

Induced Cycles and Paths Are Harder Than You Think

Mina Dalirrooyfard
EECS
MIT
Cambridge, USA
minad@mit.edu

Virginia Vassilevska Williams
EECS
MIT
Cambridge, USA
virgi@mit.edu

Abstract—The goal of the paper is to give fine-grained hardness results for the Subgraph Isomorphism (SI) problem for fixed size induced patterns H , based on the k -Clique hypothesis that the current best algorithms for Clique are optimal.

Our first main result is that for any pattern graph H that is a *core*, the SI problem for H is at least as hard as t -Clique, where t is the size of the largest clique minor of H . This improves (for cores) the previous known results [Dalirrooyfard-Vassilevska W. STOC'20] that the SI for H is at least as hard as k -clique where k is the size of the largest clique *subgraph* in H , or the chromatic number of H (under the Hadwiger conjecture). For detecting *any* graph pattern H , we further remove the dependency of the result of [Dalirrooyfard-Vassilevska W. STOC'20] on the Hadwiger conjecture at the cost of a sub-polynomial decrease in the lower bound.

The result for cores allows us to prove that the SI problem for induced k -Path and k -Cycle is harder than previously known. Previously [Floderus et al. Theor. CS 2015] had shown that k -Path and k -Cycle are at least as hard to detect as a $bk/4 - O(1)$ -Clique, improving the conditional lower bound exponent by a factor of $3/2$. This shows for instance that the known $O(n^5)$ combinatorial algorithm for 7-cycle detection is conditionally tight.

Finally, we provide a new conditional lower bound for detecting induced 4-cycles: $n^{2-o(1)}$ time is necessary even in graphs with n nodes and $O(n^{1.5})$ edges. The 4-cycle is the smallest induced pattern whose running time is not well-understood. It can be solved in matrix multiplication, $O(n^\omega)$ time, but no conditional lower bounds were known until ours. We provide evidence that certain types of reductions from triangle detection to 4-Cycle would not be possible. We do this by studying a new problem called Paired Pattern Detection.

Index Terms—component, formatting, style, styling, insert

I. INTRODUCTION

A fundamental problem in graph algorithms, Subgraph Isomorphism (SI) asks, given two graphs G and H , does G contain a subgraph isomorphic to H ? While the problem is easily NP-complete, many applications only need to solve the poly-time solvable version in which the pattern H has constant size; this version of SI is often called Graph Pattern Detection and is the topic of this paper.

There are two versions of SI: *induced* and not necessarily induced, *non-induced* for short. In the induced version, the copy of H in G must have both edges and non-edges preserved, whereas in the non-induced version only edges need

to carry over, and the copy of H in G can be an arbitrary supergraph of H . It is well-known that the induced version of H -pattern detection for any H of constant size is at least as hard as the non-induced version (see e.g. [16]), and that often the non-induced version of SI has faster algorithms (e.g. the non-induced k -independent set problem is solvable in constant time).

It is well-known that the SI problem for any k -node pattern H in n -node graphs for constant k , can be reduced in linear time to detecting a k -clique in an $O(n)$ node graph (see [30]). Thus the hardest pattern to detect is k -clique. A natural question is:

How does the complexity of detecting a particular fixed size pattern H compare to that of k -clique?

Let us denote by $C(n, k)$ the best running time for k -clique detection in an n node graph. When k is divisible by 3, Nešetřil and Poljak [30] showed that $C(n, k) \leq O(n^{\omega_{k/3}})$ time, where $\omega < 2.37286$ [2] is the matrix multiplication exponent. For k not divisible by 3, $C(n, k) \leq O(n^{\omega(bk/3c, dk/3e, d(k-1)/3e)})$ time, where $\omega(a, b, c)$ is the exponent of multiplying an $n^a \times n^b$ by an $n^b \times n^c$ matrix.

This k -clique running time has remained unchallenged since the 1980s, and a natural hardness hypothesis has emerged (see e.g. [33]):

Hypothesis 1 (k -clique Hypothesis): On a word-RAM with $O(\log n)$ bit words, for every constant $k \geq 3$, k -clique requires $n^{\omega(bk/3c, dk/3e, d(k-1)/3e) - o(1)}$ time.

A “combinatorial” version¹ of the hypothesis states that the best combinatorial algorithm for k -clique runs in $n^{k-o(1)}$ time. Other hypotheses such as the Exponential Time Hypothesis for SAT [10], [22] imply weaker versions of the k -Clique Hypothesis, namely that k -clique requires $n^{\Omega(k)}$ time [11]. We will focus on the *fine-grained* k -Clique Hypothesis as we are after fine-grained lower bounds that focus on fixed exponents.

Our goal is now, for every k -vertex pattern H , determine a function $f(H)$ such that detecting H in an n -vertex graph is at least as hard (in a fine-grained sense, see [33]) as detecting

¹“Combinatorial” is not well-defined, but it is a commonly used term to denote potentially practical algorithms that avoid the generally impractical Strassen-like methods for matrix multiplication.

Identify applicable funding agency here. If none, delete this.

an $f(H)$ -clique in an n -vertex graph. We then say that H is “at least as hard as $f(H)$ -clique”.

Obtaining such results is interesting for several reasons.

- First, under the k -clique Hypothesis, we would get *fine-grained* lower bounds for detecting H . This would give us a much tighter handle on the complexity of H -detection than, say, results (such as results based on ETH, or [28]) that merely provide an $n^{\Omega(k)}$ lower bound which only talks about the *growth* of the exponent.
- Second, knowing the largest size clique that limits the complexity of H -pattern detection can allow us to compare between different patterns. The goal is to get to something like: the complexity of k -node H_1 is like the complexity of $k/10$ -clique, whereas the complexity of k -node H_2 is like the complexity of $k/2$ -clique, so H_2 seems harder.
- Third, this more structural approach uncovers interesting combinatorial and graph theoretic results. For instance, in [16] it was uncovered that the colorability of a pattern, and the Hadwiger conjecture can explain the hardness of pattern detection. This is not obvious at all apriori.

This approach has been taken by prior work (e.g. [5], [16], [20]); see the related work section for more background.

II. OUR RESULTS

Our contributions are as follows:

- 1) First, we obtain a strengthening of a recent result of [16] that implies that the hardness of certain patterns called “cores” relates to the size of their maximum *clique minor*. This hardness is stronger than what was previously known, as previously only the chromatic number, or the maximum size of a clique *subgraph* were known to imply limitations, and both of these parameters are upper-bounded by the clique minor size (under the Hadwiger conjecture, for chromatic number).
- 2) We then apply the result above to obtain much higher hardness for *induced Path and Cycle* detection in graphs: a k -path or k -cycle contains an independent set of size roughly $k/2$. Thus both k -Cycle and k -Path were shown [20] to be at least as hard as $bk/2c$ -Clique. We raise the hardness to that of $3k/4 - O(1)$ clique, thus raising the *exponent* of the lower bound running time by a factor of $3/2$. This allows us for instance to obtain a *tight* conditional lower bound of $n^{5-o(1)}$ for the running time of combinatorial algorithms for 7-Clique; an $O(n^5)$ algorithm was obtained by Bläser et al. [6].
- 3) Finally, we consider the smallest known case of induced k -Cycle whose complexity is not well-understood: induced 4-Cycle. We provide a new conditional lower bound for the problem in sparser graphs based on a popular fine-grained hypothesis, and also provide some explanation for why reductions from triangle detection to 4-Cycle have failed so far.

We now elaborate on our results.

a) *New results for core graphs.*: Dalirrooyfard, Vuong and Vassilevska W. [16] related the hardness of subgraph pattern detection to the size of the maximum clique or the chromatic number of the pattern. In particular, they showed that if H has chromatic number t , then under the Hadwiger conjecture, H is at least as hard to detect as a t -clique.

The Hadwiger conjecture basically states that the chromatic number of a graph is always at most the largest size of a clique minor of the graph. As the result of [16] was already assuming the Hadwiger conjecture, one might wonder if it can be extended to show that every pattern H is at least as hard to detect as an η -clique, where η is the size of the largest *clique minor* of H .

We first note that such an extension is highly unlikely to work for non-induced patterns: the four-cycle C_4 has a K_3 (triangle) minor, but a non-induced C_4 has an $O(n^2)$ time detection algorithm that does not use matrix multiplication, whereas any subcubic triangle detection algorithm must use (Boolean) matrix multiplication [35]. Thus any extension of the result that shows clique-minor-sized clique hardness would either only work for certain types of non-induced graphs, or will need to only work in the induced case.

Here we are able to show that H -subgraph pattern detection, even in the non-induced case, is at least as hard as η -clique, where η is the largest clique minor size of H , as long as H is a special type of pattern called a *core*. Cores include many patterns of interest, including the complements of cycles of odd length. We also give several other hardness results, such as removing the dependence on the Hadwiger conjecture from some of the results of [16] with only a slight loss in the lower bound.

We call a subgraph C of a graph H a *core* of H if there is a homomorphism $H \rightarrow C$ but there is no homomorphism $H \rightarrow C^0$ for any proper subgraph C^0 of C . Hell and Nešetřil [21] showed that every graph has a unique core (up to isomorphism), and the core of a graph is an *induced* subgraph. We denote the core of a graph H by $\text{core}(H)$. A graph which is its own core is called simply a *core*.

We prove strong hardness results for cores, relating the hardness of detecting the pattern to the size of its maximum clique minor. We then relate the hardness of detecting arbitrary patterns to the hardness of detecting their cores.

We begin with a theorem that shows hardness for detecting a “partitioned” copy of a pattern H . Here the vertex set of the host graph $G^\boxplus$ is partitioned into k parts, and one is required to detect an induced copy of a k -node H so that the image of the i th node of H is in the i th part of the vertex set of $G^\boxplus$. This version of SI is often called Partitioned Subgraph Isomorphism (PSI). Marx [29] showed that under ETH, PSI for a pattern H requires at least $n^{\Omega(tw(H)/\log tw(H))}$ time where $tw(H)$ is the treewidth of H . We give a more fine-grained lower bound for PSI. We provide a reduction from η -clique detection in an n node graph to PSI for a graph H in an $O(n)$ node host graph, for any H with maximum clique minor of size η .

Theorem 2.1: (Hardness of PSI) Let H be a k -node pattern with maximum clique minor of size $\eta(H)$, and let G be an

n -node graph. Then one can construct a k -partite $O(n)$ -node graph $G^\boxminus$ in $O(n^2)$ time such that $G^\boxminus$ has a colorful copy of H if and only if G has a clique of size $\eta(H)$.

Thus the hardness of Partitioned SI is related to the size of the largest clique minor. To obtain a bound on the size of the maximum clique minor of any graph we use a result of Thomason [32] as follows: Let $c(t)$ be the minimum number such that every graph H with $|E(H)| \geq c(t)|V(H)|$ has a K^t minor. Then $c(t) = (\alpha + o(1))t^{\frac{1}{\log t}}$, where $\alpha \leq 0.32$ is an explicit constant. Since for $t = \frac{|E(H)|/|V(H)|}{\log(|E(H)|/|V(H)|)}$ the above inequality is true, we have the following corollary.

Corollary 2.1: Let H be a k -node m -edge pattern. Then the problem of finding a partitioned copy of H in an n -node k -partite graph is at least as hard as finding a clique of size $V_{\log} \frac{m/k}{m/k}$ in an $O(n)$ -node graph.

Hence, for example if $m = ck^2$ for some constant c , then the PSI problem for H cannot be solved in $n^{O(\frac{k}{\log k})}$ time. Thus, for dense enough graphs, we improve the lower bound of $n^{\Omega(tw(H)/\log tw(H))}$ due to Marx [29], since $tw(H) \leq k$.

While Theorem 2.1 only applies to PSI, one can use it to obtain hardness for SI as well, as long as H is a core. In particular, Marx [29] showed that PSI and SI are equivalent on cores. Thus we obtain:

Corollary 2.2: (Hardness of cores in SI) Let G be an n -node m -edge graph and let H be a k -node pattern with maximum clique minor of size $\eta(H)$. If H is a core, then one can construct a graph $G^\boxminus$ with at most $O(n)$ vertices in $O(m+n)$ time such that $G^\boxminus$ has a subgraph isomorphic to H if and only if G has a $\eta(H)$ -clique as a subgraph.

As the complements of odd cycles are cores with a clique minor of size at least $b3k/4c$, for C_k when k is odd, we immediately obtain a lower bound of $C(n, b3k/4c)$ for C_k detection. When k is even, more work is needed.

Corollary 2.2 applies to the non-induced version of SI. We obtain a stronger result for the induced version in terms of the k and the size of the largest clique subgraph.

Corollary 2.3: (Hardness for induced-SI for cores) Let H be a k -node pattern which is a core. Suppose that $w(H)$ is the size of the maximum clique in H . Then detecting H in an n -node graph as an induced subgraph is at least as hard as detecting a clique of size $\max\{d^{\frac{1}{2}}(k+2w(H))/2e, d^{\frac{1}{2}}k/1.95\}e$.

For comparison, the result of [16] shows that non-induced SI for any k -node H is at least as hard as detecting a clique of size $\frac{k}{2}$, but the result is conditioned on the Hadwiger conjecture. Corollary 2.3 is the strongest known clique-based lower bound result for k -node core H that is not conditioned on the Hadwiger conjecture.

Our next theorem relates the hardness of detecting a pattern to the hardness of detecting its core.

Theorem 2.2: Let G be an n -node m -edge graph and let H be a k -node pattern. Let C be the core of H . Then one can construct a graph $G^\boxminus$ with at most $O(n)$ vertices in $O(n^2)$ time such that $G^\boxminus$ has a subgraph isomorphic to H if and only if

G has a subgraph isomorphic to C , with high probability².

One consequence of Theorem 2.2 and Corollary 2.3 is that induced-SI for any pattern H of size k is at least as hard as detecting a clique of size $dk^{1/4}/1.39e$. Note that this is the first lower bound for induced SI that is only under the k -clique hypothesis.

Corollary 2.4: (Hardness of Induced-SI) For any k -node pattern H , detecting an induced copy of H in an n -node graph is at least as hard as detecting a clique of size $dk^{1/4}/1.39e$ in an $O(n)$ graph.

b) *Hardness for induced cycles and paths.*: We now focus on k -paths P_k and k -cycles C_k for fixed k and provide highly improved fine-grained lower bounds for their detection under the K -clique Hypothesis (for k larger than some constant). The results can be viewed as relating how close induced paths and cycles are to cliques. Our techniques for proving our results can be of independent interest and can potentially be implemented to get stronger hardness results for other classes of graphs.

The fastest known algorithms for finding induced cycles or paths on k nodes can be found in Table I. For larger k , the best known algorithms are either the k -clique running time $C(n, k)$, or an $O(n^{k-2})$ time combinatorial algorithm by [5]. For $k \leq 7$, slightly faster algorithms are known.

The best known conditional lower bounds so far [20] under the k -clique hypothesis stem from the fact that the complement of C_k contains a $b3k/4c$ -clique, and the complement of P_k contains a $dk/2e$ -clique. These lower bounds show that the best known running time of $O(n^w)$ for C_5 and P_5 are likely optimal. Unfortunately, for larger k , these lower bounds are far from the best known running times.

We obtain polynomially higher lower bounds, raising the lower bound exponent from roughly $k/2$ to roughly $3k/4$.

Theorem 2.3: (Hardness of P_k and C_k) Let H be the complement of a P_k or the complement of a C_k . Suppose that t is the size of the maximum clique minor of H . Then the problem of detecting H in an $O(n)$ -node graph is at least as hard as finding a $(t-2)$ -clique in an n -node graph. If k is odd, then detecting an induced C_k is at least as hard as finding a t -clique.

The largest *clique minor*³ of the complement of C_k has size $b3k/4c$ and of the complement of P_k has size at least $b(3k+1)/4c$.

Table I summarizes our new lower bounds. Aside from obtaining a much higher conditional lower bound, our result shows that the best known combinatorial algorithm for C_7 detection is tight, unless there is a faster combinatorial algorithm for 5-clique detection. For algorithms that may be non-combinatorial, our lower bound for C_7 is at least $\Omega(n^{4.08})$ assuming that the current bound for 5-clique is optimal.

c) *The curious case of Four-Cycle.*: The complexity of SI for all patterns on at most 3 nodes in n -node graphs is well-understood, both in the induced and non-induced case:

²with probability $1/\text{poly } n$

³A t -clique minor of a graph H is a decomposition of H into t connected subgraphs such that there is at least one edge between any two subgraphs.

TABLE I
KNOWN UPPER AND LOWER BOUNDS FOR PATHS AND CYCLES. THE ALGORITHMS FOR P_3 AND C_3 ARE FOLKLORE, FOR P_4 FROM [12], FOR C_4 FROM [34], AND FOR C_k AND P_k FOR $k \geq 5$ FROM [5]. THE OLD LOWER BOUNDS ARE ALL FROM [20]; OUR NEW LOWER BOUNDS APPEAR IN THEOREM 2.3.

Pattern	Runtime	Lower Bound	Comb. Runtime	Comb. Lower Bound
P_3, P_4	$O(m + n)$		$O(m + n)$	
C_4	$\tilde{O}(n^\omega)$		$O(n^3)$	
C_3, C_5, P_5	$\tilde{O}(n^\omega)$	$n^{\omega-o(1)}$	$O(n^3)$	$n^{3-o(1)}$
C_6, P_6	$C(n, 4)$ $= O(n^{\omega(2,1,1)})$	$n^{\omega-o(1)}$	$O(n^4)$	$n^{3-o(1)}$
P_7	$C(n, 6)$ $= O(n^{2\omega})$	$n^{\omega-o(1)}$	$O(n^5)$	$n^{3-o(1)}$
C_7	$C(n, 6)$ $= O(n^{2\omega})$	$C(n, 5)$ [new] $n^{\omega-o(1)}$ [old]	$O(n^5)$	$n^{5-o(1)}$ [new] $n^{3-o(1)}$ [old]
$C_k, P_k, k \geq 8$ $C_k, \text{ odd } k$	$C(n, k)$ $C(n, k)$	$C(n, b3k/4c - 2)$ [new] $C(n, b3k/4c)$ [new] $C(n, dk/2e)$ [old]	$O(n^{k-2})$ $O(n^{k-2})$	$n^{b3k/4c-2-o(1)}$ [new] $n^{b3k/4c-o(1)}$ [new] $n^{dk/2e-o(1)}$ [old]

all patterns except the triangle and (in the induced case) the independent set can be detected in $O(n^2)$ time, whereas the triangle (and independent set in the induced case) can be detected in $O(n^\omega)$ time where $\omega < 2.373$ is the exponent of matrix multiplication [2]. The dependence on (Boolean) matrix multiplication for triangle detection was proven to be necessary [35].

Table 1 gives the best known algorithms and conditional lower bounds for induced SI for all 4-node patterns. In the non-induced case, the change is that, except for the 4-clique K_4 , the diamond, co-claw and the paw whose runtimes and conditional lower bounds stay the same, all other patterns can be solved in $O(n^2)$ time.

All conditional lower bounds in Table 1 are tight, except for the curious case of the induced 4-Cycle C_4 . Non-induced C_4 can famously be detected in $O(n^2)$ time (see e.g. [31]). Meanwhile, the fastest algorithm for induced C_4 runs in $O(n^\omega)$ time (see e.g. [34]). There is no non-trivial lower bound known for C_4 detection (except that one needs to read the graph), and obtaining a higher lower bound or a faster algorithm for C_4 has been stated as an open problem several times (see e.g. [19]).

The induced 4-cycle is the smallest pattern H whose complexity is not tightly known, under any plausible hardness hypothesis.

We make partial progress under the popular 3-Uniform 4-Hyperclique Hypothesis (see e.g. [1], [27]) that postulates that hyperclique on 4 nodes in an n vertex 3-uniform hypergraph cannot be detected in $O(n^{4-\varepsilon})$ time for any $\varepsilon > 0$, in the word-RAM model of computation with $O(\log n)$ bit words. The believability of this hyperclique hypothesis is discussed at length in [27] (see also [1]); one reason to believe it is that refuting it would imply improved algorithms for many widely-studied problems such as Max-3-SAT [36].

Theorem 2.4: Under the 3-Uniform 4-Hyperclique Hypothesis, there is no $O(m^{4/3-\varepsilon})$ time or $O(n^{2-\varepsilon})$ time algorithm for $\varepsilon > 0$ that can detect an induced 4-cycle in an n -node, m -edge undirected graph.

While our result conditionally rules out, for instance, a linear time (in the number of edges) algorithm for induced

	Best known running time	Best known lower bound
I_4  K_4 	$n^{\omega(2,1,1)}$	$n^{\omega(2,1,1)}$
co-diamond  diamond 	n^ω	n^ω
co-paw  paw 	n^ω	n^ω
$2K_2$  C_4 	n^ω	$m + n$ $m^{4/3}$ or n^2 [new for C_4]
claw  co-claw 	n^ω	n^ω
P_4 	$m + n \leq n^2$	$m + n$

Fig. 1. A list of all induced 4-node patterns with their best known algorithms and conditional lower bounds. All runtime bounds are within polylogarithmic factors, and can be found in [34] and [17]. All lower bounds except for the new one in red are under the k -clique hypothesis for $k = 3$ or 4, and every lower bound l as a function of m or n should be interpreted as $l^{1-o(1)}$. The new lower bound is under the 3-uniform 4-hyperclique hypothesis, only holds for C_4 and is in Theorem 2.4. All upper and lower bounds are tight except that for C_4 and its complement.

C_4 , it does not rule out an $O(n^2)$ time algorithm for induced C_4 in dense graphs since the number of edges in the reduction instance is $\Theta(n^{3/2})$ in terms of the number of nodes n . Ideally, we would like to have a reduction from triangle detection to induced C_4 -detection, giving evidence that $n^{\omega-o(1)}$ time is needed. Our Theorem does show this if $\omega = 2$, but we would like the reduction to hold for any value of ω , and for it to be meaningful in dense graphs. Note that even if $\omega = 2$, a reduction from triangle detection would be meaningful, as it would say that a practical, combinatorial algorithm would be extremely difficult to obtain (or may not even exist).

All known reductions from k -clique to SI for other patterns H (e.g. [16], [20], [26]) work equally well for non-induced SI. In particular, in the special case when H is bipartite, such as when $H = C_4$, the host graph also ends up being bipartite (e.g. [26] for bicliques, and [16], [20] more generally).

Unfortunately such reductions are doomed to fail for C_4 . In bipartite graphs and more generally in triangle-free graphs, any non-induced C_4 is an induced C_4 . Of course, any hypothetical fine-grained reduction from triangle detection to non-induced C_4 detection in triangle-free graphs, combined with the known $O(n^2)$ time algorithm for non-induced C_4 would solve triangle detection too fast.

The difference between induced C_4 and non-induced C_4 is that the latter calls for detecting *one* of the three patterns: C_4 , diamond or K_4 . Could we have a reduction from triangle detection to induced C_4 -detection in a graph that is *not* triangle-free, but is maybe K_4 -free? In order for such a reduction to work, it must be that detecting one of $\{C_4, K_4\}$ is computationally hard.

We show that such reductions are also doomed. We provide a fast combinatorial algorithm that detects one of $\{C_4, H\}$ for any H that contains a triangle. The algorithm in fact runs faster than the current matrix multiplication time, which (under the k -clique Hypothesis) is required for detecting any H containing a triangle. Thus, any tight reduction from triangle detection to induced C_4 must create instances that contain *every* induced 4-node H that has a triangle.

Theorem 2.5: For any 4-node graph H that contains a triangle, detecting one of $\{C_4, H\}$ as an induced subgraph of a given n -node host graph can be done in $O(n^{7/3})$ time. If H is not a diamond or K_4 , then H or C_4 can be detected $O(n^2)$ time.

The only case of Theorem 2.5 that was known is that for $\{C_4, \text{diamond}\}$. Eschen et al. [19] considered the recognition of $\{C_4, \text{diamond}\}$ -free graphs and gave a combinatorial $O(n^{7/3})$ time algorithm for the problem. We show a similar result for every H that contains a triangle.

The C_4 OR H problem solved by our theorem above is a special case of the subgraph isomorphism problem in which we are allowed to return one of a *set of possible patterns*. This version of SI is a natural generalization of non-induced subgraph isomorphism in which the set of patterns are all supergraphs of a pattern. This generalized version of SI has practical applications as well. Often computational problems needed to be solved in practice are not that well-defined, so that for instance you might be looking for something like a matching or a clique, but maybe you are okay with extra edges or some edges missing. In graph theory applications related to graph coloring, one is often concerned with $\{H, F\}$ -free graphs for various patterns H and F (e.g. [13], [14], [23]). Recognizing such graphs is thus of interest there as well. We call the problem of detecting one of *two* given induced patterns, “Paired Pattern Detection”.

Intuitively, if a set of patterns all contain a k -clique, then returning at least one of them should be at least as hard as k -clique. While this is intuitively true, proving it is not obvious

at all. In fact, until recently [16], it wasn’t even known that if a single pattern H contains a k -clique, then detecting an induced H is at least as hard as k -clique detection. We are able to reduce k -clique in a fine-grained way to “Subset Pattern Detection” for any subset of patterns that all contain the k -clique as a subgraph⁴.

Theorem 2.6: Let S be a set of patterns such that every $H \in S$ contains a k -clique. Then detecting whether a given graph contains some pattern in S is at least as hard as k -clique detection.

While having a clique in common makes a subset of patterns hard to detect, intuitively, if several patterns are very different from each other, then detecting one of them should be easier than detecting each individually. We make this formal for Paired Pattern Detection in n node graphs for $k \leq 4$ as follows:

- Paired Pattern Detection is in $O(n^2)$ time for *every pair* of 3 node patterns. Moreover, for all but two pairs of patterns, it is actually in linear time.
- Paired Pattern Detection for any pair of 4-node patterns is in $O(\tilde{n}^\omega)$ time, whereas the fastest known algorithm for 4-clique runs in supercubic, $O(n^{\omega(1,2,1)})$ time where $\omega(1,2,1) \leq 3.252$ [25] is the exponent of multiplying an $n \times n^2$ by an $n^2 \times n$ matrix.
- There is an $O(n^2)$ time algorithm that solves Paired Pattern Detection for $\{H, \bar{H}\}$ for any 4-node H , where $\bar{H}$ is the complement of H .

The last bullet is a generalization of an old Ramsey theoretic result of Erdős and Szekeres [18] made algorithmic by Boppana and Halldórsson [7]. The latter shows that in linear time for any n -node graph, one can find either a $\log(n)$ size independent set or a $\log(n)$ size clique. Thus, for every constant k and large enough n , there is a linear time algorithm that either returns a k -clique or an I_k .

We note that our generalization for $\{H, \bar{H}\}$ cannot be true in general for $k \geq 5$: both H and its complement⁵ can contain a clique of size $dk/2e \geq 3$, and thus by our Theorem 2.6, their Paired Pattern Detection is at least as hard as $dk/2e$ -clique, and thus is highly unlikely to have an $O(n^2)$ -time algorithm.

A. Related work

There is much related work on the complexity of graph pattern detection in terms of the *treewidth* of the pattern. Due to the Color-Coding method of Alon, Yuster and Zwick [3], it is known that if a pattern H has treewidth t , then detecting H as a *non-induced* pattern can be done in $O(n^{t+1})$ time. This implies for instance that non-induced k -paths and k -cycles can be found in $2^{O(k)} \text{poly}(n)$ time.

Marx [29] showed that there is an infinite family of graphs of unbounded treewidth so that under ETH, (non-induced) SI on these graphs requires $n^{\Omega(t/\log t)}$ time where t is the treewidth of the graph. Recently, Bringmann and Slusallek [8] showed that under the Strong ETH, for every $\varepsilon > 0$, there is

⁴Our reduction works in the weaker non-induced version and so it works for the induced version as well.

⁵For $k = 5$, consider H to be a triangle and two independent nodes. Both H and its complement contain a triangle.

a t and a pattern H of treewidth t so that detecting H cannot be done in $O(n^{t+1-\varepsilon})$ time. That is, for some non-induced patterns, n^{t+1} is essentially optimal.

In the induced case, many patterns are also easier than k -clique, e.g. for $k = 3, 4, 5, 6$, any H that is not the k -independent set or the k -clique can be found in the current best running time $C(n, k-1)$ for $(k-1)$ -clique [5], [16], [34]. For $k \geq 7$, Bläser et al. [5] showed the weaker result that all k -node H that are not the clique or independent set can be detected in $O(n^{k-1})$ time combinatorially, whereas the best known combinatorial algorithms for k -clique run in $n^{k-o(1)}$ time.

For induced pattern detection for patterns of size $k \geq 8$, the best algorithm for almost all of the patterns has the same running time as k -clique detection. If we only resort to combinatorial algorithms there is a slight improvement: any pattern that is not a clique or independent set can be detected in $O(n^{k-1})$ time [5].

Manurangsi, Rubinfeld and Schramm [28] formulated a brand new hypothesis on the hardness of planted clique. This new hypothesis implies many results that are not known to hold under standard hypotheses such as ETH or Strong ETH, including that for every k -node H , its induced pattern detection problem requires $n^{\Omega(k)}$ time. While identifying new plausible hypotheses is sometimes worthwhile, our work strives to get results under standard widely-believed hypotheses, and to uncover combinatorial relationships between H -pattern detection and clique-detection, as cliques are the hardest patterns to detect.

Note that the results of Marx [29] and Bringmann and Slusallek [9] show hardness for specific classes of patterns, whereas the the results of Dalirrooyfard et al. [16], Manurangsi et al. [28] and this paper aim to determine hardness for *any* k -node pattern. Our paper primarily focuses on giving lower bounds for fixed patterns such as C_4, C_7 etc., whereas the focus of [28] is more asymptotic.

B. Organization of the paper

In Section III we give a high level overview of our techniques, and a comparison to the past techniques. In Section II-B we give the necessary definitions. In Section IV we first state our hardness result for PSI (Theorem 2.1) in subsection IV-A, and then in subsection IV-B we state our hardness result for SI (Theorem 2.2). Finally, in subsection IV-C we show hardness for paths and cycles (Theorem 2.3). We state our results on Paired Pattern Detection and our lower bound for induced four cycle detection from 3-Uniform 4-Hyperclique Hypothesis in the full version of the paper [15].

For an integer k , let P_k, C_k, K_k and I_k be the path, cycle, clique and independent set on k nodes.

Let G be a graph and $S \subseteq V(G)$ be a subgraph of it. For every node $v \in V(G)$, define $N_S(v)$ to be the neighbors of v in S . Define $d_S(v) = |N_S(v)|$.

A k -partite graph G can be decomposed into k partitions $G_1, \dots, G_k$ where each G_i is an independent set. For a pattern H of size k with vertices $v_1, \dots, v_k$, we say that a graph G

is H -partite if it is a k -partite graph with $G_1, \dots, G_k$ as its partitions such that there is no edge between G_i and G_j if $v_i v_j$ is not an edge in H .

Let G be an H -partite subgraph for a pattern H . We say that subgraph H^0 of G is a *colorful* copy of H if H^0 has exactly one node in each partition of G . Note that if the vertices of H^0 are $u_1, \dots, u_k$ where u_i is a copy of v_i for all i , then u_i must be in G_i for all i ⁶. This is because for every i, j where $v_i v_j$ is an edge, there must be an edge between the vertex of H^0 that is in G_i and the vertex of H^0 that is in G_j . Otherwise, the number of edges of H^0 is going to be smaller than the number of edges of H .

For a set of patterns S , by (induced) S -detection we mean finding a (induced) copy of one of the patterns in S , or indicating that there is no copy of any of the patterns in S .

Let $f : \{1, \dots, c\} \rightarrow V(H)$ be a *proper* coloring of the graph H if the color of any two adjacent nodes is different. Let the *chromatic number* of a graph H be the smallest number c such that there exists a proper coloring of H with c colors. We say that a graph H is *color critical* if the chromatic number of H decreases if we remove any of its nodes.

We call the subgraph C of a graph H a *core* of H if there is a homomorphism $H \rightarrow C$ but there is no homomorphism $H \rightarrow C^0$ for any proper subgraph C^0 of C . Recall that a graph which is its own core is called simply a *core*. Moreover, any graph has a unique core up to isomorphisms, and the core of a graph is an *induced* subgraph of it [21].

III. TECHNICAL OVERVIEW

Here we give high level overview of our techniques. To understand our lower bounds for k -node patterns, we should first give an overview of the techniques used in [16]. In their first result [16] shows that if H is t -chromatic and has a t -clique, then it is at least as hard to detect as a t -clique.

a) *Reduction (1) [16].* To prove the result of [16], suppose that we want to reduce detecting a t -clique in a host graph $G = (V, E)$ to detecting H in a graph $G^\square$ built from G and H . We build $G^\square$ by making a copy $G_h^\square$ of the vertices of G for each node $h \in H$ as an independent set. Then if $hh^0 \in E(H)$, we put edges between $G_h^\square$ and $G_{h^0}^\square$ using E : if $uw \in E$, then we connect the copy of u in $G_h^\square$ to the copy of w in $G_{h^0}^\square$. Note that we have edges between $G_h^\square$ and $G_{h^0}^\square$ if and only if hh^0 is an edge and this enforces an encoding of H in $G^\square$ (we refer to $G^\square$ as being H -partite).

To show that this reduction works, first suppose that there is a t -clique $\{v_1, \dots, v_t\}$ in G . To prove that there is a H in $G^\square$, we consider a t coloring of the vertices of H , and then we pick a copy of v_i from $G_h^\square$ if h has color i . Using the structure of $G^\square$ and the fact that no two adjacent nodes in H have the same color, one can show these $|H|$ nodes form a copy of H . For the other direction, suppose that there is a copy of H inside $G^\square$. This copy contains a t -clique $\{w_1, \dots, w_t\}$. Since each $G_h^\square$ is an independent set, no two nodes of the t -clique are in the same $G_h^\square$. Moreover, the edges in $G^\square$ mimic

⁶Note that this statement and many more in the paper are true up to automorphisms

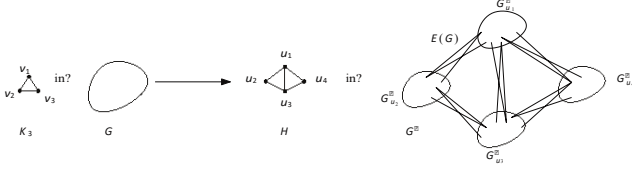


Fig. 2. Reduction (1) construction: Reducing K_3 -detection in G to H -detection in G^D for H being the Diamond. The pattern H is 3 colorable and has a 3-clique.

the edges in G and this is sufficient to conclude that no two nodes of the t -clique are copies of the same node in G , and the original nodes in G that the nodes w_i are the copies of, form a t -clique. See Figure 2.

Now we show how we modify this reduction to prove our first result, Theorem 2.1.

b) Reduction (2): We prove that if the size of the largest clique minor of a pattern H is $\mu(H)$, then detecting a $\mu(H)$ -clique in a graph G can be reduced to detecting a colorful copy of H in a graph G^D that is constructed from G and H (Theorem 2.1). Reduction (1) above is good at catching cliques that are in the pattern H , but H might not have a clique of size $\mu(H)$ in it, so we need a way to encode the clique minor of H in G^D so that it translates to a clique in G . To do that, we use a second method to put edges between G^D and G^D when hh^0 is an edge. We consider a clique minor of H of size $\mu(H)$. Note that the clique minor partitions the vertices of H into connected subgraphs with at least one edge between every two partition. Now if hh^0 is an edge in H and h and h^0 are in the same partition in the clique minor, we want to treat them as one node. So we put a “matching” between G^D and G^D : for any node $v \in V(G)$, we put an edge between the copy of v in G^D and the copy of v in G^D . This way we show that whenever there is a colorful copy of H in G^D , if h and h^0 are in the same partition of the clique minor of H , the vertices that are selected from G^D and G^D must be copies of the same node in G . This means that each clique minor partition of H represents one node in G . For h and h^0 that are not in the same clique minor partition, we put edges between G^D and G^D the same as Reduction (1) (mimicking E). Using the rest of the properties of the construction, we show that the set of nodes that each clique minor partition represents are all distinct, and they form a $\mu(H)$ -clique in G .

Note that [16] uses the idea in Reduction 2 (a second method to define the edges of G^D) in a separate result. However the use of clique minors in [16] is indirect; it is coupled with the chromatic number and proper colorings of H , and in our results we directly use clique minors without using any other properties, thus avoiding the Hadwiger conjecture.

Another thing to note about Reduction (2) is that we are reducing a clique detection problem to a “partitioned” subgraph isomorphism (PSI) problem. The reduction immediately fails if one removes the partitioned constraint. The reason is that we no longer can assume that if the reduction graph G^D has a copy of H , then the nodes are in different vertex subsets

G^D_h . If G^D has a copy of H and two nodes v, u of this copy are in one vertex subset G^D_h , then we don’t know if v and u are adjacent in G or not. This can get in the way of finding a clique of the needed size in G . So if we want to get any result stronger than Reduction (1) for SI (and not PSI), we need to add new ideas. We introduce some of these new ideas below.

c) Reduction (3): paths and cycles: In Theorem 2.3 we show that if H is the complement of a cycle or a path, then we can reduce detecting a $(\mu(H) - 2)$ -clique in a graph G to detecting a copy of H in a graph G^D constructed from G and H .

As mentioned above, removing the partitioned constraint from reduction (2) doesn’t directly work. However, when the graph is a core, it does work, and that is because PSI and SI are equivalent for cores [29]. When H is a core, there is only one homomorphism from H to itself, which means that there is only one type of “embedding” of H in the reduction graph G^D , and it is the embedding with exactly one vertex in each vertex subset G^D of G^D . However, when H is not a core, there can be multiple embeddings of H in G^D , and these embeddings do not necessarily result in finding a copy of a t -clique in H , for $t \approx \mu(H)$.

In order to solve this issue of multiple embeddings, we “shrink” some of the vertex subsets (G^D_h s) of the reduction graph G^D . More formally, we replace some of these subsets in G^D by a single vertex. We do it in such a way that the only embedding of H in G^D is the one with exactly one vertex in each subset. This way, the rest of the argument of Reduction (2) goes through. There is a cost to shrinking these subsets: shrinking more subsets results in reducing the size of the clique that we reduce from. So the harder part of this idea is to carefully decide which partitions to shrink, so that we only lose a small constant in the size of the clique detection problem that we are reducing from.

Recall that in Reduction (2) we consider a $\mu(H)$ clique minor of H which partitions the vertex set of H into $\mu(H)$ connected subgraphs. Here we observe that for H that is the complement of a path or a cycle, we can select two particular partitions of the clique minor, and shrink vertex subsets G^D for vertices h that belong to one of these two partitions. This way we eliminate all the unwanted embeddings of the pattern H in G^D , and reduce $(\mu(H) - 2)$ -clique detection in G to H detection in G^D . We note that the techniques in Reduction (3) are of independent interest and can be potentially used for other graph classes.

We now move on to our next reduction.

d) Reduction (4): Our next main result is Theorem 2.2, which states that if C is the core of the pattern H , then detecting C in a graph $G = (V, E)$ can be reduced to detecting H in a graph G^D which is constructed from G and H .

First note that Reduction (1) doesn’t directly work here. This is because if G has a copy of C , we have no immediate way of finding a copy of H in G^D . Recall that in Reduction (1) we used a coloring property of H to do this.

As a first attempt to such a reduction, one might use the following idea of Floderus et al. [20]. They showed that any

pattern that has a t -clique that is disjoint from all the other t -cliques in the pattern is at least as hard as t -clique to detect. Here we explain their idea in the context of reducing $\text{core}(H)$ -detection to H -detection. Let C^0 be a copy of C in H . The idea is to build the reduction graph of Reduction (1) using C^0 as the pattern, and to add the rest of the pattern H to it. More formally, for any node h in C^0 , let $G^{\square}$ be a copy of V , the set of vertices of G . Put edges between $G^{\square}$ and $G^{\square}_o$ same as before if hh^0 is an edge. Call this graph $G^{\square}_o$. To complete the construction of $G^{\square}$, add a copy of the subgraph $H \setminus C^0$ to $G^{\square}_o$, and connect a vertex h^0 in this copy to all the nodes in $G^{\square}$ for $h \in C^0$ if hh^0 is an edge in H .

The reason we construct the Reduction (1) graph on C^0 is that if G has a copy of C , then we can find a copy of C in $G^{\square}_o$ using the arguments in Reduction (1). This copy of C and all the vertices in $G^{\square} \setminus G^{\square}_o = H \setminus C^0$ form a copy of H . For the other direction, suppose that there is a copy $H^{\square}$ of H in $G^{\square}$. We hope that this copy contains a subgraph C^{00} that is completely inside $G^{\square}_o$, so that then this leads us to a copy of C in G using the properties of $G^{\square}_o$ and the fact that C is a core. However, such a construction cannot guarantee this, and in fact there might be no copies of C in $H^{\square}$ that are completely in $G^{\square}_o$.

So we need to find a subgraph H^0 of H , so that if we build the reduction graph $G^{\square}_{H^0}$ of Reduction (1) on it, it has the property that if G has a copy of C , then we can find a copy of H^0 in $G^{\square}_{H^0}$.

To do this, we simplify and use an idea of [16]. In particular, [16] introduces the notion of (K_t, F) -minor colorability of a pattern B , which is a coloring of B with t colors such that the coloring imposes a t -clique minor on any copy of F in B . Then using this definition, one finds a minimal covering of the graph H with (K_t, F) -minor colorable subsets and one argues that one can take one of these subsets as H^0 .

We notice that the properties that [16] uses relating the chromatic number and the clique minor of a pattern in this construction can be summarized into the core of patterns. We introduce the notion of F -coloring, which simply says that if B is F -colorable then there is a coloring such that any copy of F in B is a colorful copy under this coloring. Then we cover H with minimal number of C -colorable subsets. We show that we can take one of these subsets as H^0 .

Finally, we generalize Theorem 2.2 to the problem of detecting a pattern from a set S of patterns in Theorem ???. We show that if S is a set of patterns, there is a pattern $H \in S$, such that detecting the core of H , C , in a graph G can be reduced to detecting any pattern from S in a graph $G^{\square}$ constructed from G and S . In fact, $G^{\square}$ is the reduction graph of Reduction (4) on H as the pattern. The main part of Theorem ??? is to find the appropriate H in S . In order to find this pattern H , we look at homomorphisms between the patterns in S . In particular, we form a graph with nodes representing patterns in S and directed edges representing homomorphisms. We look at a strongly connected component of this graph that has no edges from other components to it, so there is no homomorphism from any pattern outside this component

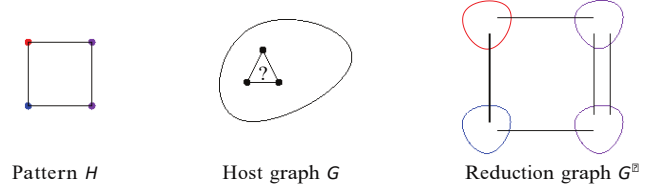


Fig. 3. Example of the Reduction of Theorem 2.1 for pattern C_4 with a K_3 -minor. From left to right: The pattern H with a K_3 minor function shown as a coloring, the host graph G in which we want to find a triangle and the reduction graph $G^{\square}$ built from G and H . The bold edges represent the edges in G , whereas the double edge represents a perfect matching. Each of the four colored parts in $G^{\square}$ are a copy of G .

to any pattern inside the component. We show that all the patterns in this component have the same core and we show that the pattern H can be any of the patterns in this component.

IV. LOWER BOUNDS

A. Hardness of PSI

In this section we prove Theorem 2.1, which reduces a $\eta(H)$ -clique detection to H -detection, where $\eta(H)$ is the size of the largest clique minor of H .

We can represent a clique minor of H of size t by a function in the following definition.

Definition 4.1: Let $f : V(H) \rightarrow \{1, \dots, t\}$ be a function such that for any $i \in \{1, \dots, t\}$, the preimage of i , $f^{-1}(i)$, induces a connected subgraph of H and for every $i, j \in \{1, \dots, t\}$, there is at least one edge between the preimages $f^{-1}(i)$ and $f^{-1}(j)$. We call such f a K_t -minor function of H .

One can think of f as a coloring on vertices of H that imposes a clique minor on H . Figure 3 shows an example of a K_3 -minor function of C_4 as a coloring. In the reduction we are going to consider a K_t -minor function f for $t = \eta(H)$. We can find a maximum clique minor of H and its associated function in $O_k(1)^7$ as follows: Check for all functions $f : V(H) \rightarrow \{1, \dots, k\}$ if f is a K_t -minor function for some t , and then take the f that creates a maximum K_t -minor.

Theorem 2.1: (Hardness of PSI) Let H be a k -node pattern with maximum clique minor of size $\eta(H)$, and let G be an n -node graph. Then one can construct a k -partite $O(n)$ -node graph $G^{\square}$ in $O(n^2)$ time such that $G^{\square}$ has a colorful copy of H if and only if G has a clique of size $\eta(H)$.

Proof. Let the size of the maximum clique minor of H be t , i.e. $\eta(H) = t$ and let $f : V(H) \rightarrow \{1, \dots, t\}$ be a K_t -minor function of the pattern H . Using the function f and the graph G , we construct the reduction graph $G^{\square}$ as follows:

The vertex set of $G^{\square}$ consists of partitions $G^{\square}_v$ for each $v \in V(H)$, where the partition $G^{\square}_v$ is a copy of the vertices of G as an independent set for all $v \in V(H)$.

The edge set of $G^{\square}$ is defined as follows. For every two vertices v and u in the pattern H where vu is an edge and $f(v) = f(u)$, we add the following edges between $G^{\square}_v$ and

⁷Any function that has dependency on k and no other parameter is of $O_k(1)$

$G_u^\square$: for each w_1 and w_2 in G , add an edge between the copy of w_1 in G_v and the copy of w_2 in G_u if and only if w_1w_2 is an edge in G . In other words, we put the same edges as $E(G)$ between $G_u^\square$ and $G_v^\square$ in this case. For any two vertices v and u in H where vu is an edge and $f(v) = f(u)$, add the following edges between $G_v^\square$ and $G_u^\square$: for any $w \in G$, connect the two copies of w in $G_v^\square$ and $G_u^\square$. In other words, we put a complete matching between $G_v^\square$ and $G_u^\square$ in this case. This completes the definition of $G^\square$. See Figure 3 for an example.

Note that $G^\square$ is an H -partite graph with nk vertices and since for each pair of vertices $u, v \in H$ we have at most m edges between $G_u^\square$ and $G_v^\square$, the construction time is at most $O(k^2m + kn) \leq O(k^2n^2)$.

Now to prove the correctness of the reduction, first we show that the reduction graph $G^\square$ has a subgraph isomorphic to H if G has a t -clique. Suppose that the vertices $w_1, \dots, w_t \in V(G)$ form a t -clique. Let $H^\square$ be the subgraph induced on the following vertices in the reduction graph $G^\square$: For each $v \in H$, pick $w_{f(v)}$ from $G^\square$. We need to show that if $vu \in E(H)$, then there is an edge between the vertices picked from $G^\square$ and $G^\square$. This is because if $f(v) = f(u) = i$, then we picked w_i from both $G^\square$ and $G^\square$ and hence they are connected. If $f(v) = f(u)$, then since $w_{f(v)}$ is connected to $w_{f(u)}$ in G , we have that their copies in $G^\square$ and $G^\square$ are connected as well. So $H^\square$ is isomorphic to H .

Now we show that G has a t -clique if $G^\square$ has a colorful subgraph $H^\square$ isomorphic to H . Let $v^\square \in V(H^\square)$ be the vertex picked from $G_v^\square$, for $v \in V(H)$. Since there is no edge between $G_v^\square$ and $G_u^\square$ if uv is not an edge in H , we have that there must be an edge between $v^\square$ and $u^\square$ if uv is an edge in H , so that the number of edges of $H^\square$ matches that of H . So if $uv \in E(H)$ and $f(u) = f(v)$, then $u^\square$ and $v^\square$ must be the copies of the same vertex in G . Since the vertices with the same value of f are connected, the vertices of $H^\square$ are the copies of exactly t vertices in G , say $\{w_1, \dots, w_t\}$, where $v^\square$ is the copy of w_i if $f(v) = i$. For each $i, j \in \{1, \dots, t\}$, there are two vertices $u, v \in V(H)$ such that $f(u) = i, f(v) = j$ and $uv \in E(H)$. So $u^\square v^\square \in E(H^\square)$, and hence $w_i w_j \in E(G)$. So the set $\{w_1, \dots, w_t\}$ induces a t -clique in G .

Recall that Corollary 2.2 gives a hardness result for cores in SI. This Corollary comes from the result of Marx [29] that PSI and SI are equivalent when the pattern is a core.

Corollary 2.2: (Hardness of cores in SI) Let G be an n -node m -edge graph and let H be a k -node pattern with maximum clique minor of size $\eta(H)$. If H is a core, then one can construct a graph $G^\square$ with at most $O(n)$ vertices in $O(m + n)$ time such that $G^\square$ has a subgraph isomorphic to H if and only if G has a $\eta(H)$ -clique as a subgraph.

We are going to use this result later for proving tighter hardness results for paths and cycles. Now we prove Corollary 2.3 that gives a lower bound for induced SI when the pattern is a core.

Corollary 2.3: (Hardness for induced-SI for cores) Let H be a k -node pattern which is a core. Suppose that $w(H)$ is the size of the maximum clique in H . Then detecting H in an n -node graph as an induced subgraph is at least as hard as detect-

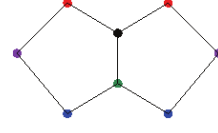


Fig. 4. C -coloring with C being the 5-cycle. The pattern has 2 copies of C_5 , and each copy is colorful with respect to the coloring given.

ing a clique of size $\max\{d^{\frac{p}{k+2w(H)}}/2e, d^{\frac{p}{k/1.95}}e\}$.

Proof. To get a lower bound for induced SI when the pattern is a core, we use two results on the connection of the maximum independent set $\alpha(H)$, maximum clique size $w(H)$ and the size of the maximum clique minor $\eta(H)$ of a pattern H . Kawarabayashi [24] showed $(2\alpha(H) - 1) \cdot \eta(H) \geq |V(H)| + w(H)$, and Balogh and Kostochka [4] showed that $\alpha(H)\eta(H) \geq |V(H)|/(2 - c)$ for a constant $c > 1/19.5$. Since $\eta(H) \geq w(H)$, these results imply that $\alpha(H)\eta(H) \geq \max\{|V(H)| + 2w(H))/2, |V(H)|/1.95\}$. Since $\eta(H) \geq w(H) = \alpha(H)$ and all these numbers are integers, we get Corollary 2.3 from Corollary 2.2.

B. Patterns are at least as hard to detect as their core

In this section we prove that detecting a pattern is at least as hard as detecting its core. In order to do so we define the notions of C -coloring and C -covering for a core subgraph C .

Definition 4.2: Let F be a graph and let C be a c -node subgraph of it. We say that the function $f : V(F) \rightarrow \{1, \dots, c\}$ is a C -coloring of F if for any copy of C in F , the vertices of this copy receive distinct colors. We say that a graph is C -colorable if it has a C -coloring.

Note that a C -coloring of F partitions F into c sections such that any copy of C in F is a colorful copy, i.e. it has exactly one vertex in each partitions⁸. See figure 4 for an example of C -coloring for C being the 5-cycle.

Definition 4.3: Let H be a graph and C be a core of H . We say that a collection $\mathcal{C} = \{C_1, \dots, C_r\}$, $C_i \subseteq V(H)$ is a C -covering for H of size r , if the following hold.

- 1) For every copy of C in H there is an i such that this copy is in the subgraph induced by C_i .
- 2) For every i the subgraph induced by C_i is C -colorable.

For any pattern H with core C there is a simple C -covering: Let the sets in the collection be the copies of C in H . However, we are interested in the “smallest” C -covering.

Definition 4.4: Define the C -covering number of H as the minimum integer r such that there is a C -covering for H of size r .

One can find a C -covering of minimum size in $O_k(1)$ by first enumerating all copies of C in H , and then considering all ways of partitioning the copies into sets, and testing if these sets are C -colorable. Before proving Theorem 2.2, we prove the following simple but useful lemma.

⁸Note that this is different than F being a C -partite graph. The colors are not assigned to any node of C , and there is no constraints on the edges of F with respect to the partitions.

Lemma 4.1: Let G be an H -partite graph where for each $v \in V(H)$, $G_v \subseteq G$ is the partition of G associated to v . Let F be a subgraph in G . Then there is a homomorphism g from F to H , defined as $g : V(F) \rightarrow V(H)$ where $g(u) = v$ if $u \in G_v$, for every $u \in V(F)$.

Proof. To prove that g is a homomorphism, we need to show that if $v_1 v_2 \in E(F)$, then $g(v_1)g(v_2) \in E(H)$. This is true because the edge $v_1 v_2$ is between $G_{g(v_1)}$ and $G_{g(v_2)}$, and from the definition of H -partite graphs this means that $g(v_1)g(v_2) \in E(H)$.

Theorem 2.2: Let G be an n -node m -edge graph and let H be a k -node pattern. Let C be the core of H . Then one can construct a graph $G^\square$ with at most $O(n)$ vertices in $O(n^2)$ time such that $G^\square$ has a subgraph isomorphic to H if and only if G has a subgraph isomorphic to C , with high probability⁹.

Proof. We use the color-coding trick of Alon, Yuster and Zwick [3]: Consider a random assignment of colors $\{1, \dots, c\}$ to the vertices of the host graph G , and a random assignment of numbers $\{1, \dots, c\}$ to the vertices of C . We can assume that if G has a copy of C , then the copy of vertex i has color i with high probability (we can repeat this reduction to produce $O(\log n)$ instances to achieve this high probability). Let the partition $G^{(i)}$ be the vertices with color i .

Let the C -covering number of the pattern H be r , and let $C = \{C_1, \dots, C_r\}$ be a C -covering of size r . Note that as explained before, we can find r and C in $O_k(1)$ time. Let $f : C_1 \rightarrow \{1, \dots, c\}$ be a C -coloring of C_1 , where $c = |V(C)|$ is the size of the core C .

We define the vertex set of the H -partite reduction graph $G^\square$ by adding a subset of vertices of G for each vertex $v \in C_1$ as the partition associated to v , and then simply adding a copy of the rest of the vertices of H to $G^\square$. More formally, for each vertex $v \in C_1$, let $G^\square_v$ be a copy of the partition $G^{(f(v))}$ as an independent set. For each vertex $v \in V(H) \setminus C_1$, let $G^\square_v = \{v^\square\}$ include a copy of v in $G^\square$. This finishes the vertex set definition.

We define the edge set of the reduction graph $G^\square$ as follows: For each pair of vertices $u, v \in C_1$, if uv is an edge and $f(u) = f(v)$, then we add a perfect matching between $G^\square_u$ and $G^\square_v$ as follows: For each $w \in G$, we add an edge between the copy of w in $G^\square_u$ and the copy of w in $G^\square_v$. If uv is an edge and $f(u) \neq f(v)$, then we add all the edges in $G^{(f(u))} \times G^{(f(v))}$ to $G^\square_u \times G^\square_v$ as follows: for each w_1 and w_2 in G , we add an edge between the copy of w_1 in $G^\square_u$ and the copy of w_2 in $G^\square_v$ if and only if $w_1 w_2$ is an edge in G . For each pair of vertices $u \in C_1$ and $v \in V(H) \setminus C_1$ such that uv is an edge in H , we add an edge between $v^\square \in G^\square_v$ and all vertices in $G^\square_u$. For each pair of vertices $u, v \in V(H) \setminus C_1$ such that uv is an edge in H , we add an edge between $u^\square \in G^\square_u$ and $v^\square \in G^\square_v$.

Note that the number of edges of $G^\square$ is at most $O(mk^2)$ where m is the number of edges of G . This is because for every $v \in V(H) \setminus C_1$, the number of edges attached to $G^\square_v = \{v^\square\}$ is at most $O(nk)$, and for every $u, v \in C_1$, there are at most

⁹with probability $1/\text{poly } n$

m edges between $G^\square_u$ and $G^\square_v$. So the construction time is $O(mk^2) \leq O(n^2)$.

Before proceeding to the proof of the reduction, note that if $uv \notin E(H)$, there is no edge between $G^\square_u$ and $G^\square_v$. So we have the following observation.

Observation 4.1: $G^\square$ is H -partite.

Now we prove that the reduction works. First suppose that G has a colorful copy $C^\square = \{v_1, \dots, v_c\}$ of C , such that v_i has color i . We are going to pick k vertices in the reduction graph $G^\square$, one from each partition, and prove that they induce a copy of H in $G^\square$. For every $v \in C_1$, we pick the copy of v_i in the partition $G^\square_v$, where i is the color of v in the C -coloring f of C_1 , i.e. $f(v) = i$. For $u \in V(H) \setminus C_1$ we pick the only vertex in $G^\square_u = \{u^\square\}$.

To prove that these k nodes induce a copy of H , consider $u, w \in V(H)$ where $uw \in E(H)$. We show that the vertices picked from $G^\square_u$ and $G^\square_w$ are connected. If one of u and w is not in C_1 , then all nodes in $G^\square$ is connected to all nodes in $G^\square$. If both u, w are in C_1 , we have two cases. If u and w have the same color, i.e. $f(u) = f(w) = i$, then we have picked copies of v_i from both $G^\square_u$ and $G^\square_w$, and from the definition of $G^\square$ they are connected. If u and w don't have the same color, i.e. $f(u) \neq f(w)$, then we have picked $v_{f(u)}$ from $G^\square_u$ and $v_{f(w)}$ from $G^\square_w$. Since $v_{f(u)}$ and $v_{f(w)}$ are connected in G , from the definition of $G^\square$ they are also connected in $G^\square$. So the vertices we picked from $G^\square$ induce a copy of H .

Now we are going to show that if there is a copy of H in the reduction graph $G^\square$, then there is a copy of C in G . For $i \in \{1, \dots, r\}$, let $S_i = \bigcup_{v \in C_i} G^\square_v$. Suppose that $G^\square$ has a subgraph $H^\square$ isomorphic to H . To show that G has a copy of C , we prove that $H^\square$ has a copy of C with all its vertices in S_1 , and then we show that this subgraph leads us to a copy of C in G .

First, consider a copy $C^\square$ of C in $H^\square$. By observation 4.1, we can consider the homomorphism that Lemma 4.1 defines from $C^\square$ to H : $u \in V(C^\square) \rightarrow v \in H$ if $u \in G^\square_v$. Since C is the core, the image of $C^\square$ defined by the homomorphism must be isomorphic to C . So this copy of C in $H^\square$ is mapped to a copy of C in H .

Thus each copy of C in $H^\square$ maps to a copy of C in H . Note that this copy is in C_i if and only if the copy of C in $H^\square$ is in S_i . Now suppose that there is no copy of C in $H^\square \cap S_1$. Then each copy of C in $H^\square$ is mapped to a copy of C in H that is not in C_1 , and thus it is in C_i for $i \geq 2$. So the copies of C in $H^\square$ are covered by $S = \{S_2 \cap H^\square, \dots, S_r \cap H^\square\}$. If we show that for all i , $S_i \cap H^\square$ is C -colorable, then S is a C -covering of size $r - 1$ for $H^\square$ and since $H^\square$ is a copy of H , this is a contradiction to the C -covering number of H .

To see that $S_i \cap H^\square$ is C -colorable, let $f_i : C_i \rightarrow \{1, \dots, c\}$ be the C -coloring of C_i , for $i = 2, \dots, r$. We color each node $v \in H^\square$ as follows. There is $u \in V(H)$ such that $v \in G^\square_u$. We color v the same as u , with $f_i(u)$. Now we show that each copy $C^\square$ of C in $S_i \cap H^\square$ has distinct colors. Consider the mapping of Lemma 4.1 from $C^\square$ in the H -partite graph $G^\square$ to H : for $v \in V(C^\square)$, we let $g(v) = u$ if $v \in G^\square_u$. Note that if $C^\square \subseteq H^\square \cap S_i$, the map g preserves colors. Since the image

of $C^\square$ in H is also a copy of C (because C is a core) and f_i is a C -coloring, this image is a colorful copy of C . So $C^\square$ is also a colorful copy of C with the coloring defined. Thus $S_i \cap H^\square$ is C -colorable.

So from above we conclude that $H^\square$ must have a copy $C^\square = \{w_1, \dots, w_c\}$ of C in S_1 , such that $w_i \in G_{v_i}$ for some $v_i \in C_1$ and $v_i = v_j$ for each $i = j$. Moreover, the mapping $w_i \rightarrow v_i$ is a homomorphism from $C^\square$ to H and since C is a core, we have that $v_1, \dots, v_c$ form a copy of C in H . Now since f is a C -coloring, $f(v_i) = f(v_j)$ for all $i = j$. This means that $w_1, \dots, w_c$ are copies of distinct vertices in G , and hence they are attached in $G^\square$ if and only if they are attached in G . So they form a subgraph isomorphic to C in G .

Now we prove Corollary 4.1 and 2.4 on induced subgraph isomorphism of *all* patterns.

Corollary 2.4: (Hardness of Induced-SI) For any k -node pattern H , detecting an induced copy of H in an n -node graph is at least as hard as detecting a clique of size $dk^{1/4}/1.39e$ in an $O(n)$ graph.

Proof. Denote the chromatic number of a graph F by $X(F)$. We know that for a k node pattern H , the chromatic number of either H or its complement is at least $\sqrt{k}$. WLOG assume that $X(H) \geq \sqrt{k}$. Lemma 4.2 proven below states that a color critical graph is a core. Since the core of H is its largest subgraph that is a core, we have that $X(\text{core}(H)) \geq \sqrt{k}$, and so in particular the size of the core of H is at least $d\sqrt{k}$. By Theorem 2.2 we have that detecting H is at least as hard as detecting $\text{core}(H)$, and by Corollary 2.3 we have that

detecting $\text{core}(H)$ is at least $d\sqrt{k}/1.95e$ hard. This gives the result that we want.

Corollary 4.1: (Hardness of Induced-SI) For any k -node pattern H , the problem of detecting an induced copy of H in an n -node graph requires $n^{\Omega(\sqrt{k}/\log k)}$ time under ETH.

Proof. Similar to the proof of Corollary 2.4, we have that $X(\text{core}(H)) \geq \sqrt{k}$. Now since by inductive coloring we have that for any graph F , $tw(F) + 1 \geq X(F)$, then $tw(\text{core}(H)) \geq \sqrt{k} - 1$. Recall that Marx [29] shows that under ETH, for any pattern F partitioned subgraph isomorphism of F in an n node graph requires $n^{\Omega(\frac{tw(F)}{\log tw(F)})}$ time. Since for cores PSI and SI are equivalent [29], we get Corollary 4.1.

Lemma 4.2: Color critical graphs are cores.

Proof. Let H be a color critical graph, and suppose that there is a homomorphism f from H to H^0 where H^0 is a proper subgraph of H . Let $c_{H^0} : V(H^0) \rightarrow \{1, 2, \dots, X(H^0)\}$ be a coloring of H^0 . Then let c_H be the following coloring for H . For each $v \in H^0$, color all vertices of $f^{-1}(v)$ the same as v . This means that for any $u \in V(H)$, $c_H(u) = c_{H^0}(f(u))$. Since $f^{-1}(v)$ is an independent set and c_{H^0} is a proper coloring, $c^{-1}(i) = f^{-1}(c^{-1}(i))$ is an independent set for any color i . So c_H is a proper coloring for H of size $X(H^0)$. This is a contradiction because H is color critical and we have that $X(H^0) < X(H)$.

C. Hardness of Paths and Cycles

In this section, we prove a stronger lower bound for induced path and cycle detection than what the previous results give

us. More precisely, we show that a cycle or path of length k is at least as hard to detect as an induced subgraph as a clique of size roughly $3k/4$. This number comes from the largest clique minor of the complement of paths and cycles. This is formalized in the next lemma which is proved in the appendix.

Lemma 4.3: Let H be a k -node pattern that is the complement of a path or a cycle. Then $\eta(H) = b^{\frac{k+\omega(H)}{2}}c$, where $\omega(H)$ is the size of the maximum clique of H . Table II shows the value of $\eta(H)$.

number of vertices (k)	$\eta(C_k)$	$\eta(P_k)$
$4t$	$3t^*$	$3t^*$
$4t + 1$	$3t$	$3t + 1^*$
$4t + 2$	$3t + 1$	$3t + 1$
$4t + 3$	$3t + 2$	$3t + 2$

TABLE II
MAXIMUM CLIQUE MINORS. (*) FOR $t = 1$, THE MAXIMUM MINOR OF P_4, C_4, P_5 IS 2,2,3 RESPECTIVELY.

Recall the main result of this section below.

Theorem 2.3: (Hardness of P_k and C_k) Let H be the complement of a P_k or the complement of a C_k . Suppose that t is the size of the maximum clique minor of H . Then the problem of detecting H in an $O(n)$ -node graph is at least as hard as finding a $(t-2)$ -clique in an n -node graph. If k is odd, then detecting an induced C_k is at least as hard as finding a t -clique.

First, we show the easier case of odd cycles which was also mentioned in Section IV-A. With a simple argument we can show that the complement of an odd cycle is a color critical graph. We prove this in the appendix for completeness.

Lemma 4.4: The complement of an odd cycle is color-critical.

Lemma 4.4 together with Lemma 4.2 show that the complement of an odd cycle is a core. Using Corollary 2.2 and Lemma 4.3, we have that detecting a C_k for odd k is at least as hard as detecting a $b3k/4c$ -clique. Since induced detection of a pattern H is at least as hard as not-necessarily-induced detection of H , we have the following Theorem.

Theorem 4.1: For odd k , Induced- C_k detection is at least as hard as $b3k/4c$ -clique detection.

Now we move to the harder case of even cycles and odd and even paths. We would like to get a hardness as strong as the one offered by Theorem 2.1 and Corollary 2.2, but we can't use these results directly since paths and even cycles (and their complements) are not cores.

As mentioned in the section III, we are going to use the construction of Theorem 2.1 and shrink a few partitions of the reduction graph $G^\square$, i.e. replacing each of these partitions with a single vertex. The next lemma helps us characterize automorphisms of paths and cycles, and so it helps us find the appropriate partitions of $G^\square$ to shrink.

Lemma 4.5: Any automorphism of paths or cycles that has a proper subset of vertices as its image has the following properties:

- Let $C_k = v_1 \dots v_k v_1$ be a k -cycle for even k . Then any homomorphism from $\bar{C}_k$ to a proper subgraph of $\bar{C}_k$ has two vertices both being mapped to either v_1 or v_k .
- Let $P_k = v_1 \dots v_k$ be a k -path. Then any homomorphism from P_k to a proper subgraph of P_k has two vertices both being mapped to either v_1 or v_k .

We include the proof Theorem 2.3 in the full version of this paper [15].

Using Lemma 4.3 and Theorem 2.3 we have the following Corollary.

Corollary 4.2: Detecting C_k or P_k as an induced subgraph is at least as hard as detecting a $(b3k/4c - 2)$ -clique.

REFERENCES

- [1] Amir Abboud, Karl Bringmann, Holger Dell, and Jesper Nederlof. More consequences of falsifying SETH and the orthogonal vectors conjecture. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 253–266. ACM, 2018.
- [2] Josh Alman and Virginia Vassilevska Williams. A refined laser method and faster matrix multiplication. In Daniel Marx, editor, *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021*, pages 522–539. SIAM, 2021.
- [3] Noga Alon, Raphael Yuster, and Uri Zwick. Color-coding. *J. ACM*, 42(4):844–856, 1995.
- [4] József Balogh and Alexandr V Kostochka. Large minors in graphs with given independence number. *Discrete mathematics*, 311(20):2203–2215, 2011.
- [5] Markus Bläser, Balagopal Komarath, and Kartek Sreenivasaiah. Graph pattern polynomials. In Sumit Ganguly and Paritosh K. Pandya, editors, *38th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2018, December 11-13, 2018, Ahmedabad, India*, volume 122 of *LIPICs*, pages 18:1–18:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.
- [6] Markus Bläser, Balagopal Komarath, and Kartek Sreenivasaiah. Graph pattern polynomials. *arXiv preprint arXiv:1809.08858*, 2018.
- [7] Ravi B. Boppana and Magnús M. Halldórsson. Approximating maximum independent sets by excluding subgraphs. *BIT*, 32(2):180–196, 1992.
- [8] Karl Bringmann and Jasper Slusallek. Current algorithms for detecting subgraphs of bounded treewidth are probably optimal. In *48th International Colloquium on Automata, Languages, and Programming, ICALP 2021, July 12-16, 2021, Glasgow, Scotland (Virtual Conference)*, volume 198 of *LIPICs*, pages 40:1–40:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.
- [9] Karl Bringmann and Jasper Slusallek. Current algorithms for detecting subgraphs of bounded treewidth are probably optimal. *arXiv preprint arXiv:2105.05062*, 2021.
- [10] Chris Calabro, Russell Impagliazzo, and Ramamohan Paturi. A duality between clause width and clause density for SAT. In *21st Annual IEEE Conference on Computational Complexity (CCC 2006), 16-20 July 2006, Prague, Czech Republic*, pages 252–260. IEEE Computer Society, 2006.
- [11] 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. In *19th Annual IEEE Conference on Computational Complexity (CCC 2004), 21-24 June 2004, Amherst, MA, USA*, pages 150–160. IEEE Computer Society, 2004.
- [12] Derek G. Corneil, Yehoshua Perl, and Lorna K Stewart. A linear recognition algorithm for cographs. *SIAM Journal on Computing*, 14(4):926–934, 1985.
- [13] Konrad K. Dabrowski, François Dross, and Daniël Paulusma. Colouring diamond-free graphs. *J. Comput. Syst. Sci.*, 89:410–431, 2017.
- [14] Konrad K. Dabrowski and Daniël Paulusma. On colouring $(2P_2, H)$ -free and (P_5, H) -free graphs. *Inf. Process. Lett.*, 134:35–41, 2018.
- [15] Mina Dalirrooyfard and Virginia Vassilevska Williams. Induced cycles and paths are harder than you think. *arXiv preprint arXiv:2209.01873*, 2022.
- [16] Mina Dalirrooyfard, Thuy Duong Vuong, and Virginia Vassilevska Williams. Graph pattern detection: hardness for all induced patterns and faster non-induced cycles. 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 1167–1178. ACM, 2019.
- [17] Friedrich Eisenbrand and Fabrizio Grandoni. On the complexity of fixed parameter clique and dominating set. *Theor. Comput. Sci.*, 326(1-3):57–67, 2004.
- [18] Paul Erdős and George Szekeres. A combinatorial problem in geometry. *Compositio Mathematica*, 2:463–470, 1935.
- [19] Elaine M. Eschen, Chinh T. Hoàng, Jeremy P. Spinrad, and R. Sriharan. On graphs without a c_4 or a diamond. *Discret. Appl. Math.*, 159(7):581–587, 2011.
- [20] Peter Floderus, Mirosław Kowaluk, Andrzej Lingas, and Eva-Marta Lundell. Induced subgraph isomorphism: Are some patterns substantially easier than others? *Theor. Comput. Sci.*, 605:119–128, 2015.
- [21] Pavol Hell and Jaroslav Nešetřil. The core of a graph. *Discrete Mathematics*, 109(1-3):117–126, 1992.
- [22] Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? In *39th Annual Symposium on Foundations of Computer Science, FOCS '98, November 8-11, 1998, Palo Alto, California, USA*, pages 653–663. IEEE Computer Society, 1998.
- [23] T. Karthick and Suchismita Mishra. On the chromatic number of $(p_6, \text{diamond})$ -free graphs. *Graphs Comb.*, 34(4):677–692, 2018.
- [24] Ken-ichi Kawarabayashi, Michael D Plummer, and Bjarne Toft. Improvements of the theorem of duchet and meyniel on hadwiger's conjecture. *Journal of Combinatorial Theory, Series B*, 95(1):152–167, 2005.
- [25] Francois Le Gall and Florent Urrutia. Improved rectangular matrix multiplication using powers of the coppersmith-winograd tensor. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 1029–1046. SIAM, 2018.
- [26] Bingkai Lin. The parameterized complexity of k -biclique. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 605–615. SIAM, 2015.
- [27] Andrea Lincoln, Virginia Vassilevska Williams, and R. Ryan Williams. Tight hardness for shortest cycles and paths in sparse graphs. In Artur Czumaj, editor, *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 1236–1252. SIAM, 2018.
- [28] Pasin Manurangsi, Aviad Rubinfeld, and Tsilil Schramm. The strongish planted clique hypothesis and its consequences. In James R. Lee, editor, *12th Innovations in Theoretical Computer Science Conference, IITCS 2021, January 6-8, 2021, Virtual Conference*, volume 185 of *LIPICs*, pages 10:1–10:21. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.
- [29] Daniel Marx. Can you beat treewidth? In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pages 169–179. IEEE, 2007.
- [30] Jaroslav Nešetřil and Svatopluk Poljak. On the complexity of the subgraph problem. *Commentationes Mathematicae Universitatis Carolinae*, 26(2):415–419, 1985.
- [31] Dana Richards and Arthur L Liestman. Finding cycles of a given length. In *North-Holland Mathematics Studies*, volume 115, pages 249–255. Elsevier, 1985.
- [32] Andrew Thomason. The extremal function for complete minors. *Journal of Combinatorial Theory, Series B*, 81(2):318–338, 2001.
- [33] Virginia Vassilevska Williams. On Some Fine-Grained Questions in Algorithms and Complexity. In *Proceedings of the International Congress of Mathematicians ICM 2018*, volume 3, pages 3447–3487, 2019.
- [34] Virginia Vassilevska Williams, Joshua R Wang, Ryan Williams, and Huacheng Yu. Finding four-node subgraphs in triangle time. In *Proceedings of the twenty-sixth annual ACM-SIAM symposium on discrete algorithms*, pages 1671–1680. SIAM, 2014.
- [35] Virginia Vassilevska Williams and R. Ryan Williams. Subcubic equivalences between path, matrix, and triangle problems. *J. ACM*, 65(5), August 2018.
- [36] Ryan Williams. A new algorithm for optimal 2-constraint satisfaction and its implications. *Theor. Comput. Sci.*, 348(2-3):357–365, 2005.