Quasipolynomiality of the Smallest Missing Induced Subgraph

David Eppstein [1] *Andrea Lincoln* [2] *Virginia Vassilevska Williams* [3]

[1]Computer Science Department, University of California, Irvine
[2]Boston University
[3]EECS and CSAIL, Massachusetts Institute of Technology

Abstract. We study the problem of finding the smallest graph that does not occur as an induced subgraph of a given graph. This missing induced subgraph has at most logarithmic size and can be found by a brute-force search, in an n-vertex graph, in time $n^{O(\log n)}$. We show that under the Exponential Time Hypothesis this quasipolynomial time bound is optimal. We also consider variations of the problem in which either the missing subgraph or the given graph comes from a restricted graph family; for instance, we prove that the smallest missing planar induced subgraph of a given planar graph can be found in polynomial time.

1 Introduction

In recent years, the older conventional wisdom that all natural computational problems have time either polynomial (or better) or exponential (or worse) has frayed, with the discovery of several natural problems whose best known time complexity is *quasipolynomial*, of the form $n^{f(n)}$ for a polylogarithmically bounded function f. Of course, it is possible to obtain quasipolynomial time bounds from parameterized time bounds, by an appropriate choice of parameter, but instead we seek problems in which quasipolynomiality arises naturally from an unparameterized problem. Such problems for which the best known time is quasipolynomial include:

- Finding minimum dominating sets in tournaments [35]

- Computing the Vapnik–Chervonenkis dimension of a family of sets [31,37,41]. Approximating VC dimension up to any constant factor [34].

E-mail addresses: eppstein@uci.edu (David Eppstein)   virgi@mit.edu (Virginia Vassilevska Williams)

- Constructing ϵ-approximate Nash equilibria, for constant ϵ [32].

- Graph isomorphism and graph canonization [6].

- Maximum independent set in unit ball graphs in the hyperbolic plane [27].

More strongly, in some of these cases, the form of the quasipolynomial time bound has been precisely determined (up to constant factors in the exponent) under the *Exponential Time Hypothesis* (ETH) of Impagliazzo and Paturi [26], the assumption that 3-SAT cannot be solved in time $2^{o(n)}$. For instance, for both ϵ-approximate Nash equilibria and hyperbolic unit ball independent sets, the upper bound is $n^{O(\log n)}$, while ETH implies the impossibility of improving this bound to $n^{o(\log n)}$ [9, 27].

   In this work, we find another natural example of a problem that has both a quasipolynomial upper bound and a matching lower bound derived from the Exponential Time Hypothesis. Our problem is the *Smallest Missing Induced Subgraph*: given an undirected graph G as input, find another undirected graph H, with as few vertices as possible, such that H is not isomorphic to an induced subgraph of G. As we show, this problem can be solved in time $n^{O(\log n)}$ in n-vertex graphs G (Theorem 1), and improving this to time $n^{o(\log n)}$ would contradict ETH (Theorem 2).

   We also consider variations of the problem in which either the missing subgraph or the given graph comes from a restricted graph family. For instance, the smallest missing planar induced subgraph, in a given graph G, measures the quality of G as a *universal graph* for planar graphs and as the basis for adjacency-labeling schemes for planar graphs [17]. We prove that the smallest missing planar induced subgraph of a given planar graph can be found in polynomial time (Theorem 3).

2   Upper bounds

The Smallest Missing Induced Subgraph can be found by exactly counting the number of induced subgraphs of each type, of increasing size, until finding a size for which one of the counts is zero. Related problems of counting all types of subgraphs, exactly or approximately, have long been studied [18, 28, 40], and have recently gained popularity in bioinformatics, where small induced subgraphs are called "graphlets" [1, 25, 38]. A naïve counting algorithm can solve this problem simply by enumerating all ordered k-tuples of vertices, determining the type of subgraph each k-tuple induces, and incrementing a counter for that type of subgraph.

   In more detail:

- We begin with $k = 2$, the smallest possible size of a missing induced subgraph in a graph with two or more vertices.

- A k-vertex subgraph can be represented by a binary value with $\binom{k}{2}$ bits, representing (the lower half of) its adjacency matrix. We use these binary values as indices into an array of counters, initially all zero. We consider the vertices of the subgraphs to be labeled, so that permutations of the vertices are considered as different subgraphs.

- We loop through all ordered k-tuples of vertices of the given graph. For each k-tuple, we can construct the binary value representing its adjacency matrix in average $O(k)$ time per k-tuple, by updating it from the previous k-tuple in the enumeration. After this step, we increment the counter indexed by that binary value. Note that by going through every k-tuple in the graph we are incrementing at least once for every possible re-labeling of each subgraph on k nodes. In other words, we can not 'miss' a valid labeling as we are going through all n^k possible combinations of nodes.

- We loop through all cells of the array of counters, searching for one that is zero. If found, we decode it and output it as a missing subgraph; otherwise, we increment k and repeat the previous steps.

The time for each k is $O(kn^k + 2^{k(k-1)/2})$. As this increases superexponentially with k, it is dominated by the last step, the one in which we find a missing induced subgraph.

Lemma 1 *This algorithm succeeds for some $k \le 2\log_2 n + 2$.*

Proof: If it ever reaches $k > 2\log_2 n + 1$, the number of array cell counters will be

$$2^{\binom{k}{2}} = 2^{k(k-1)/2} > 2^{k\log_2 n} = n^k.$$

There are more counters than increment steps, and some counter will remain unincremented. Therefore, the algorithm must necessarily terminate whenever k becomes this large. Since k is increased by one in each iteration, the value at which it exceeds this threshold is at most $2\log_2 n + 2$. □

Corollary 1 *In any n-vertex graph, the smallest missing induced subgraph has at most $2\log_2 n + 2$ vertices.*

Constructions of Alon [2] and Alstrup, Kaplan, Thorup, and Zwick [4] produce graphs in which the smallest missing induced subgraph has $2\log_2 n - O(1)$ vertices, showing that this bound is nearly tight. By plugging this value of k into the time bound for the algorithm as a function of n and k, and simplifying the resulting expression, we obtain:

Theorem 1 *The algorithm above takes time $O(n^{2\log_2 n + 3})$ to find the smallest missing induced subgraph of any n-vertex graph.*

The bottleneck in this time bound is initializing and searching the table of counters. When $k = 2\log_2 n + 2$, the size of this table is sufficiently larger than n^k to make this step slower than the search through all k-tuples of vertices. The time can be reduced to $O(n^{2\log_2 n + 2}\log n)$, at the expense of randomization, by replacing the table of counters by a hash table; we omit the details.

The copies of any single k-vertex subgraph type can be counted in the substantially smaller time bound $O(n^{0.174k + o(k)})$ [15] but multiplying this bound by the number of distinct subgraph types would lead to a slower algorithm than the one above. Although there has been research on applying fast matrix multiplication to the induced subgraph counting problem for specific small values of k [28, 29], using sparsity parameters of the host graph to reduce the time for subgraph counting [21, 23], or using the structural features of specific types of subgraphs to speed up algorithms for counting only those subgraphs [43], we do not know of significant general speedups to the naïve counting algorithm above based on these ideas. Indeed, to make any significant improvement based on counting methods we would need not only algorithmic ideas, but a better bound than Corollary 1 on the smallest k for which success is guaranteed or a counting algorithm which takes less than $O(1)$ time per possible subgraph of size k. This is because, for the value of k given by Corollary 1, our time bound is not significantly larger than what would be required merely to output all subgraph counts.
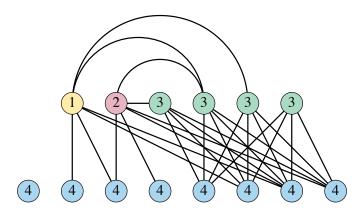
Figure 1: The labeled graph X_4 of Lemma 3. Every graph with four or fewer vertices exists as an induced subgraph of X_4, except for a clique of size four. More strongly, whenever H is a four-vertex graph H labeled by the four numbers 1, 2, 3, 4 that does not have an edge labeled 1–2, H appears as a labeled induced subgraph of X_4.

3   Lower bounds

Our lower bound on the smallest missing induced subgraph follows from the hardness of finding cliques of logarithmic size, a special case of known results on the parameterized complexity of the clique problem [11, 12, 33]. We spell out the proof to clarify its applicability to the range of parameter values that we need. We follow the proof outline from Lokshtanov et al [33].

Lemma 2 *Under the Exponential Time Hypothesis, for any constant $c > 0$, it is impossible to find a maximum clique, in an n-vertex graph for which that clique is guaranteed to have size at most $c \log n$, in time $n^{o(\log n)}$.*

Proof: We reduce 3-coloring of N-vertex graphs to clique-finding on exponentially-larger graphs, as follows. Choose a parameter $\delta > 0$, and partition the given N-vertex graph G into $t = \lfloor \delta \sqrt{N} \rfloor$ subsets of at most $\lceil N/t \rceil$ vertices. Form a graph H representing the possible 3-colorings of each subset, and add an edge in H between pairs of compatible subset 3-colorings. (Two subset 3-colorings are compatible if they represent proper colorings of the induced subgraphs of different subsets, and they do not combine to form any monochromatic edge.) Then cliques of size t in H correspond to proper 3-colorings of G. The number n of vertices in H is $O(t3^{N/t})$, so $t = \Theta(\log n)$, with a constant of proportionality that can be made arbitrarily small by choosing a sufficiently small value of δ, meeting the guarantee of the lemma. If we could find a t-clique in time $n^{o(\log n)}$, this would give us a 3-coloring algorithm with time $(t3^{N/t})^{o(t)} = 3^{o(N)}$, contradicting the exponential-time hypothesis. □

To reduce from clique-finding to missing subgraphs, we use a construction for a family of graphs in which the unique smallest missing subgraph is a clique:

Lemma 3 *For each positive integer $i > 1$, there exists a graph X_i that does not contain a clique of size i, but contains all other i-vertex induced subgraphs, and has a total of $2^i - 2$ vertices.*

Proof: More strongly, we form a family of labeled graphs X_i with the following properties:

- X_i has its vertices labeled by the numbers from 1 to i. It has one vertex labeled 1 and one vertex labled 2, and 2^{j-1} vertices labeled j for $j > 2$.

- The vertices having the same label form an independent set, so each clique must have vertices of distinct labels.

- There is no edge labeled 1–2, so it is not possible for both of these labels to appear in a clique. Therefore, the i-vertex clique is a missing induced subgraph.

- If H is any i-vertex graph H labeled by distinct integers from 1 to i, such that H does not have an edge labeled 1–2, then H appears with the same labels as an induced subgraph of X_i.

Every i-vertex graph that is not a clique can be labeled by distinct integers from 1 to i so that its vertices labeled 1 and 2 are non-adjacent; therefore, for a graph X_i with the properties listed above, every i-vertex graph that is not a clique appears as an induced subgraph, and the i-vertex clique is the unique smallest missing induced subgraph.

We construct the graphs X_i inductively, as illustrated in Fig. 1. As a base case, for $i = 2$, let X_2 consist of two isolated vertices, one with labeled 1 and the other labeled 2. For ease of notation call the set of all vertices labeled i T_i. To form X_i from X_{i-1} add 2^{i-1} vertices, labeled i, one for each subset of $\{1, 2, \ldots, i-1\}$. For a vertex v labeled i corresponding to the subset $S \in \{1, 2, \ldots, i-1\}$, add edges between v and u for all $u \in T_\ell$ for all $\ell \in S$. That is, connect v to all verticies u that have a label in S.

The resulting graph clearly has the first three properties listed above. To show that it has the last property, let H be any i-vertex graph H labeled by distinct integers from 1 to i, such that H does not have an edge labeled 1–2. Let H^- be obtained from H by removing the vertex labeled i, and let S be the set of labels of neighbors of the removed vertex. Then by induction, H^- appears as an induced subgraph of X_{i-1}, with the same labels. To form H as an induced subgraph of X_i, add one more vertex to this copy of H^-, the vertex with label i that corresponds to set S.   □

Noga Alon (personal communication) has observed that a randomized construction from his work on induced-universal graphs [2] can be modified to produce smaller graphs with the same properties as in Lemma 3, with $2^{(i-1)/2}(1 + o(1))$ vertices. This size bound is within a lower-order additive term of optimal. However, for our purposes it is more important to have a deterministic construction than to optimize the exponent.

Theorem 2 *Under the exponential-time hypothesis, it is impossible to find the smallest missing induced subgraph of every n-vertex graph in time $n^{o(\log n)}$.*

Proof: Let X_i denote the family of graphs constructed by Lemma 3. We can use an algorithm for the smallest missing induced subgraph to find a maximum clique of size up to $\log_2 n$, in a graph G for which the maximum clique has that size, by successively finding the smallest missing induced subgraph in the disjoint union of graphs $G \cup X_i$ for $i = 2, 3, \ldots$ until finding an i for which this missing subgraph is an i-vertex clique. The maximum clique in G has size one less than this stopping value of i. The addition of X_i to G at most doubles the number of vertices, which does not change the form of a time bound of form $n^{o(\log n)}$, and the loop through successive values of i also does not change this form. Therefore, an algorithm for the smallest missing induced subgraph with time $n^{o(\log n)}$ would contradict Lemma 2.   □

4   Restricted missing subgraphs

A natural extension of the smallest missing induced subgraph would be to search for the smallest missing induced subgraph from a given graph family $\mathcal{F}$, either in a larger graph from the same family or more generally. When $\mathcal{F}$ is totally ordered by the induced subgraph relation, this is equivalent to finding the largest induced subgraph in $\mathcal{F}$ in a given graph G; for instance, the classical problems of finding the largest clique, largest independent set, and longest induced path in a graph can all be expressed as a missing induced subgraph problem in this way.

For general graph families $\mathcal{F}$, the size of smallest missing induced subgraph in $\mathcal{F}$, in a graph G, is one plus the largest k for which G is k-*induced-universal*, meaning that it contains all k-vertex graphs in $\mathcal{F}$ as induced subgraphs. A universal graph in this sense is not required to belong to $\mathcal{F}$ itself. Universality is equivalent to the existence of an *adjacency labeling scheme*, an assignment of binary labels to the vertices of graphs in $\mathcal{F}$ such that the adjacency of two vertices can be determined from their labels. If a k-universal graph exists, and has n_k vertices, the identities of its vertices can be used as labels in an adjacency labeling scheme with $\lceil \log_2 n \rceil$ bits per label. Conversely, if an adjacency labeling scheme with b bits per label is possible, then one can construct a k-universal graph with 2^b vertices, one for each possible label, with adjacency determined according to the labeling scheme. The computational problem of finding the smallest missing induced subgraph from $\mathcal{F}$ in G, then, is the same as determining the k for which G is k-universal. If the smallest missing subgraph from $\mathcal{F}$ in G is of size $k+1$ then all subgraphs of size k appear G making G k-universal (for the family $\mathcal{F}$). Thus, known upper bounds on the size of universal graphs translate into lower bounds on the possible size of the smallest missing subgraph. For instance, trees have universal graphs of linear size, from which it follows that the smallest missing tree, in a given graph, may also have linear size [14]. Translating other known results on universal graphs into this form, it follows that the smallest missing induced subgraph of maximum degree Δ can have size $\Theta(n^{2/\Delta})$ [3] and that the smallest missing planar induced subgraph can have size $n^{1-o(1)}$ [17].

Additional problems of this type arise when both the induced subgraph and G are restricted to belong to the same family $\mathcal{F}$. For instance, for the smallest missing induced bipartite subgraph of a given bipartite graph, a counting argument similar to the one we used for general graphs shows that this subgraph has size $O(\log n)$, from which it follows that we can find this subgraph by a brute-force search in quasipolynomial time. However, our ability to prove the optimality of this time bound is limited in two ways: we do not have a construction for a bipartite graph in which the unique smallest missing induced bipartite subgraph is a biclique, analogous to our construction for cliques, and we do not know ETH-tight bounds on the complexity of finding bicliques in bipartite graphs [30].

Another interesting case is the problem of finding the smallest missing induced planar subgraph, in a given planar graph. Like the unconstrained smallest missing induced subgraph, the smallest planar missing induced subgraph in a planar graph has logarithmic size. However, in this case the search for the subgraph can take advantage of planarity, speeding up the overall algorithm to polynomial time. In the remainder of this section we detail this result.

Lemma 4 *The smallest missing induced planar subgraph, in a given n-vertex planar graph, has $O(\log n)$ vertices.*

Proof: The high level idea is to compare the exponential number of 4-connected planar triangulations (that is, maximal planar graphs) of size k, on the one hand, and an $O(n)$ bound on the number of 4-connected triangulations that can exist as induced subgraphs of a given n-vertex graph, on the other hand. When all k-vertex 4-connected planar triangulations are present in a graph,

their exponential number must be $O(n)$, from which it follows that k must be $O(\log n)$. Although the smallest missing induced subgraph might not be 4-connected or maximal planar, it has at most k vertices.

In more detail, the number of 4-connected planar triangulations on k unlabeled vertices can be upper-bounded by c^k for some constant c [39, 42]. However, a given planar graph G, with n vertices, can only have $O(n)$ induced subgraphs that are 4-connected and maximal planar. This is because each such subgraph is one of the linearly many graphs formed when G is partitioned along its separating triangles, which necessarily form a nested family of separations [22]. This partition cannot subdivide any 4-connected triangulation, because if it could be subdivided in this way it would not be 4-connected. Conversely, if G contains a 4-connected triangulation as an induced subgraph, each face of the triangulation must either be a face of G or a separating triangle, so this triangulation will be separated from all the other vertices of G by the partition of G on its separating triangles. Putting the two bounds on the number of k-vertex triangulations and on the number of induced 4-connected triangulations together, if G is an n-vertex graph for which all k-vertex planar graphs are present as induced subgraphs, it must be the case that $c^k = O(n)$. Taking logarithms of both sides shows that $k = O(\log n)$. For the induced subgraphs of any larger size, at least one is missing. $\qquad\square$

Although we do not need it, we note that conversely there exist planar graphs whose smallest missing induced subgraph has size $\Omega(\log n)$. This follows from the fact that the number of k-vertex unlabeled planar graphs of all types is only singly exponential in k [24]. An n-vertex planar graph whose smallest missing planar induced subgraph is logarithmic can be constructed by choosing the largest k such that the disjoint union of all k-vertex planar graphs has at most n vertices, and then padding this disjoint union to exactly n vertices by adding isolated vertices.

Theorem 3 *The smallest missing induced planar subgraph, in a given n-vertex planar graph G, can be found in time polynomial in n.*

Proof: For each positive integer k, in numerical order, we list all k-vertex planar graphs and test for each one whether it is an induced subgraph of G. The k-vertex unlabeled planar graphs can be listed in total time exponential in k using standard planar graph enumeration methods [8, 10]. An algorithm of Eppstein [20], subsequently improved by Dorn [16] and Bonsma [7], allows testing whether a graph H of size k is a subgraph or induced subgraph of a graph G of size n, in time $2^{O(k)}n$. Because we only apply these algorithms for $k = O(\log n)$, the total time bound is polynomial in n. $\qquad\square$

The constant factor in the logarithmic bound on the size of the smallest missing induced planar subgraph that can be obtained from our proof of Lemma 4 is unlikely to be tight, and the base of the exponential function in the $2^{O(k)}n$ time bound for planar subgraph isomorphism is large (approximately 76^k in Bonsma's version). For this reason we have not analyzed the time bound of Theorem 3 more carefully to determine its exact exponent.

5   Conclusions and Related Problems

We have shown that, to find the smallest missing induced subgraph of a given n-vertex graph, it is possible to use an algorithm with running time $n^{O(\log n)}$, and that under standard assumptions no better time bound is possible. We have also investigated similar problems where the induced subgraph or the given graph must belong to a more restricted family of graphs.

More generally, the problem of asking for the smallest missing subobject of an object can be extended to many other inclusion orderings of other types of objects. For substrings or subsequences, it is not difficult to solve in polynomial time. A more interesting example of the same type of problem arises for permutations and permutation patterns, smaller permutations with the same ordering as a subsequence of a larger permutation. A permutation that contains all patterns of length up to some value k is known as a *superpattern*, and so the problem of finding the smallest missing permutation is equivalent to asking for what k a given pattern is a superpattern. A counting argument shows that a k-superpattern must have length $\Omega(k^2)$, and superpatterns of this length are known [5, 13, 19, 36]. Turning this around, any permutation of length n must have a missing pattern of length $O(\sqrt{n})$, from which it follows that this missing pattern can be found in time $n^{O(\sqrt{n})}$. Is a time bound of this form tight?

References

[1] N. K. Ahmed, J. Neville, R. A. Rossi, and N. Duffield. Efficient graphlet counting for large networks. In *2015 IEEE International Conference on Data Mining*, November 2015. `doi:10.1109/icdm.2015.141`.

[2] N. Alon. Asymptotically optimal induced universal graphs. *Geom. Funct. Anal.*, 27(1):1–32, 2017. `doi:10.1007/s00039-017-0396-9`.

[3] N. Alon and R. Nenadov. Optimal induced universal graphs for bounded-degree graphs. *Math. Proc. Cambridge Philos. Soc.*, 166(1):61–74, 2019. `doi:10.1017/S0305004117000706`.

[4] S. Alstrup, H. Kaplan, M. Thorup, and U. Zwick. Adjacency labeling schemes and induced-universal graphs. *SIAM J. Discrete Math.*, 33(1):116–137, 2019. `doi:10.1137/16M1105967`.

[5] R. Arratia. On the Stanley–Wilf conjecture for the number of permutations avoiding a given pattern. *Electron. J. Combin.*, 6:N1, 1999. `doi:10.37236/1477`.

[6] L. Babai. Canonical form for graphs in quasipolynomial time: preliminary report. In M. Charikar and E. Cohen, editors, *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23–26, 2019*, pages 1237–1246, 2019. `doi:10.1145/3313276.3316356`.

[7] P. S. Bonsma. Surface split decompositions and subgraph isomorphism in graphs on surfaces. In C. Dürr and T. Wilke, editors, *29th International Symposium on Theoretical Aspects of Computer Science, STACS 2012, February 29th – March 3rd, 2012, Paris, France*, volume 14 of *LIPIcs*, pages 531–542. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2012. `doi: 10.4230/LIPIcs.STACS.2012.531`.

[8] P. Bose and F. Hurtado. Flips in planar graphs. *Comput. Geom.*, 42(1):60–80, 2009. `doi: 10.1016/j.comgeo.2008.04.001`.

[9] M. Braverman, Y. Kun-Ko, and O. Weinstein. Approximating the best Nash equilibrium in $n^{o(\log n)}$-time breaks the Exponential Time Hypothesis. In P. Indyk, editor, *Proceedings of the 26th Annual ACM–SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4–6, 2015*, pages 970–982, 2015. `doi:10.1137/1.9781611973730.66`.

[10] G. Brinkmann and B. D. McKay. Fast generation of planar graphs. *MATCH Commun. Math. Comput. Chem.*, 58(2):323–357, 2007.

[11] J. Chen, B. Chor, M. Fellows, X. Huang, D. Juedes, I. A. Kanj, and G. Xia. Tight lower bounds for certain parameterized NP-hard problems. *Inform. and Comput.*, 201(2):216–231, 2005. `doi:10.1016/j.ic.2005.05.001`.

[12] J. Chen, X. Huang, I. A. Kanj, and G. Xia. Strong computational lower bounds via parameterized complexity. *J. Comput. System Sci.*, 72(8):1346–1367, 2006. `doi:10.1016/j.jcss.2006.04.007`.

[13] Z. Chroman, M. Kwan, and M. Singhal. Lower bounds for superpatterns and universal sequences. *J. Combinatorial Theory Ser. A*, 182:105467, 2021. `doi:10.1016/j.jcta.2021.105467`.

[14] F. R. K. Chung. Universal graphs and induced-universal graphs. *J. Graph Theory*, 14(4):443–454, 1990. `doi:10.1002/jgt.3190140408`.

[15] R. Curticapean, H. Dell, and D. Marx. Homomorphisms are a good basis for counting small subgraphs. In H. Hatami, P. McKenzie, and V. King, editors, *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`.

[16] F. Dorn. Planar subgraph isomorphism revisited. In J. Marion and T. Schwentick, editors, *27th International Symposium on Theoretical Aspects of Computer Science, STACS 2010, March 4–6, 2010, Nancy, France*, volume 5 of *LIPIcs*, pages 263–274. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2010. `doi:10.4230/LIPIcs.STACS.2010.2460`.

[17] V. Dujmović, L. Esperet, C. Gavoille, G. Joret, P. Micek, and P. Morin. Adjacency labelling for planar graphs (and beyond). *J. ACM*, 68(6):A42:1–A42:33, 2021. `doi:10.1145/3477542`.

[18] R. A. Duke, H. Lefmann, and V. Rödl. A fast approximation algorithm for computing the frequencies of subgraphs in a given graph. *SIAM J. Comput.*, 24(3):598–620, 1995. `doi:10.1137/S0097539793247634`.

[19] M. Engen and V. Vatter. Containing all permutations. *Amer. Math. Monthly*, 128(1):4–24, 2021. `doi:10.1080/00029890.2021.1835384`.

[20] D. Eppstein. Subgraph isomorphism in planar graphs and related problems. *J. Graph Algorithms & Applications*, 3(3):1–27, 1999. `doi:10.7155/jgaa.00014`.

[21] D. Eppstein, M. T. Goodrich, D. Strash, and L. Trott. Extended dynamic subgraph statistics using *h*-index parameterized data structures. *Theoret. Comput. Sci.*, 447:44–52, 2012. `doi:10.1016/j.tcs.2011.11.034`.

[22] D. Eppstein and B. Reed. Finding maximal sets of laminar 3-separators in planar graphs in linear time. In T. M. Chan, editor, *Proceedings of the 30th Annual ACM–SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6–9, 2019*, pages 589–605. Society for Industrial and Applied Mathematics, 2019. `doi:10.1137/1.9781611975482.37`.

[23] D. Eppstein and E. S. Spiro. The *h*-index of a graph and its application to dynamic subgraph statistics. *J. Graph Algorithms & Applications*, 16(2):543–567, 2012. `doi:10.7155/jgaa.00273`.

[24] O. Giménez and M. Noy. Asymptotic enumeration and limit laws of planar graphs. *J. Amer. Math. Soc.*, 22(2):309–329, 2009. `doi:10.1090/S0894-0347-08-00624-3`.

[25] T. Hočevar and J. Demšar. A combinatorial approach to graphlet counting. *Bioinformatics*, 30(4):559–565, February 2014. `doi:10.1093/bioinformatics/btt717`.

[26] R. Impagliazzo and R. Paturi. Complexity of k-SAT. In *Proceedings of the 14th Annual IEEE Conference on Computational Complexity, Atlanta, Georgia, USA, May 4-6, 1999*, pages 237–240, 1999. `doi:10.1109/CCC.1999.766282`.

[27] S. Kisfaludi-Bak. Hyperbolic intersection graphs and (quasi)-polynomial time. In S. Chawla, editor, *Proceedings of the 31st Annual ACM–SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5–8, 2020*, pages 1621–1638, 2020. `doi:10.1137/1.9781611975994.100`.

[28] T. Kloks, D. Kratsch, and H. Müller. Finding and counting small induced subgraphs efficiently. *Inform. Process. Lett.*, 74(3-4):115–121, 2000. `doi:10.1016/S0020-0190(00)00047-8`.

[29] M. Kowaluk, A. Lingas, and E.-M. Lundell. Counting and detecting small subgraphs via equations. *SIAM J. Discrete Math.*, 27(2):892–909, 2013. `doi:10.1137/110859798`.

[30] B. Lin. The parameterized complexity of the k-biclique problem. *J. ACM*, 65(5):A34:1–A34:21, 2018. `doi:10.1145/3212622`.

[31] N. Linial, Y. Mansour, and R. L. Rivest. Results on learnability and the Vapnik-Chervonenkis dimension. *Inform. and Comput.*, 90(1):33–49, 1991. `doi:10.1016/0890-5401(91)90058-A`.

[32] R. J. Lipton, E. Markakis, and A. Mehta. Playing large games using simple strategies. In D. A. Menascé and N. Nisan, editors, *Proceedings of the 4th ACM Conference on Electronic Commerce (EC-2003), San Diego, California, USA, June 9–12, 2003*, pages 36–41, 2003. `doi:10.1145/779928.779933`.

[33] D. Lokshtanov, D. Marx, and S. Saurabh. Lower bounds based on the exponential time hypothesis. *Bull. Eur. Assoc. Theor. Comput. Sci. EATCS*, 105:41–71, 2011.

[34] P. Manurangsi. Improved inapproximability of VC dimension and littlestone's dimension via (unbalanced) biclique. In Y. T. Kalai, editor, *14th Innovations in Theoretical Computer Science Conference, ITCS 2023, January 10-13, 2023, MIT, Cambridge, Massachusetts, USA*, volume 251 of *LIPIcs*, pages 85:1–85:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023. `doi:10.4230/LIPIcs.ITCS.2023.85`.

[35] N. Megiddo and U. Vishkin. On finding a minimum dominating set in a tournament. *Theoret. Comput. Sci.*, 61(2-3):307–316, 1988. `doi:10.1016/0304-3975(88)90131-4`.

[36] A. Miller. Asymptotic bounds for permutations containing many different patterns. *J. Combinatorial Theory Ser. A*, 116(1):92–108, 2009. `doi:10.1016/j.jcta.2008.04.007`.

[37] C. H. Papadimitriou and M. Yannakakis. On limited nondeterminism and the complexity of the V-C dimension. *J. Comput. System Sci.*, 53(2):161–170, 1996. `doi:10.1006/jcss.1996.0058`.

[38] N. Pržulj. Biological network comparison using graphlet degree distribution. *Bioinformatics*, 23(2):e177–e183, January 2007. `doi:10.1093/bioinformatics/btl301`.

[39] P. N. Rathie. A census of simple planar triangulations. *J. Combinatorial Theory Ser. B*, 16:134–138, 1974. `doi:10.1016/0095-8956(74)90055-0`.

[40] P. Ribeiro, P. Paredes, M. E. P. Silva, D. Aparicio, and F. Silva. A survey on subgraph counting. *ACM Computing Surveys*, 54(2):1–36, March 2022. `doi:10.1145/3433652`.

[41] A. Shinohara. Complexity of computing Vapnik-Chervonenkis dimension and some generalized dimensions. *Theoret. Comput. Sci.*, 137(1):129–144, 1995. `doi:10.1016/0304-3975(94)00164-E`.

[42] W. T. Tutte. A census of planar triangulations. *Canadian J. Math.*, 14:21–38, 1962. `doi:10.4153/CJM-1962-002-9`.

[43] V. Vassilevska Williams and R. Williams. Finding, minimizing, and counting weighted subgraphs. *SIAM J. Comput.*, 42(3):831–854, 2013. `doi:10.1137/09076619X`.