Detecting and Counting Small Subgraphs, and Evaluating a Parameterized Tutte Polynomial: Lower Bounds via Toroidal Grids and Cayley Graph Expanders

Marc Roth ✉ 🆔
Merton College, University of Oxford, UK

Johannes Schmitt ✉ 🆔
Mathematical Institute, University of Bonn, Germany

Philip Wellnitz ✉ 🆔
Max Planck Institute for Informatics, Saarland Informatics Campus (SIC), Saarbrücken, Germany

─── **Abstract** ───

Given a graph property Φ, we consider the problem EDGESUB(Φ), where the input is a pair of a graph G and a positive integer k, and the task is to decide whether G contains a k-edge subgraph that satisfies Φ. Specifically, we study the parameterized complexity of EDGESUB(Φ) and of its counting problem #EDGESUB(Φ) with respect to both approximate and exact counting. We obtain a complete picture for minor-closed properties Φ: the decision problem EDGESUB(Φ) always admits an FPT ("fixed-parameter tractable") algorithm and the counting problem #EDGESUB(Φ) always admits an FPTRAS ("fixed-parameter tractable randomized approximation scheme"). For exact counting, we present an exhaustive and explicit criterion on the property Φ which, if satisfied, yields fixed-parameter tractability and otherwise #W[1]-hardness. Additionally, most of our hardness results come with an almost tight conditional lower bound under the so-called Exponential Time Hypothesis, ruling out algorithms for #EDGESUB(Φ) that run in time $f(k) \cdot |G|^{o(k/\log k)}$ for any computable function f.

As a main technical result, we gain a complete understanding of the coefficients of toroidal grids and selected Cayley graph expanders in the homomorphism basis of #EDGESUB(Φ). This allows us to establish hardness of exact counting using the Complexity Monotonicity framework due to Curticapean, Dell and Marx (STOC'17). This approach does not only apply to #EDGESUB(Φ) but also to the more general problem of computing weighted linear combinations of subgraph counts. As a special case of such a linear combination, we introduce a parameterized variant of the Tutte Polynomial T_G^k of a graph G, to which many known combinatorial interpretations of values of the (classical) Tutte Polynomial can be extended. As an example, $T_G^k(2,1)$ corresponds to the number of k-forests in the graph G. Our techniques allow us to completely understand the parameterized complexity of computing the evaluation of T_G^k at every pair of rational coordinates (x,y). In particular, our results give a new proof for the #W[1]-hardness of the problem of counting k-forests in a graph.

48th International Colloquium on Automata, Languages, and Programming (ICALP 2021).
Editors: Nikhil Bansal, Emanuela Merelli, and James Worrell; Article No. 108; pp. 108:1–108:16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1   Extended Abstract

Be it searching for cliques in social networks or understanding protein-protein interaction networks, many interesting real-life problems boil down to finding (or counting) small patterns in large graphs. Hence, to no surprise, finding (and counting) small patterns in large graphs are among the most well-studied computational problems in the fields of database theory [12, 40, 32, 13, 27], molecular biology and bioinformatics [38, 1, 56, 59], and network science [60, 52, 53]. In fact, already in the 1970s, the relevance of *finding* patterns became apparent in the context of finding cliques, finding Hamiltonian paths, or finding specific subgraphs in general [19, 18, 62, 12]. However, with the advent of motif counting for the frequency analysis of small structures in complex networks [52, 53], it became evident that detecting the existence of a pattern graph is not enough; we also need to *count* all of the occurrences of the pattern.

In this work, our patterns are (not necessarily induced) edge subgraphs that satisfy a certain graph property: for instance, given a graph, we want to count all occurrences of edge subgraphs that are are planar or connected.

From a classical point of view, often the problem of *finding* patterns is already NP-hard: prime examples include the aforementioned problems of finding (maximum) cliques or Hamiltonian paths. However, for the task of network motif counting, the patterns are (almost) always much smaller than the network itself (see [52, 53, 1]). This motivates a *parameterized* view: can we obtain fast algorithms to compute the number of occurrences of "small" patterns? If we cannot, can we at least obtain fast (randomized) algorithms to compute an *estimate* of this number? And if we cannot even do this, can we at least obtain fast algorithms to *detect* an occurrence? In this work, we completely answer all of the above questions for patterns that are specified by *minor-closed* graph properties (such as planarity) or selected other graph properties (such as connectivity).

As it turns out, the techniques we develop for answering the above questions are quite powerful: they easily generalize to a parameterized version of the Tutte polynomial. Specifically, our techniques allow us to completely understand at which rational points we can evaluate said parameterized Tutte polynomial in reasonable time, and at which rational points this is not feasible. This dichotomy turns out to be similar, but not equal, to the complexity landscape of the classical Tutte polynomial due to Jaeger et al. [42].

Parameterized Counting and Hardness

By now, *counting complexity theory* is a well established subfield of theoretical computer science. Already in the 1970s, Valiant started a formal study of counting problems when investigating the complexity of the permanent [63, 64]: counting the number of perfect matchings in a graph is #P-complete, and hence harder than any problem in the polynomial-time hierarchy PH by Toda's Theorem [61]. In contrast, *detecting* a perfect matching in a graph is much easier and can be done in polynomial time [33]. Hence, counting problems can be much harder than their decision problem counterparts.

As an attempt to overcome the hardness of counting problems in general, the focus shifted to a multivariate or *parameterized* view on these problems. Consider for example the following problem: given a query φ of size k and a database B of size n, we want to count the number of answers to φ in B. If we make the very reasonable assumption that k is much smaller than n, then we may consider an algorithm running in time $O(2^k \cdot n)$ as *tractable*. Note that in particular, such an algorithm may even outperform an algorithm running in time $O(n^2)$. Also consider [39] for a more detailed and formal discussion.

Formally, given a problem P and a *parameterization* κ that maps each instance I of P to a parameter $\kappa(I)$, we say that P is *fixed-parameter tractable* (FPT) with respect to κ, if there is an algorithm that solves each instance I of size n in time $f(\kappa(I)) \cdot n^{O(1)}$, for some computable function f. This notion was introduced by Downey and Fellows in the early 1990s [29, 30] and has itself spawned a rich body of literature (see [35, 31, 23]). In the context of the problems of detecting and counting small patterns in large networks, we parameterize by the size of the pattern: given a pattern of size k and a network of size n, we aim for algorithms that run in time $f(k) \cdot n^{O(1)}$, for some computable function f. However, for some patterns, even this goal is too ambitious: it is widely believed that even finding a clique of size k is not fixed-parameter tractable; in particular, an FPT algorithm for finding a clique of size k would also imply a breakthrough result for the Satisfiability Problem and thereby refute the widely believed Exponential Time Hypothesis [15, 16]. If a problem P is at least as hard as finding a clique (or counting all cliques) of size k, we say that P is W[1]-hard (or #W[1]-hard, respectively).

For such a (#)W[1]-hard problem, the hope is to (significantly) improve upon the naive brute-force algorithm, which runs in time $n^{O(k)}$ for the problems considered in this work. However, in view of the aforementioned reduction from the Satisfiability Problem to the problem of finding cliques of size k [14, 15], we can see that for finding cliques this, too, would require a breakthrough for the Satisfiability Problem, which, again, is believed to be unlikely [41]. In our paper, via suitable reductions from the problem of finding cliques, we establish that exact algorithms significantly faster than the brute-force algorithms are unlikely for the problems we study.

Parameterized Detection and Counting of Edge Subgraphs

Vertex-induced subgraphs as patterns are notoriously hard to detect or to count. The long line of research on this problem [47, 17, 43, 44, 51, 45, 21, 57, 28, 58] showed that this holds even if the patterns are significantly smaller than the host graphs, as witnessed by W[1] and #W[1]-hardness results and almost tight conditional lower bounds. In case of exact counting, it is in fact an open question whether there are non-trivial instances of induced subgraph counting that admit efficient algorithms; recent work [58] supports the conjecture that no such instances exist.

In search for fast algorithms, in this work, we hence consider a related, but different version of network-motif counting: for a computable graph property Φ, in the problem #EDGESUB(Φ) we are given a graph G and a positive integer k, and the task is to compute the number of (not necessarily induced) edge subgraphs[1] with k edges in G that satisfy Φ. Similarly, we write EDGESUB(Φ) for the corresponding decision problem. Then, in contrast to the case of counting vertex-induced subgraphs, for (#)EDGESUB(Φ), we identify non-trivial

---

[1] Recall that an edge subgraph G' of a graph G may have fewer edges than the subgraph of G that is induced by the vertices of G'.

properties Φ for which $(\#)\text{EDGESUB}(\Phi)$ is fixed-parameter tractable; we discuss this in more detail later. First, however, let us take a detour to elaborate more on what is known already for $(\#)\text{EDGESUB}(\Phi)$.

If the property Φ is satisfied by at most a single graph for each value of the parameter k, the decision problem $\text{EDGESUB}(\Phi)$ becomes the subgraph isomorphism problem. Hence, naturally there is a vast body of known techniques and results for special properties Φ: for FPT algorithms, think of the Colour-Coding technique by Alon, Yuster and Zwick [3], the "Divide and Colour"-technique [16], narrow sieving [7], representative sets [36], or "extensor-coding" [9] to name but a few. For hardness results, apart from the aforementioned example of detecting a clique, Lin quite recently established that detecting a k-biclique is also W[1]-hard [48]. However, a complete understanding of the parameterized decision version of the subgraph isomorphism is one of the major open problems of parameterized complexity theory [31, Chapter 33.1], that is still to be solved.

In the setting of parameterized *counting*, the situation is much better understood: Flum and Grohe [34] proved $\#\text{EDGESUB}(\Phi)$ to be $\#\text{W}[1]$-hard when Φ is the property of being a cycle, or the property of being a path. Curticapean [20] established the same result for the property of being a matching. In [22], Curticapean and Marx established a complete classification in case Φ does not hold on two different graphs with the same number of edges, which is essentially the parameterized subgraph counting problem. In particular, they identified a bound on the matching number as the tractability criterion. In a later work, together with Dell [21], they presented what is now called the framework of Complexity Monotonicity, which can be considered to be one of the most powerful tools in the field of parameterized counting problems. Note that this does not classify the decision version, as $\#\text{W}[1]$-hardness for a counting problem does not imply W[1]-hardness for the corresponding decision problem.

In contrast to the parameterized subgraph detection/counting problems, the problem $(\#)\text{EDGESUB}(\Phi)$ allows to search for more general patterns. For example, while the (parameterized) complexity of counting all subgraphs of a graph G isomorphic to a *fixed* connected graph H with k edges is fully understood [22], the case of counting all connected k-edge subgraphs of a graph G remained open so far. As one of our main results, we completely understand the problem $\#\text{EDGESUB}(\Phi)$ for the property $\Phi = $ connectivity. In what follows, we present our results, followed by an exposition of the most important techniques. Due to the space constraints, we have to defer the proofs to the full version.

Main Results

In a first part, we present our results on $(\#)\text{EDGESUB}(\Phi)$; we continue with a definition and our results for a parameterized Tutte polynomial in a second part.

Our main results on $(\#)\text{EDGESUB}(\Phi)$ can be categorized in roughly three categories: (1) exact algorithms and hardness results for the counting problem; (2) approximation algorithms for the counting problem; and (3) algorithms for the decision problem. For minor-closed properties Φ, we obtain exhaustive results for all three categories, for other (classes of) properties that we study, we obtain partial criteria. For an overview over our results on $\#\text{EDGESUB}(\Phi)$, also consider Table 1; we go into more detail in the following.

Complete Classification for Minor-Closed Properties

Let us start with the case where the graph property Φ is closed under taking minors, that is, if Φ holds for a graph, then Φ still holds after removing vertices or edges, or after contracting edges. For minor-closed properties Φ, we obtain a complete picture of the complexity of

■ **Table 1** An overview of the complexity of (#)EDGESUB(Φ) for different classes and examples of properties Φ, with respect to exact counting, approximate counting and decision. See further below for the definition of the matching and star criterion. All run-time lower bounds rely on the Exponential Time Hypothesis, and the absence of FPTRASes relies on the assumption that W[1] does not coincide with FPT under randomised parameterized reductions. We write "mixed" whenever the respective classes contain both tractable properties and hard properties. The known results about the clique problem are added for completeness; note that W[1]-hardness of decision immediately rules out an FPTRAS for approximate counting under the previous assumptions.

Property Φ	Exact Counting	Apx. Counting	Decision
Minor-closed[†] (e.g. Φ = planarity)	#W[1]-hard not in $f(k) \cdot |G|^{o(k/\log k)}$ (Main Theorem 1)	FPTRAS (Main Theorem 1)	FPT (Main Theorem 1)
Φ = connectivity	#W[1]-hard not in $f(k) \cdot |G|^{o(k/\log k)}$ (Main Theorem 2)	FPTRAS (follows from [26])	FPT (easy)
Φ = Hamiltonicity	#W[1]-hard not in $f(k) \cdot |G|^{o(k/\log k)}$ (Main Theorem 2)	unknown	unknown
Φ = Eulerianity	#W[1]-hard not in $f(k) \cdot |G|^{o(k/\log k)}$ (Main Theorem 2)	unknown	unknown
Φ = claw-freeness	#W[1]-hard not in $f(k) \cdot |G|^{o(k/\log k)}$ (Main Theorem 2)	unknown	unknown
Bounded matching number	FPT (Proposition 1.2)	FPTRAS (by exact counting)	FPT (by exact counting)
Bounded treewidth	mixed[‡]	FPTRAS (Main Theorem 3)	FPT (follows from [55])
Matching crit. **and** star crit.	mixed[*]	FPTRAS (Main Theorem 3)	FPT (Main Theorem 4)
Matching crit. **or** star crit.	mixed[‡]	mixed[§]	FPT (Main Theorem 4)
Φ = Ψ (see full version)	#W[1]-hard (full version)	no FPTRAS (full version)	FPT (full version)
Φ = CLIQUE	#W[1]-hard ([34])	no FPTRAS (implicitly by [30])	W[1]-hard ([30])

[†] We assume that the minor-closed property Φ does not have bounded matching number, is not trivially true and that each forbidden minor has a vertex of degree at least 3.

[‡] Φ = true and Φ = false always yield fixed-parameter tractability of exact counting. Φ(H) = 1 ⇔ H is a matching yields #W[1]-hardness of exact counting [20]; note that the latter property is of bounded treewidth and satisfies the matching criterion.

[*] Φ = true always yields fixed-parameter tractability of exact counting. Φ(H) = 1 ⇔ (H is a matching or a star) yields #W[1]-hardness by Theorem 1.4; note that the latter property satisfies the matching criterion and the star criterion.

[§] Φ = true always yields an FPTRAS for approximate counting. Φ = Ψ (from the full version) does not allow for an FPTRAS while satisfying the matching criterion.

$\#\mathrm{EDGESUB}(\Phi)$ and $\mathrm{EDGESUB}(\Phi)$. In what follows, we say that a property Φ has *bounded matching number* if there is a constant bound on the size of a largest matching in graphs satisfying Φ.

▶ **Main Theorem 1.** *Let Φ denote a minor-closed graph property.*
1. Exact Counting*: If Φ is either trivially true or of bounded matching number, then the (exact) counting version $\#\mathrm{EDGESUB}(\Phi)$ is fixed-parameter tractable. Otherwise, the problem $\#\mathrm{EDGESUB}(\Phi)$ is $\#\mathsf{W}[1]$-hard. If, additionally, each forbidden minor of Φ has a vertex of degree at least 3, and the Exponential Time Hypothesis holds, then $\#\mathrm{EDGESUB}(\Phi)$ cannot be solved in time $f(k) \cdot |G|^{o(k/\log k)}$, for any function f.*
2. Approximate Counting*: The problem $\#\mathrm{EDGESUB}(\Phi)$ always has a fixed-parameter tractable randomised approximation scheme (FPTRAS).[2]*
3. Decision*: The problem $\mathrm{EDGESUB}(\Phi)$ is always fixed-parameter tractable.*

Consider for example the property Φ of being planar: planar graphs do not have bounded matching number. Additionally, by Kuratowski's Theorem, the forbidden minors of planar graphs are the 3-biclique $K_{3,3}$ and the 5-clique K_5. Since both $K_{3,3}$ and K_5 contain a vertex of degree at least 3, we conclude that computing the number of planar subgraphs with k edges in a graph G is $\#\mathsf{W}[1]$-hard and, assuming ETH, cannot be solved in time $f(k) \cdot |G|^{o(k/\log k)}$ for any function f. In sharp contrast, approximating the number of planar subgraphs with k edges in a graph, as well as deciding whether there is such a planar subgraph can be done efficiently. We obtain Main Theorem 1 as a combination of our (more general) results for each of the three settings that we study; we discuss these results next.

Results for Exact Counting

Let us return to the case of arbitrary graph properties Φ. Without any further assumptions on Φ, the naive algorithm for $\#\mathrm{EDGESUB}(\Phi)$ on the input (k, G) proceeds by enumerating the k-edge subsets of G and counting the number of cases where the corresponding subgraph satisfies Φ. This leads to a running time of the form $f(k) \cdot |V(G)|^{2k+O(1)}$. However, at least the linear constant in the exponent can be substantially improved using the currently fastest known algorithm for counting subgraphs with k edges due to Curticapean, Dell and Marx [21]. In the full version, we show that it easily extends to the case of $\#\mathrm{EDGESUB}(\Phi)$:

▶ **Proposition 1.1.** *Let Φ denote a computable graph property. Then $\#\mathrm{EDGESUB}(\Phi)$ can be solved in time $f(k) \cdot |V(G)|^{0.174k+o(k)}$, where f is some computable function.*

On the other hand, it was shown by Curticapean and Marx [22] that for the property Φ of being a matching, the problem $\#\mathrm{EDGESUB}(\Phi)$ cannot be solved in time $f(k) \cdot |V(G)|^{o(k/\log k)}$ for any function f, unless ETH fails. In other words, asymptotically and up to a factor of $1/\log k$, the exponent of $|V(G)|$ in the running time of $\#\mathrm{EDGESUB}(\Phi)$ cannot be improved without posing any restriction on Φ.

The goal is hence to identify properties Φ for which the algorithm in Proposition 1.1 *can* be (significantly) improved. In the best possible outcome, we hope to identify the properties for which the exponent of $|V(G)|$ does not depend on k; those cases are precisely the fixed-parameter tractable ones. An easy consequence of known results for subgraph counting (see for instance [22]) establishes the following tractability criterion; we include the proof only for the sake of completeness in the full version:

---

[2] The formal definition is given in the full version; intuitively an FPTRAS is the parameterized equivalent of a fully polynomial-time randomised approximation scheme (FPRAS).

▶ **Proposition 1.2.** *Let Φ denote a computable graph property satisfying that there is $M > 0$ such that for all k either the graphs with k edges satisfying Φ or the graphs with k edges satisfying $\neg\Phi$ have matching number bounded by M. Then* #EDGESUB(Φ) *is fixed-parameter tractable.*

Examples of properties satisfying the tractability criterion of Proposition 1.2 include, among others, the property of being a star, or the complement thereof. We conjecture that all remaining properties induce #W[1]-hardness and rule out any algorithm running in time $f(k) \cdot |G|^{o(k/\log k)}$ for any function f, unless ETH fails.[3] For the case of minor-closed graph properties, we have seen above that this conjecture holds.

Further, the techniques we develop to prove hardness of #EDGESUB(Φ) for minor-closed properties Φ in Main Theorem 1 can also be applied directly to show hardness for other specific properties Φ. Below, we record several natural examples of such properties which are covered by our methods.

▶ **Main Theorem 2.** *Consider the following graph properties.*
- $\Phi_C(H) = 1$ *if and only if H is connected.*
- $\Phi_H(H) = 1$ *if and only if H is Hamiltonian.*
- $\Phi_E(H) = 1$ *if and only if H is Eulerian.*
- $\Phi_{CF}(H) = 1$ *if and only if H is claw-free.*

For $\Phi \in \{\Phi_C, \Phi_H, \Phi_E, \Phi_{CF}\}$, the problem #EDGESUB($\Phi$) *is* #W[1]*-hard. Further, unless ETH fails, the problem* #EDGESUB(Φ) *cannot be solved in time $f(k) \cdot |G|^{o(k/\log k)}$ for any function f.*

Results for Approximate Counting and Decision

Our results on exact counting indicate that we have to relax the problem if we aim for tractability results for a larger variety of properties. One approach is to only ask for an *approximate* count of the number of k-edge subgraphs satisfying Φ. Tractability of approximation in the parameterized setting is given by the notion of a *fixed-parameter tractable randomized approximation scheme* (FPTRAS) as introduced by Arvind and Raman [5]. While we give the formal definition in full version, it suffices for now to think of an FPTRAS as a fixed-parameter tractable algorithm that can compute an arbitrarily good approximation of the answer with high probability. Readers familiar with the classical notions of approximate counting algorithms should think of an FPTRAS as an FPRAS in which we additionally allow a factor of $f(k)$ in the running time, for any computable function f.

For the statement of our results, we say that a property Φ satisfies the *matching criterion* if it is true for all but finitely many matchings, and we say that it satisfies the *star criterion* if it is true for all but finitely many stars. Furthermore, we say that Φ has bounded treewidth if there is a constant upper bound on the treewidth of graphs that satisfy Φ.

▶ **Main Theorem 3.** *Let Φ denote a computable graph property. If Φ satisfies the matching criterion **and** the star criterion, or if Φ has bounded treewidth, then* #EDGESUB(Φ) *admits an FPTRAS.*

For example, the property of being planar satisfies both, the star and the matching criterion. Moreover, we can show that every minor-closed graph property Φ has either bounded treewidth or satisfies matching and star criterion, and thus always admits an

---

[3] Note that it does not matter whether we choose $|G|$ or $|V(G)|$ for the size of the large graph since we care about the asymptotic behaviour of the exponent.

FPTRAS. Additionally, if not only exact but also approximate counting is intractable, we ask whether we can at least obtain an efficient algorithm for the decision version EDGESUB(Φ). Again, we obtain a tractability criterion; observe the subtle difference in the tractability criterion compared to Main Theorem 3.

▶ **Main Theorem 4.** *Let Φ denote a computable graph property. If Φ satisfies the matching criterion **or** the star criterion, or if Φ has bounded treewidth, then* EDGESUB(Φ) *is fixed-parameter tractable.*

As an easy corollary, we can conclude that for monotone, that is, subgraph-closed properties Φ, the problem EDGESUB(Φ) is always fixed-parameter tractable.[4]

For many previously studied problems, the complexity analysis of approximate counting and decision were related: often an algorithm solving one setting can be used to solve the other setting [51, 26]. However, in our results Main Theorems 3 and 4 we see an asymmetry between the two settings: it suffices for Φ to satisfy only one of the star and the matching criterion to induce tractability of the decision version, but we require satisfaction of both for approximate counting. One might expect that this reflects a shortcoming of our proof methods (and that in fact it suffices to check one of the criteria to have tractability of approximate counting). Interestingly, this is *not* the case:

▶ **Proposition 1.3.** *There is a computable graph property Ψ (see full version) that satisfies the matching criterion, but not the star criterion, such that* EDGESUB(Ψ) *is fixed-parameter tractable, but* #EDGESUB(Ψ) *does not admit an FPTRAS unless* W[1] *coincides with* FPT *(the class of all fixed-parameter tractable decision problems) under randomised parameterized reductions.*

Dichotomy for Evaluating a parameterized Tutte Polynomial

As a final part of the presentation of our main results, let us discuss our results on a parameterized Tutte polynomial.

The classical Tutte polynomial (as well as its specializations like the chromatic, flow or reliability polynomial) have received widespread attention, both from a combinatorial as well as a complexity theoretic perspective [42, 2, 65, 6, 37, 25, 10, 8]. The classical Tutte polynomial is of special interest from a complexity theoretic perspective, as the Tutte polynomial encodes a plethora of properties of a graph: prominent examples include the chromatic number, the number of acyclic orientations, and the number of spanning trees; we refer the reader to the work of Jaeger et al. [42] for a comprehensive overview. Formally, the Tutte polynomial is a bivariate graph polynomial defined as follows (see [42]):

$$T_G(x,y) := \sum_{A \subseteq E(G)} (x-1)^{k(A)-k(E(G))} \cdot (y-1)^{k(A)+\#A-\#V(G)} \,,$$

where $k(S)$ is the number of connected components of the graph $(V(G), S)$. In the aforementioned work, Jaeger et al. [42] also classified the complexity of evaluating the Tutte Polynomial in every pair of (complex) coordinates, that is, for every pair (a, b), the complexity of computing the function $G \mapsto T_G(a, b)$ is fully understood.

---

[4] Every graph property has either bounded treewidth or unbounded matching number. In the latter case, if the property is additionally monotone, it must satisfy the matching criterion.

In this work, we consider the following parameterized version of the Tutte Polynomial by restricting to edge-subsets A in G of size k:

$$T_G^k(x,y) := \sum_{A \in \binom{E(G)}{k}} (x-1)^{k(A)-k(E(G))} \cdot (y-1)^{k(A)+k-\#V(G)} .$$

We observe that the parameterized Tutte polynomial can be seen as a weighted version of counting small k-edge subgraph patterns by assigning to each k-edge subset A of G the weight

$$(x-1)^{k(A)-k(E(G))} \cdot (y-1)^{k(A)+k-\#V(G)} .$$

Moreover, we point out that $T_G^k(x,y)$ is related to a generalization of the bases generating function for matroids [4]. By establishing a so-called deletion-contraction recurrence, we show that $T_G^k(x,y)$ has similar expressive power as its classical counterpart $T_G(x,y)$:

▶ **Main Theorem 5.** *For any graph G and positive integer k, the following graph invariants are encoded in $T_G^k(x,y)$:*
1. *$T_G^k(2,1)$ is the number of k-forests in G. In other words $T_G^k(2,1)$ corresponds to the problem #EDGESUB(Φ) for the property Φ of being a forest.*
2. *For each positive integer c, the values of $T_G^k(1-c,0)$ determine[5] the numbers of pairs (A,σ), where A is a k-edge subset of G, and σ is a proper c-colouring of $(V(G),A)$.*
3. *From $T_G^k(2,0)$ we can compute the numbers of pairs $(A,\vec{\eta})$, where A is a k-edge subset of G, and $\vec{\eta}$ is an acyclic orientation of $(V(G),A)$.*
4. *$T_G^k(2,0)$ also determines the number of k-edge subsets A of G, such that $(V(G),A)$ has even Betti Number (we give a formal definition of the Betti number in the full version).*
5. *$T_G^k(0,2)$ determines the number of k-edge subsets A of G, such that $(V(G),A)$ has an even number of components.*

Note that, while #EDGESUB(Φ) only allows us to count the number of subgraphs with k edges that satisfy Φ, the parameterized Tutte polynomial allows us to count more intricate objects, such as tuples of an edge-subset and a colouring (or acyclic orientation) on the induced graph. From a complexity theoretic point of view, we obtain a similar result as [42], albeit only for rational coordinates: for each *fixed* pair (x,y) of coordinates, we consider the problem receiving as input a graph G and a positive integer k and computing $T_G^k(x,y)$. Following the paradigm of this work, we choose k as a parameter, that is, we consider inputs in which k is significantly smaller than $|G|$.

▶ **Main Theorem 6.** *Let (x,y) denote a pair of rational numbers. The problem of computing $T_G^k(x,y)$ is solvable in polynomial-time if $x=y=1$ or $(x-1)(y-1)=1$, fixed-parameter tractable, but #P-hard, if $x=1$ and $y \neq 1$, and #W[1]-hard otherwise.*

The class #P is the counting version of NP [63, 64] and, in particular, the #P-hard cases in the above classification are not polynomial-time tractable unless the polynomial-time hierarchy collapses to P [61]. Consider Figure 1 for a depiction of the classification. Note that Main Theorem 6 yields #W[1]-hardness for each of the aforementioned problems from Main Theorem 5. Note further, that the tractable cases are similar, but not equal to the classical counterpart [42].

---

[5] They are equal up to trivial modifications; in particular, their complexities coincide.
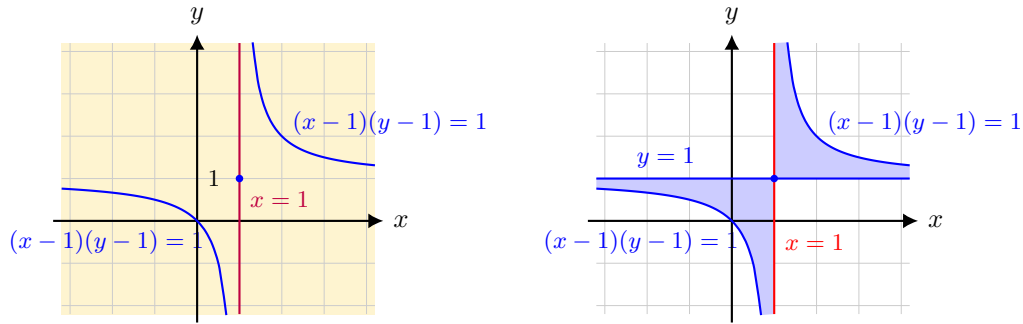
(a) Points of the parameterized Tutte polynomial that can be computed in polynomial-time (blue) and that are fixed-parameter tractable, but #P-hard (red). Exact computation at any other point (yellow) is #W[1]-hard.

(b) Points of the parameterized Tutte polynomial that allow for an FPRAS (blue) and for an FPTRAS (red); all points on the boundary of the blue area are included. The complexity of approximation is open for all points outside of the coloured region.

■ **Figure 1** Points of the parameterized Tutte polynomial that can be computed in FPT-time (a) exactly or (b) approximately. We emphasize that a full classification for exact counting is established, while the complexity of approximation remains open outside of the coloured area.

Moreover, our proof uses entirely different tools than [42] and illustrates the power and utility of the method presented in the subsequent discussion of our techniques.

Having fully classified the complexity of exact evaluation of the parameterized Tutte Polynomial, we also consider the complexity of approximate evaluation. We identify two regions bounded by the hyperbola $(x-1)(y-1) = 1$ and the lines $x = 1$ and $y = 1$ as efficiently approximable; consider Figure 1b for a depiction.

▶ **Main Theorem 7.** *Let (x, y) denote a pair of rational numbers. If $0 \leq (x-1)(y-1) \leq 1$, then $T_G^k(x, y)$ has an FPTRAS. If additionally $x \neq 1$ or $y = 1$ then $T_G^k(x, y)$ even has a fully polynomial-time randomized approximation scheme (FPRAS).*

Techniques

Our Main Theorems 3, 4, and 7 are obtained easily: the proof of Main Theorem 3 is a standard application (see for instance [51]) of the Monte-Carlo approach, in combination with Ramsey's theorem, and Arvind and Raman's algorithm for approximately counting subgraphs of bounded treewidth [5]. The proof of Main Theorem 4 uses a standard parameterized Win-Win approach for graphs of bounded treewidth or bounded degree. Finally, the proof of Main Theorem 7 is an easy consequence of the work of Anari et al. [4] on approximate counting via log-concave polynomials.

Hence, in this technical discussion, we want to focus on the technique that enables us to prove the lower bounds for Main Theorems 1 and 2 and, perhaps surprisingly, also for Main Theorem 6.

As a main component, we use the Complexity Monotonicity framework of Curticapean, Dell and Marx [20]. Given a property Φ and a positive integer k, we write #EdgeSub$(\Phi, k \rightarrow \star)$ for the function that maps a graph G to the number of k-edge subgraphs of G that satisfy Φ. Using a well-known transformation via Möbius inversion [49, Chapter 5.2], we can show that there are rational numbers $a_1, \ldots, a_\ell$ and graphs $H_1, \ldots, H_\ell$ such that for each graph G we have

$$\#\mathsf{EdgeSub}(\Phi, k \to G) = \sum_{i=1}^{k} a_i \cdot \#\mathsf{Hom}(H_i \to G)\,, \tag{1}$$

where $\#\mathsf{Hom}(H_i \to G)$ is the number of graph homomorphisms from H_i to G. In other words, we can express $\#\mathsf{EdgeSub}(\Phi, k \to \star)$ as a finite linear combination of homomorphism counts. Here, we can then apply the Complexity Monotonicity framework [21], which asserts that computing a finite linear combination of homomorphism counts is *precisely as hard as* its hardest term (among the terms with a non-zero coefficient). However, the complexity of computing the number of homomorphisms from small pattern graphs to large host graphs is very well-understood [24, 50]. Roughly speaking, the higher the treewidth of the pattern graph, the harder the problem becomes; we make this formal in the full version.

Instead of our original problem $\#\mathrm{EDGESUB}(\Phi)$, we can thus consider the problem of computing linear combinations of graph homomorphism counts. In particular, to obtain hardness, it suffices to understand for which of the coefficients in equation (1) we have $a_i \neq 0$, depending on k and Φ.

Relying on the well-known fact that the Möbius function of the partition lattice alternates in sign, Curticapean, Dell, and Marx [21] observed that non-trivial cancellations cannot occur in equation (1) if, for each k, every k-edge graph that satisfies Φ must have the same number of vertices. Consequently, if the matching number is unbounded, those properties yield $\#\mathsf{W}[1]$-hardness. An example for such a property is the case of $\Phi(H) = 1$ if and only if H is a tree. In contrast, the intractability result for the case of $\Phi = $ acyclicity (that is, being a forest) turned out to be much harder to show [11], indicated by connections to parameterized counting problems in matroid theory.

In later work, the coefficients a_i were shown to have even more interesting structure: the coefficients a_i describe topological and algebraic invariants of the set of pattern graphs. For example, in [57] it was shown that the coefficient of the k-clique in case of counting vertex-induced subgraphs with property Φ is the reduced Euler characteristic of a simplicial complex associated with Φ and can thus, if non-zero, be used to establish evasiveness of certain graph properties [46].

In this work, we prove additional insights into said coefficients a_i.[6] For any graph H we give an explicit formula for its coefficient a_H in terms of a sum over the *fractures* on H, an additional combinatorial structure on a graph H resembling, to some extent, a gadget construction used for the classification of the subgraph counting problem [22] (see full version for details). Our most crucial insight is then that we can drastically simplify the expression of the coefficient a_H modulo a prime ℓ if H admits a vertex-transitive action of a group of order given by a power of ℓ. In this case, we obtain an action of the group on the set of fractures on H and in the formula for a_H all contributions from fractures not fixed by the group cancel out modulo ℓ.

In particular, we consider graphs H which are Cayley graphs of a finite group of prime power order and a symmetric set of generators. Since the Cayley graph of a group always has a natural vertex-transitive action of this group, such Cayley graphs always have the desired symmetry properties. We exploit this by showing that there is a constant number of fractures fixed by the group action. This in turn allows us to write (a_H modulo ℓ) as a finite sum of terms depending on the value of Φ on some explicit graphs.

---

[6] For technical reasons, the approach we describe below requires us to consider a coloured version of $\#\mathrm{EDGESUB}(\Phi)$, which is, however, shown to be interreducible with the uncoloured one.
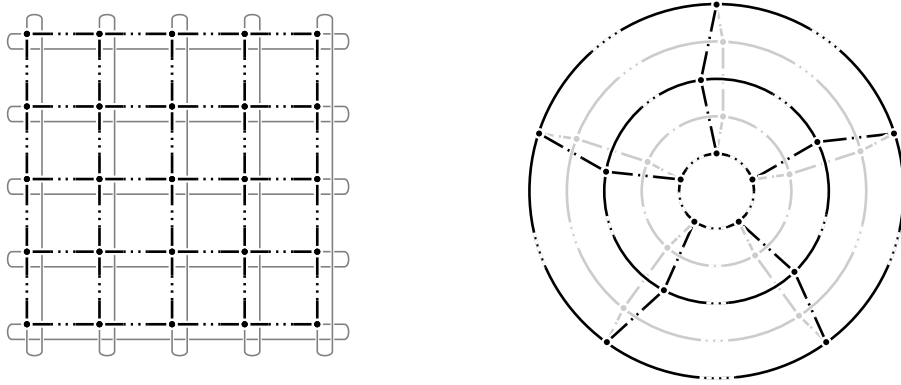
Figure 2 Two isomorphic representations of the toroidal grid $\circledcirc_\ell$: On the left hand side as a grid with connected endpoints, on the right hand side as a stylized torus.

Specifically, the first set of Cayley graphs we consider are the toroidal grids $\circledcirc_\ell$, which are depicted in Figure 2. Since the treewidth of $\circledcirc_\ell$ diverges with ℓ, we thus obtain a #W[1]-hardness result whenever the coefficient $a_{\circledcirc_\ell}$ does not vanish for infinitely many ℓ. Writing M_k for the matching of size k, P_2 for the path consisting of 2 edges, C_k for the cycle of length k, S_k for a sun (a cycle with dangling edges) of size k, and $\circledcirc_k$ for the toroidal grid of size k, our first main technical result reads as follows:

▶ **Theorem 1.4** (Simplified version). *Let Φ denote a computable graph property and assume that infinitely many primes ℓ satisfy the equation[7]*

$$-6\Phi(M_{2\ell^2}) + 4\Phi(M_{\ell^2} + \ell C_\ell) + 8\Phi(\ell^2 P_2) - \Phi(2\ell C_\ell) - 2\Phi(\ell C_{2\ell}) - 4\Phi(\ell S_\ell) + \Phi(\circledcirc_\ell) \neq 0 \mod \ell. \quad (2)$$

Then #EdgeSub(Φ) *is* #W[1]-*hard.*

As a toy example for an application of Theorem 1.4, let us consider the property Φ of being connected. Observe that among the graphs in (2), only $\circledcirc_\ell$ is connected, and thus the sum is always 1 for $\ell \geq 2$. Thus, indeed the left-hand side of (2) is nonzero, proving that #EdgeSub(Φ) is #W[1]-hard.

Using Theorem 1.4, we can prove most of the #W[1]-hardness results of Main Theorem 1. However, using the toroidal grid $\circledcirc_\ell$ we cannot prove (almost) tight conditional lower bounds: the treewidth of $\circledcirc_\ell$ grows only with the *square-root* of the parameter k (that is the number of edges of the graph). To address this problem, we consider a second family of 4-regular Cayley graphs, constructed explicitly by Peyerimhoff and Vdovina [54], which have the additional property of being *expander graphs*. In particular, for these graphs, the treewidth grows *linearly* in the number of edges. This allows us to obtain almost tight conditional lower bounds. The variant of Theorem 1.4 for these Cayley graph expanders can be found in the full version of this paper.

The only drawback of the Cayley graphs from [54] is that the corresponding groups always have orders given by powers of 2 (in contrast to having arbitrary primes ℓ in Theorem 1.4). Hence, our criterion for hardness is the nonvanishing of some expression modulo 2. Ultimately, this is the reason why for the conditional lower bounds in Main Theorem 1 we need to exclude forbidden minors having a vertex of degree 2 or less.

---

[7] We write $+$ for (disjoint) graph union and ℓH for the graph consisting of ℓ disjoint copies of H. Further, we set $\Phi(H) = 1$ if H satisfies Φ and $\Phi(H) = 0$ otherwise.

Finally, to obtain Main Theorem 6, we express the parameterized Tutte polynomial at a rational point (x, y) as a linear combination of (fractures of) toroidal grids; the proof of Theorem 1.4 then essentially shows that this linear combination always contains a graph with unbounded treewidth, yielding #W[1]-hardness.

──── **References** ────

1 Noga Alon, Phuong Dao, Iman Hajirasouliha, Fereydoun Hormozdiari, and S. Cenk Sahinalp. Biomolecular network motif counting and discovery by color coding. *Bioinformatics*, 24(13):i241–i249, July 2008. `doi:10.1093/bioinformatics/btn163`.

2 Noga Alon, Alan M. Frieze, and Dominic Welsh. Polynomial Time Randomized Approximation Schemes for Tutte-Gröthendieck Invariants: The Dense Case. *Random Struct. Algorithms*, 6(4):459–478, 1995. `doi:10.1002/rsa.3240060409`.

3 Noga Alon, Raphael Yuster, and Uri Zwick. Color-Coding. *J. ACM*, 42(4):844–856, 1995. `doi:10.1145/210332.210337`.

4 Nima Anari, Kuikui Liu, Shayan Oveis Gharan, and Cynthia Vinzant. Log-concave polynomials II: high-dimensional walks and an FPRAS for counting bases of a matroid. In Moses Charikar and Edith Cohen, editors, *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, pages 1–12. ACM, 2019. `doi:10.1145/3313276.3316385`.

5 Vikraman Arvind and Venkatesh Raman. Approximation Algorithms for Some Parameterized Counting Problems. In *Algorithms and Computation, 13th International Symposium, ISAAC 2002 Vancouver, BC, Canada, November 21-23, 2002, Proceedings*, pages 453–464, 2002. `doi:10.1007/3-540-36136-7_40`.

6 Andreas Björklund, Thore Husfeldt, Petteri Kaski, and Mikko Koivisto. Computing the Tutte Polynomial in Vertex-Exponential Time. In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25-28, 2008, Philadelphia, PA, USA*, pages 677–686. IEEE Computer Society, 2008. `doi:10.1109/FOCS.2008.40`.

7 Andreas Björklund, Thore Husfeldt, Petteri Kaski, and Mikko Koivisto. Narrow sieves for parameterized paths and packings. *J. Comput. Syst. Sci.*, 87:119–139, 2017. `doi:10.1016/j.jcss.2017.03.003`.

8 Andreas Björklund and Petteri Kaski. The Fine-Grained Complexity of Computing the Tutte Polynomial of a Linear Matroid. Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Alexandria, VA, USA, January 10-13, 2021, to appear.

9 Cornelius Brand, Holger Dell, and Thore Husfeldt. Extensor-coding. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 151–164, 2018. `doi:10.1145/3188745.3188902`.

10 Cornelius Brand, Holger Dell, and Marc Roth. Fine-Grained Dichotomies for the Tutte Plane and Boolean #CSP. *Algorithmica*, 81(2):541–556, 2019. `doi:10.1007/s00453-018-0472-z`.

11 Cornelius Brand and Marc Roth. Parameterized Counting of Trees, Forests and Matroid Bases. In *Computer Science - Theory and Applications - 12th International Computer Science Symposium in Russia, CSR 2017, Kazan, Russia, June 8-12, 2017, Proceedings*, pages 85–98, 2017. `doi:10.1007/978-3-319-58747-9_10`.

12 Ashok K. Chandra and Philip M. Merlin. Optimal Implementation of Conjunctive Queries in Relational Data Bases. In *Proceedings of the 9th Annual ACM Symposium on Theory of Computing, May 4-6, 1977, Boulder, Colorado, USA*, pages 77–90, 1977. `doi:10.1145/800105.803397`.

13 Hubie Chen and Stefan Mengel. Counting Answers to Existential Positive Queries: A Complexity Classification. In *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2016, San Francisco, CA, USA, June 26 - July 01, 2016*, pages 315–326, 2016. `doi:10.1145/2902251.2902279`.

14   Jianer Chen, Benny Chor, Mike Fellows, Xiuzhen Huang, David W. Juedes, Iyad A. Kanj, and Ge Xia. Tight lower bounds for certain parameterized NP-hard problems. *Inf. Comput.*, 201(2):216–231, 2005. `doi:10.1016/j.ic.2005.05.001`.

15   Jianer Chen, Xiuzhen Huang, Iyad A. Kanj, and Ge Xia. Strong computational lower bounds via parameterized complexity. *J. Comput. Syst. Sci.*, 72(8):1346–1367, 2006. `doi:10.1016/j.jcss.2006.04.007`.

16   Jianer Chen, Joachim Kneis, Songjian Lu, Daniel Mölle, Stefan Richter, Peter Rossmanith, Sing-Hoi Sze, and Fenghui Zhang. Randomized Divide-and-Conquer: Improved Path, Matching, and Packing Algorithms. *SIAM Journal on Computing*, 38(6):2526–2547, 2009. `doi:10.1137/080716475`.

17   Yijia Chen, Marc Thurley, and Mark Weyer. Understanding the Complexity of Induced Subgraph Isomorphisms. In *Automata, Languages and Programming, 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings, Part I: Tack A: Algorithms, Automata, Complexity, and Games*, pages 587–596, 2008. `doi:10.1007/978-3-540-70575-8_48`.

18   Stephen A. Cook. The Complexity of Theorem-Proving Procedures. In *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing, May 3-5, 1971, Shaker Heights, Ohio, USA*, pages 151–158, 1971. `doi:10.1145/800157.805047`.

19   Derek G. Corneil and C. C. Gotlieb. An Efficient Algorithm for Graph Isomorphism. *J. ACM*, 17(1):51–64, 1970. `doi:10.1145/321556.321562`.

20   Radu Curticapean. Counting matchings of size k is w[1]-hard. In Fedor V. Fomin, Rusins Freivalds, Marta Z. Kwiatkowska, and David Peleg, editors, *Automata, Languages, and Programming - 40th International Colloquium, ICALP 2013, Riga, Latvia, July 8-12, 2013, Proceedings, Part I*, volume 7965 of *Lecture Notes in Computer Science*, pages 352–363. Springer, 2013. `doi:10.1007/978-3-642-39206-1_30`.

21   Radu Curticapean, Holger Dell, and Dániel Marx. Homomorphisms are a good basis for counting small subgraphs. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 210–223, 2017. `doi:10.1145/3055399.3055502`.

22   Radu Curticapean and Dániel Marx. Complexity of Counting Subgraphs: Only the Boundedness of the Vertex-Cover Number Counts. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 130–139, 2014. `doi:10.1109/FOCS.2014.22`.

23   Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. `doi:10.1007/978-3-319-21275-3`.

24   Víctor Dalmau and Peter Jonsson. The complexity of counting homomorphisms seen from the other side. *Theor. Comput. Sci.*, 329(1-3):315–323, 2004. `doi:10.1016/j.tcs.2004.08.008`.

25   Holger Dell, Thore Husfeldt, Dániel Marx, Nina Taslaman, and Martin Wahlen. Exponential Time Complexity of the Permanent and the Tutte Polynomial. *ACM Trans. Algorithms*, 10(4):21:1–21:32, 2014. `doi:10.1145/2635812`.

26   Holger Dell, John Lapinskas, and Kitty Meeks. Approximately counting and sampling small witnesses using a colourful decision oracle. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 2201–2211. SIAM, 2020. `doi:10.1137/1.9781611975994.135`.

27   Holger Dell, Marc Roth, and Philip Wellnitz. Counting Answers to Existential Questions. In *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece*, pages 113:1–113:15, 2019. `doi:10.4230/LIPIcs.ICALP.2019.113`.

28   Julian Dörfler, Marc Roth, Johannes Schmitt, and Philip Wellnitz. Counting Induced Subgraphs: An Algebraic Approach to #W[1]-hardness. In Peter Rossmanith, Pinar Heggernes, and Joost-Pieter Katoen, editors, *44th International Symposium on Mathematical Foundations of Computer Science, MFCS 2019, August 26-30, 2019, Aachen, Germany*, volume 138 of *LIPIcs*, pages 26:1–26:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. `doi:10.4230/LIPIcs.MFCS.2019.26`.

1none

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

29 Rodney G. Downey and Michael R. Fellows. Fixed-Parameter Tractability and Completeness I: Basic Results. *SIAM J. Comput.*, 24(4):873–921, 1995. `doi:10.1137/S0097539792228228`.

30 Rodney G. Downey and Michael R. Fellows. Fixed-Parameter Tractability and Completeness II: On Completeness for W[1]. *Theor. Comput. Sci.*, 141(1&2):109–131, 1995. `doi:10.1016/0304-3975(94)00097-3`.

31 Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013. `doi:10.1007/978-1-4471-5559-1`.

32 Arnaud Durand and Stefan Mengel. Structural Tractability of Counting of Solutions to Conjunctive Queries. *Theory Comput. Syst.*, 57(4):1202–1249, 2015. `doi:10.1007/s00224-014-9543-y`.

33 Jack Edmonds. Paths, Trees, and Flowers. *Canadian Journal of Mathematics*, 17:449–467, 1965. `doi:10.4153/CJM-1965-045-4`.

34 Jörg Flum and Martin Grohe. The Parameterized Complexity of Counting Problems. *SIAM J. Comput.*, 33(4):892–922, 2004. `doi:10.1137/S0097539703427203`.

35 Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2006. `doi:10.1007/3-540-29953-X`.

36 Fedor V. Fomin, Daniel Lokshtanov, and Saket Saurabh. Efficient Computation of Representative Sets with Applications in Parameterized and Exact Algorithms. In Chandra Chekuri, editor, *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 142–151. SIAM, 2014. `doi:10.1137/1.9781611973402.10`.

37 Leslie Ann Goldberg and Mark Jerrum. The Complexity of Computing the Sign of the Tutte Polynomial. *SIAM J. Comput.*, 43(6):1921–1952, 2014. `doi:10.1137/12088330X`.

38 Joshua A. Grochow and Manolis Kellis. Network Motif Discovery Using Subgraph Enumeration and Symmetry-Breaking. In Terry Speed and Haiyan Huang, editors, *Research in Computational Molecular Biology*, pages 92–106, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

39 Martin Grohe. Parameterized Complexity for the Database Theorist. *SIGMOD Rec.*, 31(4):86–96, 2002. `doi:10.1145/637411.637428`.

40 Martin Grohe, Thomas Schwentick, and Luc Segoufin. When is the evaluation of conjunctive queries tractable? In *Proceedings on 33rd Annual ACM Symposium on Theory of Computing, July 6-8, 2001, Heraklion, Crete, Greece*, pages 657–666, 2001. `doi:10.1145/380752.380867`.

41 Russell Impagliazzo and Ramamohan Paturi. On the Complexity of k-SAT. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001. `doi:10.1006/jcss.2000.1727`.

42 F. Jaeger, Dirk L. Vertigan, and Dominic J. A. Welsh. On the computational complexity of the Jones and Tutte polynomials. *Mathematical Proceedings of the Cambridge Philosophical Society*, 108(1):35–53, 1990. `doi:10.1017/S0305004100068936`.

43 Mark Jerrum and Kitty Meeks. The parameterised complexity of counting connected subgraphs and graph motifs. *J. Comput. Syst. Sci.*, 81(4):702–716, 2015. `doi:10.1016/j.jcss.2014.11.015`.

44 Mark Jerrum and Kitty Meeks. Some Hard Families of Parameterized Counting Problems. *TOCT*, 7(3):11:1–11:18, 2015. `doi:10.1145/2786017`.

45 Mark Jerrum and Kitty Meeks. The parameterised complexity of counting even and odd induced subgraphs. *Combinatorica*, 37(5):965–990, 2017. `doi:10.1007/s00493-016-3338-5`.

46 Jeff Kahn, Michael E. Saks, and Dean Sturtevant. A topological approach to evasiveness. *Combinatorica*, 4(4):297–306, 1984. `doi:10.1007/BF02579140`.

47 Subhash Khot and Venkatesh Raman. Parameterized complexity of finding subgraphs with hereditary properties. *Theor. Comput. Sci.*, 289(2):997–1008, 2002. `doi:10.1016/S0304-3975(01)00414-5`.

48 Bingkai Lin. The Parameterized Complexity of the *k*-Biclique Problem. *J. ACM*, 65(5):34:1–34:23, 2018. `doi:10.1145/3212622`.

49 László Lovász. *Large Networks and Graph Limits*, volume 60 of *Colloquium Publications*. American Mathematical Society, 2012. URL: `http://www.ams.org/bookstore-getitem/item=COLL-60`.

50 Dániel Marx. Can You Beat Treewidth? *Theory of Computing*, 6(1):85–112, 2010. `doi:10.4086/toc.2010.v006a005`.

51 Kitty Meeks. The challenges of unbounded treewidth in parameterised subgraph counting problems. *Discrete Applied Mathematics*, 198:170–194, 2016. `doi:10.1016/j.dam.2015.06.019`.

52 R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network Motifs: Simple Building Blocks of Complex Networks. *Science*, 298(5594):824–827, 2002. `doi:10.1126/science.298.5594.824`.

53 Ron Milo, Shalev Itzkovitz, Nadav Kashtan, Reuven Levitt, Shai Shen-Orr, Inbal Ayzenshtat, Michal Sheffer, and Uri Alon. Superfamilies of evolved and designed networks. *Science*, 303(5663):1538–1542, 2004. `doi:10.1126/science.1089167`.

54 Norbert Peyerimhoff and Alina Vdovina. Cayley graph expanders and groups of finite width. *J. Pure Appl. Algebra*, 215(11):2780–2788, 2011. `doi:10.1016/j.jpaa.2011.03.018`.

55 Jürgen Plehn and Bernd Voigt. Finding minimally weighted subgraphs. In Rolf H. Möhring, editor, *Graph-Theoretic Concepts in Computer Science, 16rd International Workshop, WG '90, Berlin, Germany, June 20-22, 1990, Proceedings*, volume 484 of *Lecture Notes in Computer Science*, pages 18–29. Springer, 1990. `doi:10.1007/3-540-53832-1_28`.

56 Ofer Rahat, Uri Alon, Yaakov Levy, and Gideon Schreiber. Understanding hydrogen-bond patterns in proteins using network motifs. *Bioinformatics*, 25(22):2921–2928, September 2009. `doi:10.1093/bioinformatics/btp541`.

57 Marc Roth and Johannes Schmitt. Counting Induced Subgraphs: A Topological Approach to #W[1]-hardness. *Algorithmica*, 82(8):2267–2291, 2020. `doi:10.1007/s00453-020-00676-9`.

58 Marc Roth, Johannes Schmitt, and Philip Wellnitz. Counting Small Induced Subgraphs Satisfying Monotone Properties. 61th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, to appear.

59 Benjamin Schiller, Sven Jager, Kay Hamacher, and Thorsten Strufe. StreaM - A Stream-Based Algorithm for Counting Motifs in Dynamic Graphs. In Adrian-Horia Dediu, Francisco Hernández-Quiroz, Carlos Martín-Vide, and David A. Rosenblueth, editors, *Algorithms for Computational Biology*, pages 53–67, Cham, 2015. Springer International Publishing.

60 Falk Schreiber and Henning Schwöbbermeyer. Frequency Concepts and Pattern Detection for the Analysis of Motifs in Networks. In Corrado Priami, Emanuela Merelli, Pablo Gonzalez, and Andrea Omicini, editors, *Transactions on Computational Systems Biology III*, pages 89–104, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

61 Seinosuke Toda. PP is as Hard as the Polynomial-Time Hierarchy. *SIAM J. Comput.*, 20(5):865–877, 1991. `doi:10.1137/0220053`.

62 Julian R. Ullmann. An algorithm for subgraph isomorphism. *J. ACM*, 23(1):31–42, 1976. `doi:10.1145/321921.321925`.

63 Leslie G. Valiant. The Complexity of Computing the Permanent. *Theor. Comput. Sci.*, 8:189–201, 1979. `doi:10.1016/0304-3975(79)90044-6`.

64 Leslie G. Valiant. The Complexity of Enumeration and Reliability Problems. *SIAM J. Comput.*, 8(3):410–421, 1979. `doi:10.1137/0208032`.

65 Dirk Vertigan. Bicycle Dimension and Special Points of the Tutte Polynomial. *J. Comb. Theory Ser. B*, 74(2):378–396, 1998. `doi:10.1006/jctb.1998.1860`.