to G)$ in time $k^{2^{\mathcal{O}(\tau)}} n^{\tau+\mathcal{O}(1)}$.*

A. Colored graphs

We will sometimes count occurrences of colored graphs H within colored graphs G: Firstly, we will count copies of vertex-colored graphs H within vertex-colored graphs G, where each vertex of H has a distinct color. Secondly, we will count edge-colored matchings in edge-colored graphs.

Definition II.4. Let Γ be a set of colors. A *colored graph* is a graph G together with a coloring $c_G : V(G) \to \Gamma$ or $c_G : E(G) \to \Gamma$. In the first case, we call G vertex-colored, otherwise edge-colored. For $\gamma \in \Gamma$, let $V_\gamma(G)$ denote the set of all γ-colored vertices of G. For $S \subseteq \Gamma$, let $V_S(G) := \bigcup_{\gamma \in S} V_\gamma(G)$. Define E_γ and E_S likewise.

We call G *colorful* if c_G is bijective. In such cases, it will be convenient to identify Γ with $V(G)$ or $E(G)$, depending on whether G is vertex- or edge-colored.

Two Γ-colored graphs H and H' are *color-preserving isomorphic* if there is an isomorphism from H to H' that maps each γ-colored vertex (or edge) of H to a γ-colored vertex (or edge) of H'.

The following counting problems associated with colored graphs will occur in the paper.

Definition II.5. For Γ-vertex-colored graphs H, G with colorful H, let $\mathsf{PartitionedSub}(H \to G)$ denote the set of all subgraphs $F \subseteq G$ such that F is color-preserving isomorphic to H.

Given a class $\mathcal{H}$ of uncolored graphs, the problem $\#\mathsf{PartitionedSub}(\mathcal{H})$ asks for $\#\mathsf{PartitionedSub}(H \to G)$, where H is a Γ-vertex-colorful graph whose underlying uncolored graph is contained in $\mathcal{H}$, and G is Γ-vertex-colored. The parameter in this problem is $|V(H)|$.

For a Γ-edge-colored G and $X \subseteq \Gamma$, let $\mathcal{M}_X[G]$ denote the set of all X-colorful matchings in G, i.e., matchings in G that choose exactly one edge from each color in X. In $\#\mathsf{ColMatch}$, we are given a *bipartite* Γ-edge-colored graph G and $X \subseteq \Gamma$ and ask for $\#\mathcal{M}_X[G]$. The parameter is $|X|$.

Note that $\#\mathsf{PartitionedSub}(\mathcal{H})$ is defined for a class $\mathcal{H}$ for uncolored graphs, while its inputs are vertex-colored graphs.

Remark II.6. Let H, G be Γ-vertex-colored graphs and let F be a subgraph of G that is color-preserving isomorphic to H. If $uv \in E(F)$ is an edge with endpoints of color $\gamma_u, \gamma_v \in \Gamma$, then there is an edge between vertices of colors γ_u, γ_v in $E(H)$. We may therefore assume that, whenever $uv \in E(G)$ is an edge with endpoints of color $\gamma_u, \gamma_v \in \Gamma$ in G, then $\{\gamma_u, \gamma_v\} \in E(H)$. In other words, we may assume that G has edges between two color classes if H has an edge with endpoints of this color, otherwise the edges between the classes are clearly useless.

The principle of inclusion and exclusion will be an important ingredient of reduction between the colored and the uncolored versions of the problems defined above. As a first demonstration of this principle, we obtain a reduction from the colorful problem to the uncolored problem.

Lemma II.7. *For any class $\mathcal{H}$, we have $\#\mathsf{PartitionedSub}(\mathcal{H}) \leq^{\mathrm{T},\ell}_{\mathsf{fpt}} \#\mathsf{Sub}(\mathcal{H})$. Furthermore, $\#\mathsf{ColMatch} \leq^{\mathrm{T},\ell}_{\mathsf{fpt}} \#\mathsf{Match}$.*

B. Unbounded-treewidth graphs

We prove that $\#\mathsf{PartitionedSub}(\mathcal{H})$ is #W[1]-hard whenever $\mathcal{H}$ has unbounded treewidth, i.e., if for every $b \in \mathbb{N}$ some $H \in \mathcal{H}$ has treewidth at least b. By Lemma II.7(1), the same hardness result follows for $\#\mathsf{Sub}(\mathcal{H})$, proving Theorem I.1 for unbounded-treewidth classes.

Theorem II.8. *The problems $\#\mathsf{PartitionedSub}(\mathcal{H})$ and $\#\mathsf{Sub}(\mathcal{H})$ are #W[1]-complete whenever $\mathcal{H}$ is recursively enumerable and has unbounded treewidth.*

As already stated in the introduction, the proof (which is contained in the full version) uses standard techniques and could in fact be adapted from ideas in [22], [23], [24], [25].

On the algorithmic side, it was shown by Arvind and Raman [42, Lemma 1] that $\#\mathsf{PartitionedSub}(H \to G)$ can be computed in time $\mathcal{O}(c^{b^3} k + n^{b+2} 2^{b^2/2})$, where b is the treewidth of H. Therefore, $\#\mathsf{PartitionedSub}(\mathcal{H})$ is polynomial-time solvable if $\mathcal{H}$ has bounded treewidth. Together with our #W[1]-hardness result, this yields a dichotomy for $\#\mathsf{PartitionedSub}(\mathcal{H})$. Note that the algorithm for the bounded-treewidth cases of $\#\mathsf{PartitionedSub}(\mathcal{H})$ does not settle the dichotomy for $\#\mathsf{Sub}$: the reduction in Lemma II.7(1) goes the opposite direction. In fact, there are bounded-treewidth classes $\mathcal{H}$, most notably, matchings and paths, for which $\#\mathsf{PartitionedSub}(\mathcal{H})$ is polynomial-time solvable, but $\#\mathsf{Sub}(\mathcal{H})$ is #W[1]-hard. It is precisely the bounded-treewidth classes where the complexity of the two problems can deviate.

C. Bipartite 3-regular graphs

In Section III, the #W[1]-hardness proof for bipartite k-matching will be by reduction from $\#\mathsf{PartitionedSub}$. It is essential for the hardness proof that the graph H appearing in the $\#\mathsf{PartitionedSub}$ instance is bipartite and 3-regular. Therefore, we establish here the #W[1]-hardness of $\#\mathsf{PartitionedSub}(\mathcal{H}_{\text{bicub}})$, where $\mathcal{H}_{\text{bicub}}$ is the class of all bipartite cubic graphs.

Lemma II.9. $\#\mathsf{PartitionedSub}(\mathcal{H}_{\text{bicub}})$ *is* #W[1]*-hard.*

It is known that, assuming ETH, $\#\mathsf{Clique}$ cannot be solved in time $f(k)n^{o(k)}$ for any computable function f [43], [44]. We would like to have a similar lower bound for $\#\mathsf{PartitionedSub}(\mathcal{H}_{\text{bicub}})$ and then, via the reduction in Section III, a lower bound for counting bipartite k-matchings. It turns out that we need a source problem different from $\#\mathsf{Clique}$ to prove (almost) tight lower bounds for $\#\mathsf{PartitionedSub}(\mathcal{H}_{\text{bicub}})$.

Theorem II.10 ([39, Corollaries 6.2–6.3])**.** *Assuming ETH, there is a universal constant D such that $\#\mathsf{PartitionedSub}$ cannot be solved in time $f(k)n^{o(k/\log k)}$, where $k = |V(H)|$ and f is any computable function, even under the restriction that H has maximum degree at most D.*

Using this theorem, we show in the full version:

Lemma II.11. *Assuming ETH, the problem $\#\mathsf{PartitionedSub}(\mathcal{H}_{\text{bicub}})$ admits no $f(k)n^{o(k/\log k)}$ time algorithm, where $k = |V(H)|$ and f is computable.*

III. Bipartite edge-colorful matchings

In this section, we prove #W[1]-hardness of counting k-matchings in bipartite graphs G. While this is interesting on its own, as previously only #W[1]-hardness for general graphs G was known, we mainly use this problem as a reduction source for the next section, where it will be crucial to assume that G is bipartite. In fact, we prove the stronger statement that counting edge-colorful k-matchings is #W[1]-hard (by Lemma II.7(2), this statement is indeed stronger). This might come as a surprise as the vertex-colorful version is fixed-parameter tractable (even on general graphs) by the discussion in the last section.

Furthermore, our reduction bypasses the algebraic machinery of [37], which built upon a technique introduced in [33] that could only guarantee that the parameter increase in the reduction is computable. Therefore, while showing #W[1]-hardness, this proof was inherently unable to show lower bounds under ETH. In the following proof, we reduce from the problem $\#\mathsf{PartitionedSub}(\mathcal{H}_{\text{bicub}})$, which admits no $f(k)n^{o(k/\log k)}$ algorithm under ETH by Lemma II.11. Our reduction will preserve this lower bound.

Theorem III.1. *The problem $\#\mathsf{Match}$ of counting k-matchings in bipartite graphs and the problem $\#\mathsf{ColMatch}$ of counting edge-colorful k-matchings in edge-colored bipartite graphs are* #W[1]*-complete and, assuming ETH, admit no $f(k)n^{o(k/\log k)}$ algorithms.*

We show the second claim, from which the first claim follows with Lemma II.7(2). The following technical lemma will be needed in the proof, and illustrates how polynomials appear in the context of counting matchings.

Lemma III.2. *Let A and B be edge-colorful graphs using colors from a set Δ. For $n \geq 0$, let $A + n \cdot B$ denote A together with n vertex-disjoint copies of B. Then for every $X \subseteq \Delta$, the value $\#\mathcal{M}_X(A + n \cdot B)$ is a polynomial in n of maximum degree $|X|$.*

Proof of Theorem III.1: We prove the statement by reduction from $\#\mathsf{PartitionedSub}(\mathcal{H}_{\text{bicub}})$. Let H and G be $[k]$-vertex-colored graphs such that H is 3-regular, bipartite and colorful. Without limitation of generality, G satisfies the condition stated in Remark II.6: There are no edges between color classes i and j of G if there is no edge between the i-colored vertex and the j-colored vertex of H.

Moreover, let $n_0 \in \mathbb{N}$ with $n_0 \geq 3$ be a fixed universal constant (independent of H and G) whose value will be determined at the end of the proof. We assume that there is some $n \in \mathbb{N}$ such that $|V_i(G)| = n$ for all $i \in [k]$ and $n > n_0$. This can be ensured by adding isolated vertices to G. (Note that isolated vertices cannot appear in subgraphs $F \subseteq G$ with $F \simeq H$ as H is 3-regular.) In the following, consider H as a Γ-edge-colorful graph, where $\Gamma \simeq E(H)$ is a set of $3k/2$ colors.

For each vertex of H, let us fix an arbitrary ordering of the three edges incident to it. Let $\Delta := [k] \times [6]$ and let $G^{\triangle}$ be the edge-colored graph with colors $\Gamma \cup \Delta$, which is obtained from G as follows:

1) Replace each $v \in V(G)$ by a cycle C_6 on the vertices $w_{v,1},\ z_{v,1},\ w_{v,2},\ z_{v,2},\ w_{v,3},\ z_{v,3}$. The edges of the cycle are colored with $\{i\} \times [6]$ the way it is shown in Figure 1.
2) Let us define the independent set $I(v) = \{w_{v,1}, w_{v,2}, w_{v,3}\}$. For each vertex-color $i \in [k]$ of G, define $\mathcal{I}(i) = \bigcup_{v \in V_i(G)} I(v)$.
3) For $e \in E(H)$ with $e = \{i, j\}$, let $a, b \in [3]$ be such that e is the a-th edge incident with i and the b-th edge incident with j. Replace each $\{u, v\} \in E(G)$ where u is i-colored and v is j-colored by the edge $\{w_{u,a}, w_{v,b}\}$ of color $\gamma(e) \in \Gamma$.

It is easy to see that $G^{\triangle}$ is bipartite. For $X \subseteq \Gamma \cup \Delta$, recall that $\mathcal{M}_X(G^{\triangle})$ denotes the set of matchings of $G^{\triangle}$ that contain exactly one edge of each color in X. At first, we

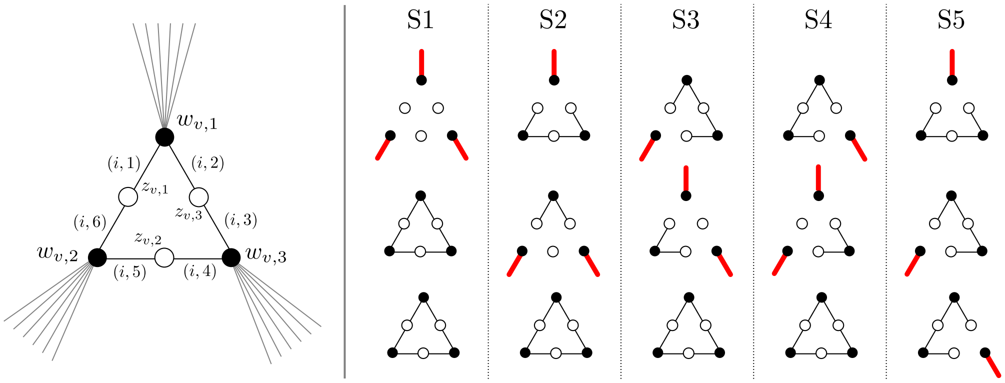

Figure 1. Each column represents one type. The partition of M_i is depicted with red edges. The black edges show the edges of the cycles not incident to the matching; these edges form the graphs R_s.

will only be interested in $\mathcal{N} := \mathcal{M}_\Gamma(G^\triangle)$, i.e., in colorful matchings of the subgraph of $G^\triangle$ that contains no C_6-edges. Observe that for $M \in \mathcal{N}$ and $i \in [k]$, the set $V(M) \cap \mathcal{I}(i)$ contains exactly three vertices, which could be contained within a single set $I(v)$ for some $v \in V(G)$, or they could be spread over different such sets. That is, the three vertices can be all in the same $I(v)$, or be in three different sets $I(v_1)$, $I(v_2)$, $I(v_3)$, or one of them can be in some $I(v_1)$ and the other two in some $I(v_2)$. This last case further splits into three subcases: there is an $i \in [3]$ such that $w_{v_1,i}$ is used from $I(v_1)$ and the two vertices $w_{v_2,j}$ for $j \in [3] \setminus \{i\}$ are used from $I(v_2)$. In total, this yields five possibilities how the matching M can look like from the viewpoint of the cycles representing $V_i(G)$, as shown in Figure 1.

We formally define the five possible *types* depicted in Figure 1 as follows. For $M \in \mathcal{N}$ and $i \in [k]$, call $u, v \in V(M) \cap \mathcal{I}(i)$ equivalent if there exists some $w \in V(G)$ such that $u, v \in I(w)$. This equivalence notion induces a partition $\theta_i(M)$ of $V(M) \cap \mathcal{I}(i)$, which we refer to by its index in Figure 1. Let the vector $\theta(M) = (\theta_1(M), \ldots, \theta_k(M))$ be the *type* of M, and let $\Theta := [5]^k$ be the set of all types. For $\theta \in \Theta$, let $\mathcal{N}[\theta] := \{M \in \mathcal{N} \mid \theta(M) = \theta\}$ denote the matchings of type θ. Let $\theta^* = (1, \ldots, 1)$ denote the good type. We write $\theta(i)$ for the i-th coordinate of a type θ.

Then $\mathcal{N}[\theta^*] \simeq \mathsf{PartitionedSub}(H \to G)$: Every $M \in \mathcal{N}[\theta^*]$ encodes a copy of H as the edges in M involve exactly one vertex of color i for every $i \in [k]$; conversely every H-copy induces a unique $M^* \in \mathcal{N}[\theta^*]$. However, $\mathcal{N}[\theta]$ for $\theta \neq \theta^*$ stands in no useful relation to H-copies.

In the following, we consider the edge-colorful matchings in $\mathcal{M}_X(G^\triangle)$, for certain sets $\Gamma \subseteq X \subseteq \Gamma \cup \Delta$. Each matching in $\mathcal{M}_X(G^\triangle)$ is an extension of a matching $M \in \mathcal{N}$. Different matchings $M \in \mathcal{N}$ have different numbers of extensions in $\mathcal{M}_X(G^\triangle)$, but we show that the contribution of M depends only on its type $\theta(M)$. Therefore, the size of $\mathcal{M}_X(G^\triangle)$ can be interpreted as a weighted sum over $M \in \mathcal{N}$ with weights depending on $\theta(M)$. Our goal is to deduce the number of matchings $M \in \mathcal{N}$ of type θ^* from the resulting system of linear equations.

This task requires a few definitions. For $t \in [5]$, define $A_1 := \{4, 5\}$, $A_2 := \{2, 3\}$, $A_3 := \{1, 6\}$, $A_4 := \{2, 3, 4, 5\}$ and $A_5 := \{1, 2, 3, 4, 5, 6\}$. For $i \in [k]$, write $A_t^i = \{i\} \times A_t$. Note that these are the colors appearing on the cycles representing vertices of $V_i(G)$. For $\mathbf{t} \in [5]^k$, let

$$X(\mathbf{t}) := \Gamma \cup A^1_{\mathbf{t}(1)} \cup \ldots \cup A^k_{\mathbf{t}(k)}.$$

For $s \in [5]$ and $i \in \Gamma$, let C_6^i be the cycle representing vertices of $V_i(G)$. We introduce a specific auxiliary graph R_s, which is an induced subgraph of three disjoint copies of C_6^i, after removing vertices incident to a matching of type s; clearly, R_s has exactly $3 \cdot 6 - 3 = 15$ vertices. These graphs are drawn in Figure 1. By Lemma III.2, for all $s, t \in [5]$, the quantity

$$p_{s,t}(n) := \#\mathcal{M}_{A_t^i}(R_s + n \cdot C_6^i)$$

is a polynomial in n of maximum degree 6 which is independent of H and G. In principle, the 25 polynomials $p_{s,t}$ could be calculated and written out explicitly, but we will see that it is sufficient to know that they are polynomials.

By invoking the next claim on all possible $\mathbf{t} \in [5]^k$, we obtain 5^k linear equations involving

- the results $\#\mathcal{M}_{X(\mathbf{t})}(G^\triangle)$ of oracle calls on $\#\mathsf{ColMatch}$, for $\mathbf{t} \in [5]^k$, and
- products of numbers $p_{s,t}(n)$ for $s, t \in [5]$, where $p_{s,t}$ are defined above, and
- the number of matchings $\#\mathcal{N}[\theta]$ for all 5^k types $\theta \in \Theta$, including the desired $\#\mathcal{N}[\theta^*]$.

Claim III.3. *Let $n \geq 3$, as assumed in this section. For every $\mathbf{t} \in [5]^k$, it holds that*

$$\#\mathcal{M}_{X(\mathbf{t})}(G^\triangle) = \sum_{\theta \in \Theta} \#\mathcal{N}[\theta] \cdot \prod_{i \in [k]} p_{\theta(i),\mathbf{t}(i)}(n-3). \quad (1)$$

For $\mathbf{t} \in [5]^k$, consider (1) as a linear equation in the unknowns $\#\mathcal{N}[\theta]$: We obtain T equations in T unknowns, where $T = 5^k$. By Gaussian elimination, a solution to this system can be found in time $\mathcal{O}(T^3)$, but it is crucial to show that this solution is unique, i.e., that its system matrix $R_k(n)$ has full rank. We show that there is a number $n_0 \in \mathbb{N}$ independent of H and G, such that for all $n, k \in \mathbb{N}$ with $n > n_0$, the matrix $R_k(n)$ has full rank.

For $R \in \mathbb{Z}^{\ell \times \ell}$ and $k \in \mathbb{N}$, we write $R^{\otimes k}$ for the k-th Kronecker power of R: The ℓ^k rows and columns of $R^{\otimes k}$ are indexed by the lexicographical ordering of vectors $\mathbf{i}, \mathbf{j} \in [\ell]^k$, and it holds that $(R^{\otimes k})_{\mathbf{i},\mathbf{j}} = \prod_{s \in [k]} R_{\mathbf{i}(s),\mathbf{j}(s)}$. Let us observe that $R_k(n) = (R_1(n))^{\otimes k}$, where $R_1(n)$ is the 5×5 matrix with $(R_1(n))_{s,t} = p_{s,t}(n)$ for $s, t \in [5]$. It is a basic property of the Kronecker product that the k-th Kronecker power of a nonsingular square matrix is also nonsingular. Therefore, we only need to verify that $R_1(n)$ is nonsingular. By Lemma III.2, the value $p_{s,t}(n)$ is a polynomial in n for every $s, t \in [5]$, hence the determinant $\det(R_1(n))$ is also a polynomial in n. This means that it is either zero for every $n \in \mathbb{Z}$, or zero only for finitely many n. Recall that $(R_1(0))_{s,t} = p_{s,t}(0) = \#\mathcal{M}_{A_t^i}(R_s)$, that is, the number of A_t^i-colored matchings in a specific 15-vertex graph R_s, which can be computed with some effort. In the full version, we show that $\det(R_1) \neq 0$.

This implies that $\det(R_1(n))$, interpreted as a polynomial in n, is not identically 0, which in turn implies that $R_1(n)$ is singular only for finitely many n. Hence the linear system admits a unique solution if $n > n_0$, which we assumed in the beginning by adding isolated vertices to G. ∎

We can transfer the lower bound of Theorem III.1 for bipartite k-matchings to k-cycles and k-paths.

Theorem III.4. *The problems of counting (directed or undirected) paths or cycles of length k are* #W[1]*-hard and admit no $f(k)n^{o(k/\log k)}$ algorithm unless ETH fails.*

IV. Reducing matchings to $\#\mathsf{Sub}(\mathcal{H})$

To show hardness for $\mathsf{Sub}(\mathcal{H})$, we develop a general machinery of k-matching gadgets, which are graphs $H \in \mathcal{H}$ together with a partition of $V(H)$ into an induced matching M and some remainder C. These gadgets satisfy certain technical properties which will be used in Theorem IV.7, which is the main reduction of this paper. It states that, if $\mathcal{H}$ is a class of graphs that contains k-matching gadgets for all $k \in \mathbb{N}$, then there is a parameterized Turing reduction from the problem of counting (uncolored) k-matchings in bipartite graphs G to the problem $\#\mathsf{Sub}(\mathcal{H})$.

In the remainder of this section, we define k-matching gadgets formally, give some first examples of their properties, and then prove Theorem IV.7. Proving the actual existence of k-matching gadgets in graph classes $\mathcal{H}$ will be the task of the subsequent sections.

Definition IV.1. Let H be a graph. For $C \subseteq V(H)$, let $\partial_H(C)$ denote the set of vertices in C that have a neighbor in $V(H) \setminus C$. If f is an isomorphism from $H[C]$ to $H[C']$ for some $C, C' \subseteq V(H)$ such that $f(\partial_H(C)) = \partial_H(C')$, then we say that f is *boundary preserving.*

Observe that $X \subseteq Y$ implies $(X \setminus \partial_H(X)) \subseteq (Y \setminus \partial_H(Y))$: if $v \in X$ has no neighbor outside X, then it has no neighbor outside Y either.

The following definition formulates the properties of the gadgets we need in the main reduction (Theorem IV.7).

Definition IV.2. Let H be a graph, M be an induced k-matching in H, and let $C := V(H) \setminus V(M)$. We say that (H, M) is a *k-matching gadget* if whenever an isomorphism f from $H[C]$ to $H[C']$ for some $C' \subseteq V(H)$ satisfies

(P1) $H \setminus C'$ has no isolated vertex,
(P2) $H \setminus C'$ is bipartite, and
(P3) f is boundary preserving,

then it is also true that $H \setminus C'$ is a k-matching, i.e., $H \setminus C'$ is isomorphic to the graph on $2k$ vertices that contains k vertex-disjoint edges.

Using a rather extensive graph-theoretical analysis, we will show in Sections V-VII:

Theorem IV.3. *Let $\mathcal{H}$ be a graph class of unbounded vertex-cover number and bounded treewidth. Then, for all $k \in \mathbb{N}$, there exists a graph $H \in \mathcal{H}$ and a subset $M \subseteq V(H)$ such that (H, M) is a k-matching gadget.*

It indeed suffices to consider classes $\mathcal{H}$ covered by this theorem: By Theorem II.3, the problem $\#\mathsf{Sub}(\mathcal{H})$ admits a polynomial-time algorithm if $\mathcal{H}$ has bounded vertex-cover number. If $\mathcal{H}$ has unbounded treewidth, then $\#\mathsf{Sub}(\mathcal{H})$ is #W[1]-complete by Theorem II.8.

It will be convenient to know that if a k-matching gadget exists, then a k_0-matching gadget also exists for every $k_0 < k$. This is not obvious from the definition and requires a nontrivial proof.

Lemma IV.4. *If (H, M) is a k-matching gadget and $M_0 \subseteq M$ is a k_0-matching, then (H, M_0) is a k_0-matching gadget.*

The following lemma shows a simple condition that guarantees the correctness of a k-matching gadget.

Lemma IV.5. *Let M be an induced k-matching in a graph H such that every vertex of $C := V(H) \setminus V(M)$ is adjacent to at most one vertex of $V(M)$. Then (M, H) is a k-matching gadget.*

We will see condition (P1) is usually easy to achieve (by making M somewhat smaller), but ensuring condition (P2) will be more involved.

The main reduction is described by the following lemma and theorem, which provide a reduction from counting bipartite k-matchings to $\mathsf{Sub}(\mathcal{H})$ whenever $\mathcal{H}$ contains k-matching gadgets of all sizes.

Lemma IV.6. *Let G be a graph and let (H, M) be a k-matching gadget of size $t = |V(H)|$. Then we can compute* *the number of k-matchings in G from $2k \cdot 2^{\mathcal{O}(t^2)}$ oracle queries of the form $\#\mathsf{Sub}(H \to G')$, where G' is an arbitrary graph.*

This readily implies the hardness result.

Theorem IV.7. *If $\mathcal{H}$ is a recursively enumerable graph class that contains a k-matching gadget for every $k \in \mathbb{N}$, then $\#\mathsf{Sub}(\mathcal{H})$ is #W[1]-complete.*

V. Bounded-degree graphs

The goal of this section is to prove Theorem IV.3, the existence of k-matching gadgets, for the special case of graph classes $\mathcal{H}$ with bounded maximum degree and unbounded vertex-cover number. The results in Sections VI and VII for other graph classes are based on this result for bounded-degree graphs. The basic idea is that in bounded-degree graphs we are close to the situation described by Lemma IV.5: clearly, the two endpoints of an edge in the matching can have only a bounded number of common neighbors; in this sense property (P2) "almost holds." We choose a candidate (H, M) for the k-matching gadget and see how it can fail. If for every C' satisfying (P1)-(P3), the graph $H \setminus C'$ still has many components of size 2 (so it is "almost a matching"), then we can extract a correct k'-matching gadget for some relatively large $k' < k$. Suppose therefore that (H, C) "spectacularly fails": $H \setminus C'$ has only few components of size 2. As $H \setminus C'$ has no isolated vertices, this is only possible if $H \setminus C'$ has many more edges than the k-matching M. Then we argue that now the total degree on the boundary of C' is much smaller than on the boundary of C, and we can use this to find an induced matching in $H \setminus C'$ whose endpoints have strictly fewer common neighbors than in M. As the graph has bounded degree, repeating this argument a constant number of times eventually leads to a matching where the endpoints of the edges have no common neighbors, hence Lemma IV.5 can be invoked.

In a bounded-degree graph, any sufficiently large set of edges contains a large matching and in fact a large induced matching: we can greedily select edges and we need to throw away only a bounded number of edges after each selection. Moreover, in order to move closer to the situation described in Lemma IV.5, we may also satisfy the requirement that the selected edges have no common neighbors (but it is possible that the two endpoints of an edge have common neighbors).

Lemma V.1. *Let F be a set of edges in a graph G with maximum degree D. Then there is an induced matching $M' \subseteq F$ of size at least $|F|/(2D^2)$. Furthermore, there is an induced matching $M'' \subseteq F$ of size at least $|F|/(2D^3)$ such that every vertex of $V(G) \setminus V(M'')$ is adjacent to at most one edge of M''.*

For bounded-degree graphs, Lemma V.1 implies that there is not much difference between having a large set of edges, a large matching, a large induced matching, or a large induced matching satisfying the requirement that every vertex outside the matching is adjacent to at most one edge of the matching.

Lemma V.2. *There is a function $f_d(k_0, D)$ such that the following holds. If H is a graph with maximum degree at most D and contains a matching of size at least $f_d(k_0, D)$, then there is a k_0-matching gadget (H, M_0).*

VI. Graphs with no large subdivided stars

A *subdivided ℓ-star* consists of a center vertex v and ℓ paths of length 2 starting at v that do not share any vertex other than v. We denote by $\psi(v)$ the largest integer ℓ such that v is the center of a subdivided ℓ-star. We denote by $\psi(G)$ the maximum of $\psi(v)$ for every $v \in V(G)$. The goal of this section is to prove Theorem IV.3, the existence of k-matching gadgets, for graphs where $\psi(G)$ is bounded.

We develop a technology that allows us to "ignore" certain sets Q of vertices: if $H \setminus Q$ has a k-matching gadget, then so does H. This works for sets Q where the vertices have some characteristic property (e.g., based on degrees) that allows us to distinguish them from the vertices not in Q (see below). We use this technique to reduce the problem to bounded-degree graphs. If we have a large induced matching where every vertex has small degree, then we define Q to be the vertices of "large degree." Now $H \setminus Q$ is clearly a bounded-degree graph and hence Lemma V.2 can be invoked. Suppose therefore that we have an induced matching where every vertex has large degree. Then we define Q to be the vertices of "small degree." Somewhat unexpectedly, $H \setminus Q$ is a bounded-degree graph also in this case: this follows from the fact that if $\psi(G)$ is bounded, then a vertex cannot have many neighbors of large degree.

Proposition VI.1. *Every vertex $v \in V(G)$ has at most $\psi(v)$ neighbors with degree at least $2\psi(v) + 2$.*

Therefore, we can reduce the problem to bounded-degree graphs also in the case of a matching with large degree vertices. Finally, if we have a large induced matching with "mixed" edges, that is, each having both a small-degree and a large-degree endpoint, then we can reduce to one of the previous two cases by looking at the common neighbors of the endpoints.

The following definition will be crucial for the clean treatment of the problem. We show that if a set is "well identifiable" (for example, based on degrees etc.) then we can remove it from the graph and it is sufficient to show that the remaining part of the graph has a k-matching gadget. The definition formulates this condition as invariance under certain isomorphisms.

Definition VI.2. Let H be a graph and let $X \subseteq C \subseteq V(H)$ be two subsets of vertices. We say that X is a *strong set* with respect to C if whenever f is a boundary-preserving isomorphism from $H[C]$ to $H[C']$ for some $C' \subseteq V(H)$, then $f(X) = X$ (in particular, this implies $X \subseteq C'$).

Observe that $f(X)$ and X have the same size, thus to prove $f(X) = X$, it is sufficient to prove $f(X) \subseteq X$, that is, $v \in X$ implies $f(v) \in X$.

As a simple example, suppose that every vertex in H has either degree at most d or degree at least $d + 2k + 1$ and $M \subseteq H$ is a k-matching with every vertex having degree at most d in H. Let $C = V(H) \setminus V(M)$ and let $X \subseteq C$ be the set of vertices with degree at least $d + 2k + 1$. Then X is a strong set: every vertex $x \in X$ has at least $d + 2k + 1 - |V(M)| = d + 1$ neighbors in C, hence $f(v)$ has at least $d + 1$ neighbors in C', implying that $f(v) \in X$ (as we assumed that degree larger than d implies that the degree is at least $d+2k+1$). In fact, it is sufficient to enforce the degree requirement only for vertices $v \in \partial_H(C)$: it is sufficient if we require that the degree of every vertex in $\partial_H(C)$ is either at most d or at least $d + 2k + 1$, but the degrees of the vertices in $C \setminus \partial_H(C)$ can be arbitrary. This is sufficient, as if $v \in C \setminus \partial_H(C)$, then every neighbor of v is in C and (P3) of f implies that every neighbor of $f(v)$ is in C', hence (as $H[C]$ and $H[C']$ are isomorphic) vertices v and $f(v)$ have exactly the same degree.

We show now that removing a strong set disjoint from M does not affect whether a k-matching gadget is correct.

Lemma VI.3. *Let H be a graph containing an induced k-matching M, let $C := V(H) \setminus V(M)$, and let $X \subseteq C$ be a strong set with respect to C. If $(H \setminus X, M)$ is a k-matching gadget, then so is (H, M).*

Similarly to bounded-degree graphs (Lemma V.1), we can use a bound on $\psi(H)$ to argue that not too many edges can be in the neighborhood of an edge and therefore a large set of edges implies a large induced matching. However, all we need now is that a large induced matching implies that there is a large induced matching such that every vertex outside the matching is adjacent to at most one edge of the matching.

Lemma VI.4. *Let M be an induced matching of size at least $2kL^2$ in a graph H with $\psi(G) \leq L$. Then there is an $M' \subseteq M$ of size at least k such that every vertex of $V(G) \setminus V(M')$ is adjacent to at most one edge of M'.*

Recall the example after Definition VI.2: if there is a sufficiently large "gap" in the degrees of the vertices of $N(V(M))$ for a matching M, then we can define a strong set simply based on the degrees. The following lemma creates such a gap of arbitrary large size by throwing away at most half of the edges of a matching.

Lemma VI.5. *Let F be an induced matching in a graph H with $\psi(H) \leq L$. For every $x \geq 2L + 2$, $y \geq 1$, there is an induced matching $F' \subseteq F$ of size at least $|F|/2$ and an $x \leq g \leq x + 4(2L+2)y$ such that $N(V(F'))$ has no vertex whose degree in H is in the range $\{g, \dots, g + y - 1\}$.*

Now we are ready to prove that main result for graphs not having large subdivided stars. The proof uses Lemma VI.3 to remove a set of vertices, making the graph bounded degree, and then the bounded-degree result Lemma V.2 can be invoked.

Lemma VI.6. *There is a function $f_s(k_0, L)$ such that if graph H with $\psi(H) \leq L$ has an induced matching of size $f_s(k_0, L)$, then there is a k_0-matching gadget (H, M_0).*

VII. Bounded-treewidth graphs

In this section, we complete the proof of Theorem IV.3 by showing that if a bounded-treewidth graph has large vertex-cover number, then it contains a k-matching gadget. From a Ramsey argument contained in the full version, it follows that bounded-treewidth graphs with large vertex-cover number contain large induced matchings. We give another proof (Lemma VII.1) of this fact by looking at the tree decomposition instead of using Ramsey arguments. The proof finds an induced matching such that $\psi(v)$ is bounded for every vertex v of the matching, that is, there are no large subdivided stars centered on them. Then we define Q to be the set of vertices with large ψ-number (this require some care) and use the technology developed in Section VI (Lemma VI.3) to argue that it is sufficient to find a k-matching gadget in $H \setminus Q$. Clearly, $\psi(H \setminus Q)$ is bounded, hence Lemma VI.6 can be invoked.

Lemma VII.1. *Let w and k be integers and let H be a graph of treewidth at most w and vertex cover number greater than $3k(w + 1)$. Then there is an induced matching $M = \{u_1v_1, \dots, u_kv_k\}$ such that $\psi(u_i), \psi(v_i) \leq 2(w + 1)$ for every $1 \leq i \leq k$.*

The following two technical lemmas will be used in the proof of Lemma VII.4.

Lemma VII.2. *If $\mathcal{H}$ is a multiset of at least $(1 + z \cdot r)k$ subsets of a universe U, each having size at most r, then there is a subcollection $\mathcal{H}' \subseteq \mathcal{H}$ of size k such that for every $x \in U$, either there is at most one set in $\mathcal{H}'$ containing x, or there are at least z sets in $\mathcal{H} \setminus \mathcal{H}'$ containing x.*

Lemma VII.3. *Let H be of treewidth $\leq w$ and let $Z \subseteq V(H)$. If for every $v \in Z$ there is a subdivided star S_v centered at v covering every vertex of Z, then $|Z| \leq w + 1$.*

We are now ready to prove the main result for bounded-treewidth graphs, which completes the proof Theorem IV.3.

Lemma VII.4. *There is a function $f(k, w)$ such that if a graph H with treewidth $\leq w$ has vertex cover number $> f(k, w)$, then there is a k-matching gadget (H, M).*

References

[1] L. G. Valiant, "The complexity of computing the permanent," *Theoret. Comput. Sci.*, vol. 8, no. 2, pp. 189–201, 1979.

[2] A. A. Bulatov, "The complexity of the counting constraint satisfaction problem," *J. ACM*, vol. 60, no. 5, p. 34, 2013.

[3] A. A. Bulatov, M. E. Dyer, L. A. Goldberg, M. Jerrum, and C. McQuillan, "The expressibility of functions on the boolean domain, with applications to counting CSPs," *J. ACM*, vol. 60, no. 5, p. 32, 2013.

[4] A. A. Bulatov, M. E. Dyer, L. A. Goldberg, M. Jalsenius, M. Jerrum, and D. Richerby, "The complexity of weighted and unweighted #CSP," *J. Comput. Syst. Sci.*, vol. 78, no. 2, pp. 681–688, 2012.

[5] M. E. Dyer, L. A. Goldberg, and M. Jerrum, "The complexity of weighted boolean CSP," *SIAM J. Comput.*, vol. 38, no. 5, pp. 1970–1986, 2009.

[6] A. A. Bulatov, M. E. Dyer, L. A. Goldberg, M. Jalsenius, and D. Richerby, "The complexity of weighted boolean #CSP with mixed signs," *Theor. Comput. Sci.*, vol. 410, no. 38-40, pp. 3949–3961, 2009.

[7] T. Schank and D. Wagner, "Finding, counting and listing all triangles in large graphs, an experimental study," in *WEA*, 2005, pp. 606–609.

[8] C. E. Tsourakakis, "Fast counting of triangles in large real networks without counting: Algorithms and laws," in *ICDM*, 2008, pp. 608–617.

[9] L. Becchetti, P. Boldi, C. Castillo, and A. Gionis, "Efficient semi-streaming algorithms for local triangle counting in massive graphs," in *KDD*, 2008, pp. 16–24.

[10] M. N. Kolountzakis, G. L. Miller, R. Peng, and C. E. Tsourakakis, "Efficient triangle counting in large graphs via degree-based vertex partitioning," *Internet Mathematics*, vol. 8, no. 1-2, pp. 161–185, 2012.

[11] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon, "Network motifs: simple building blocks of complex networks." *Science*, vol. 298, no. 5594, pp. 824–827, October 2002.

[12] A. Itai and M. Rodeh, "Finding a minimum circuit in a graph," *SIAM J. Comput.*, vol. 7, no. 4, pp. 413–423, 1978.

[13] N. Alon, R. Yuster, and U. Zwick, "Color-coding," *J. ACM*, vol. 42, no. 4, pp. 844–856, 1995.

[14] ——, "Finding and counting given length cycles," *Algorithmica*, vol. 17, no. 3, pp. 209–223, 1997.

[15] T. Kloks, D. Kratsch, and H. Müller, "Finding and counting small induced subgraphs efficiently," *Inf. Process. Lett.*, vol. 74, no. 3-4, pp. 115–121, 2000.

[16] A. Björklund, T. Husfeldt, P. Kaski, and M. Koivisto, "Counting paths and packings in halves," in *ESA*, 2009, pp. 578–586.

[17] F. V. Fomin, D. Lokshtanov, V. Raman, S. Saurabh, and B. V. R. Rao, "Faster algorithms for finding and counting subgraphs," *J. Comput. Syst. Sci.*, vol. 78, no. 3, pp. 698–706, 2012.

[18] V. V. Williams and R. Williams, "Finding, minimizing, and counting weighted subgraphs," *SIAM J. Comput.*, vol. 42, no. 3, pp. 831–854, 2013.

[19] P. Floderus, M. Kowaluk, A. Lingas, and E.-M. Lundell, "Detecting and counting small pattern graphs," in *ISAAC*, 2013, pp. 547–557.

[20] A. Björklund, P. Kaski, and L. Kowalik, "Counting thin subgraphs via packings faster than meet-in-the-middle time," in *SODA*, 2014, pp. 594–603.

[21] M. Kowaluk, A. Lingas, and E.-M. Lundell, "Counting and detecting small subgraphs via equations," *SIAM J. Discrete Math.*, vol. 27, no. 2, pp. 892–909, 2013.

[22] M. Grohe, T. Schwentick, and L. Segoufin, "When is the evaluation of conjunctive queries tractable?" in *STOC '01: Proceedings of the thirty-third annual ACM symposium on Theory of computing*. New York, NY, USA: ACM Press, 2001, pp. 657–666.

[23] M. Grohe, "The complexity of homomorphism and constraint satisfaction problems seen from the other side," *J. ACM*, vol. 54, no. 1, p. 1, 2007.

[24] V. Dalmau and P. Jonsson, "The complexity of counting homomorphisms seen from the other side," *Theor. Comput. Sci.*, vol. 329, no. 1-3, pp. 315–323, 2004.

[25] Y. Chen, M. Thurley, and M. Weyer, "Understanding the complexity of induced subgraph isomorphisms," in *ICALP (1)*, 2008, pp. 587–596.

[26] D. Marx, "Tractable hypergraph properties for constraint satisfaction and conjunctive queries," *J. ACM*, vol. 60, no. 6, p. 42, 2013.

[27] K. Meeks, "The challenges of unbounded treewidth in parameterised subgraph counting problems," *CoRR*, vol. abs/1402.5857, 2014.

[28] M. Jerrum and K. Meeks, "The parameterised complexity of counting connected subgraphs," *CoRR*, vol. abs/1308.1575, 2013.

[29] ——, "Some hard classes of parameterised counting problems," *CoRR*, vol. abs/1310.6524, 2013.

[30] M. Frick and M. Grohe, "Deciding first-order properties of locally tree-decomposable structures," *J. ACM*, vol. 48, no. 6, pp. 1184–1206, 2001.

[31] M. Grohe, S. Kreutzer, and S. Siebertz, "Deciding first-order properties of nowhere dense graphs," *CoRR*, vol. abs/1311.3899, 2013.

[32] Z. Dvorak, D. Král, and R. Thomas, "Testing first-order properties for subclasses of sparse graphs," *J. ACM*, vol. 60, no. 5, p. 36, 2013.

[33] J. Flum and M. Grohe, "The parameterized complexity of counting problems," *SIAM J. Comput.*, vol. 33, no. 4, pp. 892–922, 2004.

[34] J. Kneis, D. Mölle, S. Richter, and P. Rossmanith, "Divide-and-color," in *WG*, 2006, pp. 58–67.

[35] R. Williams, "Finding paths of length k in $O^*(2^k)$ time," *Inf. Process. Lett.*, vol. 109, no. 6, pp. 315–318, 2009.

[36] A. Björklund, "Determinant sums for undirected Hamiltonicity," in *FOCS*, 2010, pp. 173–182.

[37] R. Curticapean, "Counting matchings of size k is #W[1]-hard," in *ICALP (1)*, 2013, pp. 352–363.

[38] M. Bläser and R. Curticapean, "Weighted counting of k-matchings is #W[1]-hard," in *IPEC*, 2012, pp. 171–181.

[39] D. Marx, "Can you beat treewidth?" *Theory of Computing*, vol. 6, no. 1, pp. 85–112, 2010.

[40] R. Impagliazzo, R. Paturi, and F. Zane, "Which problems have strongly exponential complexity?" *J. Comput. System Sci.*, vol. 63, no. 4, pp. 512–530, 2001.

[41] N. Robertson and P. D. Seymour, "Graph minors. V. Excluding a planar graph," *J. Combin. Theory Ser. B*, vol. 41, no. 1, pp. 92–114, 1986.

[42] V. Arvind and V. Raman, "Approximation algorithms for some parameterized counting problems," in *ISAAC*, 2002, pp. 453–464.

[43] J. Chen, X. Huang, I. A. Kanj, and G. Xia, "Strong computational lower bounds via parameterized complexity," *Journal of Computer and System Sciences*, vol. 72, no. 8, pp. 1346 – 1367, 2006.

[44] D. Lokshtanov, D. Marx, and S. Saurabh, "Lower bounds based on the Exponential Time Hypothesis," *Bulletin of the EATCS*, vol. 84, pp. 41–71, 2011.