Fine-Grained Classification of Detecting Dominating Patterns

Jonathan Dransfeld
Karlsruhe Institute of Technology, Germany

Marvin Künnemann
Karlsruhe Institute of Technology, Germany

Mirza Redzic
Karlsruhe Institute of Technology, Germany

Abstract

We consider the following generalization of dominating sets: Let G be a host graph and P be a pattern graph P. A dominating P-pattern in G is a subset S of vertices in G that (1) forms a dominating set in G *and* (2) induces a subgraph isomorphic to P. The graph theory literature studies the properties of dominating P-patterns for various patterns P, including cliques, matchings, independent sets, cycles and paths. Previous work (Kunnemann, Redzic 2024) obtains algorithms and conditional lower bounds for detecting dominating P-patterns particularly for P being a k-clique, a k-independent set and a k-matching. Their results give conditionally tight lower bounds if k is sufficiently large (where the bound depends the matrix multiplication exponent ω). We ask: Can we obtain a classification of the fine-grained complexity for *all* patterns P?

Indeed, we define a graph parameter $\rho(P)$ such that if $\omega = 2$, then

$$\left(n^{\rho(P)} m^{\frac{|V(P)|-\rho(P)}{2}}\right)^{1\pm o(1)}$$

is the optimal running time assuming the Orthogonal Vectors Hypothesis, for all patterns P except the triangle K_3. Here, the host graph G has n vertices and $m = \Theta(n^\alpha)$ edges, where $1 \leq \alpha \leq 2$.

The parameter $\rho(P)$ is closely related (but sometimes different) to a parameter $\delta(P) = \max_{S \subseteq V(P)} |S| - |N(S)|$ studied in (Alon 1981) to tightly quantify the maximum number of occurrences of induced subgraphs isomorphic to P. Our results stand in contrast to the lack of a full fine-grained classification of detecting an arbitrary (not necessarily *dominating*) induced P-pattern.

2012 ACM Subject Classification Theory of computation → Graph algorithms analysis; Theory of computation → Parameterized complexity and exact algorithms; Theory of computation → Problems, reductions and completeness

Keywords and phrases fine-grained complexity theory, domination in graphs, subgraph isomorphism, classification theorem, parameterized algorithms

Digital Object Identifier 10.4230/LIPIcs.ESA.2025.98

Funding Research supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 462679611.

1 Introduction

Among the most intensively investigated graph problems is the dominating set problem: Given a graph $G = (V, E)$, find a (small) vertex subset $S \subseteq V$ that *dominates* all vertices, i.e., for each $v \in V$, we have $v \in S$ or there is some $s \in S$ with $\{s, v\} \in E$. In many scenarios, one might not merely want to find *any* dominating set, but rather a dominating set S satisfying some additional requirements. Possibilities include S forming a connected subgraph, admitting a perfect matching, or more generally being connected by a prescribed topology (e.g., rings or cliques) or even being fully disconnected (i.e., forming an independent

33rd Annual European Symposium on Algorithms (ESA 2025).
Editors: Anne Benoit, Haim Kaplan, Sebastian Wild, and Grzegorz Herman; Article No. 98; pp. 98:1–98:15
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

set). In this work, we study a general form of such problems: For any pattern graph H, the (Non-induced) Dominating P-Pattern problem asks to determine, given a graph G, whether there exists a (non-)induced copy of P in G that dominates all vertices of G.

Indeed, for various patterns P, the structural properties of Dominating P-Patterns have been well investigated, e.g., Dominating Cliques [5, 7, 12, 15, 26, 27, 28], Dominating Independent Sets[1] [8, 9, 10, 29, 34], Dominating (Induced) Matchings [22, 37, 38], Dominating Cycles [17, 18] or Dominating Paths [19, 20, 42]. Recently, the algorithmic complexity of detecting Dominating P-Patterns in general graphs has been performed by Künnemann and Redzic, focusing on cliques, independent sets and perfect matchings [30] (see below for further details). In this work, we set out to understand the fine-grained complexity of this problem for *all* patterns P.

Note that the Dominating P-Pattern problem is the natural combination of two classic problems which are heavily studied in isolation: Dominating Set and P-Pattern Detection.

- *k-Dominating Set*: Dominating set is a notoriously difficult problem. When parameterized by the solution size k (i.e., $|S|$), it is the arguably most natural W[2]-complete problem and hard even to approximate within $f(k)$ factors [6, 24]. The best known algorithm is based on fast matrix multiplication and solves it in time $n^{k+o(1)}$ for all sufficiently large k [16]. This is tight in the sense that an $\mathcal{O}(n^{k-\epsilon})$-time algorithm for any $k \geq 3$ and $\epsilon > 0$ would refute the k-Orthogonal Vectors Hypothesis (k-OVH) and thus the Strong Exponential Time Hypothesis [36] (see Section 2 for details). Taking the graph sparsity (i.e., the number m of edges) into account as well, a recent result [21] gives upper and conditional lower bounds establishing a tight time complexity of $mn^{k-2\pm o(1)}$ if the matrix multiplication exponent ω is equal to 2.
- *P-Pattern Detection (aka Induced P-Subgraph Isomorphism):* The complexity of P-Pattern Detection for general P is sensitive to the considered pattern P. The probably most notable special case is k-Clique Detection. It is the most natural $W[1]$-complete problem and also resists good approximations [6, 31, 23]. Its best known algorithm solves it in time essentially $\mathcal{O}(n^{\frac{\omega}{3}k})$ [33][2] and is conjectured to be optimal, see, e.g., [1]. For other (classes of) patterns P, however, only partial results are known: While for any k-node pattern P, the problem can be reduced to k-Clique Detection, this approach is not necessarily optimal. E.g., all 4-node patterns except clique and independent set can be detected polynomially faster than the conjectured time for clique and independent set, see in particular [25, 16, 39, 13]. Despite significant effort (see [25, 16, 39, 14, 13] for a selection), the task of finding matching upper and conditional lower bounds appears far from completed. Let us remark that also for Non-induced P-Pattern Detection a fine-grained classification of all patterns P remains open (see, e.g., [32, 13]).

We ask: *How does the time complexity of Dominating P-Pattern relate to the complexity of P-Pattern Detection versus to the complexity of k-Dominating Set?* Can we, despite the lack of a complete classification for P-Pattern Detection classify the fine-grained complexity of **all** *dominating* patterns?

Previous work appears to indicate that the complexity should be governed by the domination aspect: By adapting the conditional lower bound of Patrascu and Williams for k-Dominating Set [36] (see also [21, 30]) it is not too difficult to obtain a conditional lower

[1] We remark that a Dominating Independent Set is equivalent to the well-studied notion of a Maximal Independent Set.
[2] If k is divisible by 3.

bound of $n^{k-o(1)}$ in dense graphs for *any* pattern P, based on k-OVH. However, the situation becomes much more interesting for *sparse* graphs. Here, it has already been observed that the fine-grained time complexity is highly sensitive to the specific pattern P. Specifically, Künnemann and Redzic [30] give some curious insights into selected patterns: (Here, to simplify the presentation, we assume that $\omega = 2$ and that k-OVH holds.)

- If P is the k-star (a tree with a root and $k-1$ leaves), the tight time complexity is $mn^{k-2\pm o(1)}$, i.e., even in very sparse graphs with $m = \Theta(n)$, we only save a linear factor in n compared to the $n^{k\pm o(1)}$ complexity in dense graphs.
- In contrast, if P is the clique with $k \geq 5$ vertices or the independent set with $k \geq 3$ vertices, we obtain a particularly simple case with tight time complexity of $(m^{\frac{k+1}{2}}/n)^{1\pm o(1)}$. In very sparse graphs with $m = \mathcal{O}(n)$, the resulting running time of $n^{\frac{k-1}{2}\pm o(1)}$ is less than the square root of the running time of $n^{k\pm o(1)}$ in dense graphs.
- Finally, the substantially different pattern of a perfect matching on $k \geq 4$ vertices achieves the only slightly worse time complexity of $m^{\frac{k}{2}\pm o(1)}$.

We remark that the above results suggest that in sparse graphs, Dominating P-Pattern shares an additional flavor with the P-Pattern *Enumeration* problem: It is not too difficult to obtain an algorithm whose running time is roughly bounded by the time required to list all occurrences of the pattern P. However, it turns out that this number alone cannot fully explain the time complexity: While for the case of perfect matchings, the time complexity coincides precisely with the maximum number of occurrences of the pattern, for others (e.g., independent sets or cliques), the time complexity is polynomially less than the maximum number of occurrences of the pattern. This begs the question: which other parameter of the pattern P captures the time complexity of Dominating P-Pattern?

Our results. To state the time complexity for any pattern P, we introduce the following graph parameter $\rho(P)$. Here for any graph P and $S \subseteq V(P)$, we let $N(S)$ denote the (open) neighborhood of the vertices S, where $N(\emptyset) = \emptyset$.[3]

▶ **Definition 1.1.** *Let P be an arbitrary pattern graph, and denote by $I(P) \subseteq V(P)$ the set of isolated nodes of P. Choose an* independent *set $S \subseteq V(P) \setminus I(P)$ maximizing $|S| - |N(S)|$; if S is not uniquely defined, take any choice maximizing $|S|$. We define*

$$\rho(P) := \begin{cases} |S| - |N(S)| & \text{if } S \neq \emptyset \\ -1 & \text{if } S = \emptyset. \end{cases}$$

and set

$$t_P(n, m) := n^{\rho(P)} \cdot m^{\frac{|V(P)|-\rho(P)}{2}}.$$

These quantities turn out similar (but sometimes different) to the maximum number of occurrences of the pattern graph P for patterns *without isolated nodes* (i.e., $I(P) = \emptyset$). Specifically, if $I(P) = \emptyset$ and $S \neq \emptyset$, the maximum number of induced copies of P in an n-vertex m-edge graph G is $\Theta(t_P(n, m))$. Indeed, for connected graphs G and patterns P, Alon [3] defines the parameter $\delta(P) := \max_{S \subseteq V(P)} |S| - |N(S)|$ and establishes $\Theta(n^{\delta(P)} m^{\frac{|V(P)|-\delta(P)}{2}})$ as the maximum number of induced copies of P in G whenever G has $m = \Theta(n)$ edges

[3] For any graph G, we will always denote by $V(G)$ and $E(G)$ its set of vertices and edges, respectively.

(which generalizes to arbitrary $m \geq n$).[4] Note that $\rho(P)$ and $\delta(P)$ may only differ for some patterns P with $\delta(P) = 0$, in which case $\rho(P) \in \{0, -1\}$. In contrast, if P contains isolated nodes (e.g., the case of Dominating Independent Set), the number of occurrences of P and $t_P(n,m)$ may differ vastly (e.g., between $\Theta(n^k)$ and $\Theta(m^{k/2})$).

Our results determine that for all patterns (possibly except the triangle K_3), $t_P(n,m)^{1\pm o(1)}$ is the conditionally tight time complexity of detecting a Dominating P-Pattern if $\omega = 2$. Put differently, for any pattern P (except K_3), we can easily determine the conditionally optimal running time (assuming $\omega = 2$)! Specifically, we obtain the following algorithmic result:

▶ **Theorem 1.2** (Upper Bound). *For any pattern graph P with at least 16 vertices, there is an algorithm solving Dominating P-Pattern problem in time $t_P(n,m)^{1+o(1)}$. Further, if $\omega = 2$, this algorithm exists for all patterns P except K_3.*

We remark that our algorithms have a running time close to $t_P(n,m)^{1\pm o(1)}$ even under current values of ω and small pattern sizes – the small polynomial overhead depends on ω and vanishes if $\omega = 2$ (except for the triangle K_3). Further, we complement our algorithmic result by a corresponding conditional lower bound of $t_P(n,m)^{1-o(1)}$ based on the k-Orthogonal Vectors Hypothesis (and thus the Strong Exponential Time Hypothesis).

▶ **Theorem 1.3** (Conditional Lower Bound). *For any pattern graph P with at least 2 vertices, there is no algorithm solving the P-Dominating Set problem in time $\mathcal{O}(t_P(n,m)^{1-\varepsilon})$ for any $\varepsilon > 0$ unless the k-Orthogonal Vectors Hypothesis fails.*

The only pattern P for which we do not obtain an algorithm matching the lower bound of Theorem 1.3 is the triangle K_3. For this pattern, our best algorithm only leaves a time overhead of $m^{1/3+o(1)}$ if $\omega = 2$, yielding a bound of $(t_{K_3}(n,m) \cdot m^{\frac{1}{3}})^{1+o(1)} = (m^{\frac{7}{3}}/n)^{1+o(1)}$ in this case.

▶ **Theorem 1.4.** *We can solve Dominating Triangle in time $\left(m^{1+\frac{2\omega}{\omega+1}}/n\right)^{1+o(1)}$.*

Technical Outline and Overview. In Section 3, we obtain our algorithmic results: We first study sufficiently large connected patterns P (different from K_3 and K_4), which we call *basic patterns*. We argue that there are only $\mathcal{O}(t_P(n,m))$ sets that might form a Dominating P-Pattern (using similar arguments to Alon [3]). Moreover, we show how to enumerate all such candidate sets in time $t_P(n,m)^{1+o(1)}$ in such a way that allows us to reduce to fast matrix multiplication to perform a dominance check as in Eisenbrand and Grandoni [16]. Notably, this approach introduces a polynomial overhead for K_3 and K_4, as the created matrices are too large compared to the lower bound. We handle these cases separately: Using a careful combination of ideas from sparse Triangle Counting [4] and sparse 2-Dominating Set [21], we are able to improve the algorithms for K_4 and K_3; if $\omega = 2$, this completely eliminates the overhead for K_4 and reduces the overhead for K_3 to $m^{1/3+o(1)}$. Finally, we extend our algorithm beyond connected patterns by showing how to handle isolated nodes:

[4] To clarify the correspondence, we remark that already Alon (see [3, proof of Lemma 8]) observed that in the definition of $\delta(P)$, one may without loss of generality let S range only over independent sets: for any set S maximizing $|S| - |N(S)|$, the set $S' = S \setminus (S \cap N(S))$ is an independent set with $|S'| - |N(S')| \geq |S| - |N(S)|$. Thus, whenever a nonempty set S maximizes $|S| - |N(S)|$, we have $\rho(P) = \delta(P)$. Alon proves that a connected m-edge graph G has a maximum number of $\Theta(m^{\frac{|V(P)|+\delta(P)}{2}})$ induced copies of P, where P is any connected pattern. The generalization to an arbitrary trade-off between n and m is implicit in our proofs.

We can no longer enumerate all candidate sets, as this number becomes too large. Instead, we apply a recursive approach similar to the algorithm for Dominating Independent Sets given in [30]. Here, particularly the case of a single isolated node requires great care.

Due to space reasons, our conditional lower bounds had to be deferred to the full version of the paper. Our conditional lower bound construction generalizes the ones in [30] (which in turn are based on [36, 21]): Specifically, we reduce from a version of the k-OV problem with carefully chosen set sizes. Intuitively, the parameter $\rho(P)$ spells out how to choose these sizes: (1) for every vertex $v \in S$, we have a set of n vectors, (2) for every vertex $v \in N(S)$, we have a set of m/n vectors, and (3) for every vertex $v \in V(P) \setminus (S \cup N(S))$, we have a set of $\sqrt{m}$ vectors. The precise construction requires care; in particular if $S = \emptyset$, we need to use an alternative choice of a single set of m/n vectors and $k-1$ sets of $\sqrt{m}$ vectors.

We additionally study variants of Dominating P-Pattern where we are given a set $\mathcal{Q}$ of patterns rather than a single pattern. The corresponding problem Dominating $\mathcal{Q}$-Pattern asks to detect a dominating set S that induces a subgraph isomorphic to *some* $P \in \mathcal{Q}$. A notable special case is the Non-Induced Dominating P-Pattern problem in which the task is to detect a dominating set S such that $G[S]$ contains P as a subgraph. Combining the following theorem with Theorems 1.2 and 1.3, our results settle the fine-grained complexity of Non-Induced Dominating P-Pattern for any pattern P except K_3 (assuming $\omega = 2$).

▶ **Theorem 1.5** (Dominating $\mathcal{Q}$-Pattern, Informal version)**.** *Let Q be a finite set of patterns of identical order. The fine-grained complexity of Dominating $\mathcal{Q}$-Pattern is dominated by the pattern $P \in \mathcal{Q}$ with the highest time complexity.*

Due to the space restrictions, the proof of this theorem is provided only in the full version of the paper.

2 Preliminaries

For a $n \in \mathbb{N}$, $[n]$ denotes the set $\{1, \ldots, n\}$. For a set X, its power set is denoted by 2^X and the set of all subsets of size k by $\binom{X}{k}$. For two sets X, Y, by $X \times Y$ we denote a set of unordered pairs, i.e. $X \times Y := \{\{x, y\} \mid x \in X, y \in Y\}$. For a pattern graph P and a host graph G, we use k and n, respectively, to denote their order, i.e., number of vertices. As a shorthand for $\{u, v\}$, the notation uv denotes an edge between u and v. The set $N(v) = \{u \mid uv \in E\}$ is the *neighborhood* of v. The *closed neighborhood* of v is $N[v] = N(v) \cup \{v\}$. For a set of vertices $S \subseteq V$, by $N(S)$ (resp. $N[S]$), we denote the set $\bigcup_{v \in S} N(v)$ (resp. $\bigcup_{v \in S} N[v]$). Moreover, the subgraph of G *induced* by S is denoted by $G[S]$. For a vertex $v \in V$, the graph $G - v$ denotes the graph G with the vertex v deleted. Likewise, for a set $X \subseteq V$, we use $G - X := G[V \setminus X]$. A set of vertices $D \subseteq V$ dominates the graph G if every vertex of v is either in D or has a neighbor in D.

The matrix multiplication exponent $\omega \geq 2$ is the smallest constant such that there is an algorithm multiplying any two $n \times n$ matrices in $n^{\omega + o(1)}$ time. For two rectangular matrices of size $n_1 = n \times n_2$ and $n_2 \times n_3$, $\mathrm{MM}(n_1, n_2, n_3)$ denotes the time complexity of multiplying these matrices. Similarly, for matrix of size $n^a \times n^b$ and $n^b \times n^c$, $n^{\omega(a,b,c)+o(1)}$ denotes the time complexity of multiplying them. Further, $\alpha \leq 1$ is the largest constant such that $\omega(1, \alpha, 1) = 2$. The best known bounds for α and ω are $\omega \leq 2.3713$ [2], and $\alpha \geq 0.3213$ [41].

3 Algorithm for Pattern Dominating Set

In this section we develop the algorithms for Dominating P-Pattern problem for different choices of patterns P and prove the following main theorem as stated in the Introduction.

▶ **Theorem 1.2** (Upper Bound). *For any pattern graph P with at least 16 vertices, there is an algorithm solving Dominating P-Pattern problem in time $t_P(n,m)^{1+o(1)}$. Further, if $\omega = 2$, this algorithm exists for all patterns P except K_3.*

In order to prove this theorem, we proceed in three steps. We first develop algorithms for a class of patterns P we call *basic*: all patterns that contain no isolated vertices and are not isomorphic to a K_3 or K_4. We then handle patterns with isolated vertices and the remaining small cliques K_3, K_4 separately. An important ingredient that we use to speed up the dominance check is an approach via fast matrix multiplication due to Eisenbrand and Grandoni [16].

▶ **Lemma 3.1.** *Let X, Y be the sets of vertices and $\phi : 2^X \to \{0,1\}$ be a predicate such that for any $D \subseteq X$ we can check $\phi(D)$ in constant time. Let $\mathcal{V}_A, \mathcal{V}_B \subseteq 2^X$ be sets of subsets of X such that any subset $D \subseteq X$ that satisfies ϕ can be written as a union of two sets $A \in \mathcal{V}_A$ and $B \in \mathcal{V}_B$. Then in time $\mathrm{MM}(|\mathcal{V}_A|, |Y|, |\mathcal{V}_B|)$ we can enumerate all subsets of X that satisfy ϕ and dominate Y.*

We remark that in the k-Dominating Set algorithm [16], the predicate $\phi(D)$ from the previous lemma is true if $|D| = k$ and false otherwise. In our case, for a fixed pattern P, the predicate $\phi(D)$ will be true if the subgraph of G induced by D is isomorphic to P (i.e. $G[D] \cong P$).[5] Recall also our definition of the parameter $\rho(P)$, which will be relevant in this section.

▶ **Definition 1.1.** *Let P be an arbitrary pattern graph, and denote by $I(P) \subseteq V(P)$ the set of isolated nodes of P. Choose an* independent *set $S \subseteq V(P) \setminus I(P)$ maximizing $|S| - |N(S)|$; if S is not uniquely defined, take any choice maximizing $|S|$. We define*

$$\rho(P) \coloneqq \begin{cases} |S| - |N(S)| & \text{if } S \neq \emptyset \\ -1 & \text{if } S = \emptyset. \end{cases}$$

and set

$$t_P(n,m) \coloneqq n^{\rho(P)} \cdot m^{\frac{|V(P)| - \rho(P)}{2}}.$$

Let S be a (possibly empty) independent set of non-isolated vertices in P that maximizes the value $|S| - |N(S)|$ as in Definition 1.1. We define the *remainder* set R to be $V(P) \setminus N[S]$. The following observation follows directly from the fact that S is an independent set.

▶ **Observation 3.2.** *Let P be a pattern on k vertices and $S, N(S), R$ be as defined above. Then $S, N(S), R$ form a partition of $V(P)$.*

Before proceeding with the algorithm, we state another observation, that gives us a nice way to think about the value of $t_P(n,m)$ depending on whether set S is empty or not.

▶ **Observation 3.3.** *Let P be a pattern on k vertices and $S, N(S), R$ be the partition of $V(P)$ as defined above.*

- *If $S = \emptyset$, then $t_P(n,m) = \frac{m^{(k+1)/2+o(1)}}{n}$.*
- *If $S \neq \emptyset$, then $t_P(n,m) = t_{P[N[S]]}(n,m) \cdot m^{|R|/2}$.*

If we consider a partition into sets $S, N(S)$ and R similar as above, intuitively we have some structure on the parts S (independent set in P) and $N(S)$ (there exists a maximal matching that matches each vertex in $N(S)$ to a vertex in S), but we have very little structure on how the remainder set looks like. In the following lemma we show that we can decompose the remainder set R into much simpler subgraphs (disjoint edges and cycles).

---

[5] Note that verifying ϕ takes $f(|P|) = \mathcal{O}(1)$ time, for constant size patterns.

▶ **Lemma 3.4.** *Let S, $N(S)$ and R be a partition of $V(P)$ as defined above and assume that the induced subgraph $P[R]$ contains no isolated vertices. Then the following holds.*

- *There exists a spanning subgraph of $P[R]$ that is isomorphic to a disjoint union of edges and odd cycles.*
- *For any vertex $x \in R$, there exists a spanning subgraph of $P[R - x]$ that is isomorphic to a disjoint union of edges and odd cycles.*

The proof follows from a structural theorem proved in [11] together with the choice of the set S in Definition 1.1. For details we refer the reader to the full version of the paper. We are now equipped with all the tools we need to construct the algorithm for the first family of patterns P that we call *basic patterns.*

3.1 Basic Patterns

In this section we construct an algorithm that solves the P-Dominating Set problem in the (conditionally) optimal time for most of the patterns. More precisely, we say a pattern graph P is *basic* if: 1) it has at least three vertices; 2) it has no isolated vertices; 3) it is neither isomorphic to a K_3 nor a K_4. We prove the following theorem for basic patterns.

▶ **Theorem 3.5.** *For any basic pattern P on at least 16 vertices, there exists an algorithm solving the Dominating P-Pattern problem in time $t_P(n,m)^{1+o(1)}$. Moreover, if $\omega = 2$, this time complexity can be achieved for all basic patterns.*

The high level idea is to first decompose the pattern into (not necessarily induced) odd cycles, edges and isolated vertices, and then enumerate all possible *valid* choices for each of those parts efficiently, and then use Lemma 3.1 to check if the union of guessed parts induces a subgraph isomorphic to P, and if it dominates G. We first show that we can enumerate all subgraphs isomorphic to the remainder set R efficiently. Let $k := |V(P)|$ and $n := |V(G)|$. We say a vertex $v \in V(G)$ is *heavy* if $\deg(v) \geq \frac{n}{k} - 1$. We prove the following lemma in the full version of the paper.

▶ **Lemma 3.6.** *Let P be a basic pattern. Let $S, N(S), R$ be a partition of $V(P)$ as defined in Definition 1.1. Then the following holds.*

1. *We can enumerate all subgraphs of G that are isomorphic to $P[R]$ in time $\mathcal{O}(m^{|R|/2})$.*
2. *We can enumerate all subgraphs of G that are isomorphic to $P[R]$ and contain a heavy vertex in time $\mathcal{O}(\frac{m^{(|R|+1)/2}}{n})$.*

We now use the lemma above to handle the case when the decomposition into S, $N(S)$ and R as in Definition 1.1 yields $S = \emptyset$. Due to space constraints, we only sketch the proof here. The details can be found in the full version.

▶ **Lemma 3.7.** *Let P be a basic pattern and $S, N(S), R$ be a partition of $V(P)$ as defined in Definition 1.1, such that $S = \emptyset$. If the basic pattern P has at least 16 vertices, then there exists an algorithm solving Dominating P-Pattern in time $\left(\frac{m^{\frac{k-1}{2}}}{n}\right)^{1+o(1)} = t_P(n,m)^{1+o(1)}$. If $\omega = 2$, this time complexity can be achieved for all basic patterns P.*

Proof (sketch). If $S = \emptyset$, then $V(P) = R$ and Lemma 3.6 applies to all of P. Recall that any dominating set of size k contains a heavy vertex v_h (a vertex of degree at least $n/k - 1$), hence we can first guess this vertex in time $\mathcal{O}(\frac{m}{n})$. By Lemma 3.4, there is a spanning subgraph of P that is isomorphic to a disjoint union of edges and odd cycles and we can find such a spanning subgraph in $f(k) = \mathcal{O}(1)$ time. Assume that this decomposition yields α odd cycles and β disjoint edges. If $\beta > 0$, we show that we can decompose $P - v_h$ further into two graphs B_1, B_2 such that they satisfy the following conditions:

- Each B_i is a disjoint union of independent P_3's (paths of length two) and K_2's (independent edges), such that each copy of P_3 will correspond to a subgraph of G that is isomorphic to P_3 and whose all vertices either have degree at most $\sqrt{m}$, or one of the endpoints has degree at least $\sqrt{m}$. [6]
- $B_1 \cup B_2$ is a spanning subgraph of $P - v_h$.
- (Without loss of generality) $0 \leq |V(B_1)| - |V(B_2)| \leq 2$.
- We can enumerate all copies of G isomorphic to B_i that satisfy the degree constraints of P_3 in time $\mathcal{O}(m^{|B_i|/2})$.

In particular, for $k \geq 16$, we have $|V(B_1)| + |V(B_2)| = k - 1 \geq 15$, and combining this with the inequality above, we can conclude that $|V(B_1)| \geq |V(B_2)| \geq 7$. Using that the rectangular matrix multiplication constant $\alpha \geq 0.321$, and that $m \geq n$, we for the exponent

$$\omega(1, \frac{2}{|V(B_1')|}, 1) = \omega(1, \frac{2}{|V(B_2')|}, 1) = 2, \tag{1}$$

and hence we obtain for all $k \geq 16$:

$$\begin{aligned}
\mathcal{O}\left(\frac{m}{n}\,\mathrm{MM}\left(m^{\frac{|V(B_1')|}{2}}, n, m^{\frac{|V(B_2')|}{2}}\right)\right) &\leq \mathcal{O}\left(\frac{m}{n}\,\mathrm{MM}\left(m^{\frac{|V(B_1')|}{2}}, m, m^{\frac{|V(B_2')|}{2}}\right)\right) && (m \geq n)\\
&\leq \mathcal{O}\left(\frac{m}{n} \cdot m^{\frac{|V(B_1')|+|V(B_2')|}{2}+o(1)}\right) && (\text{Eq. } 1)\\
&= \mathcal{O}\left(\frac{m}{n} \cdot m^{\frac{k-1}{2}+o(1)}\right) = \left(\frac{m^{\frac{k+1}{2}}}{n}\right)^{1+o(1)}.
\end{aligned}$$

If on the other hand $\beta = 0$ (i.e. the decomposition contains only copies of odd cycles), we proceed similarly, but the best bound on the size difference between B_1 and B_2 is $0 \leq |V(B_1)| - |V(B_2)| \leq 3$, which is not good enough in general. To get around this issue, we distinguish between two cases. If there exists a vertex in $P - v_h$ that is mapped to a vertex of degree $\geq \sqrt{m}$ in G, we can make sure that there exists a copy of P_3 whose one endpoint will be mapped to a vertex in G that has degree at least $\sqrt{m}$. We can decompose this P_3 further into an edge and a vertex of degree $\geq \sqrt{m}$, and intuitively use these components to "re-balance" the sets $V(B_1)$ and $V(B_2)$ as above. If on the other hand all vertices in $P - v_h$ are mapped to vertices in G that have degree at most $\sqrt{m}$, we then know that if $|Y| := |G - N_G[v_h]| \geq (k-1)\sqrt{m} + 1$, there is no valid solution. If on the other hand $|Y| \leq (k-1)\sqrt{m}$, by Lemma 3.1, we can find any such solution in time bounded by $\mathcal{O}\left(\frac{m}{n}\,\mathrm{MM}\left(m^{\frac{|V(B_1)|}{2}}, \sqrt{m}, m^{\frac{|V(B_2)|}{2}}\right)\right)$. Similarly as in previous case, for $k \geq 16$, we get that $|V(B_1)| \geq |V(B_1)| \geq 6$ and again since $\alpha \geq 0.321$, we obtain $\omega(1, \frac{1}{|V(B_1)|}, 1) = \omega(1, \frac{1}{|V(B_2)|}, 1) = 2$. Hence, we can bound the time complexity as

$$\begin{aligned}
\mathcal{O}\left(\frac{m}{n}\,\mathrm{MM}\left(m^{\frac{|V(B_1)|}{2}}, \sqrt{m}, m^{\frac{|V(B_2)|}{2}}\right)\right) &\leq \mathcal{O}\left(\frac{m}{n} \cdot \left(\sqrt{m}\right)^{|V(B_1)|+|V(B_2)|+o(1)}\right)\\
&= \mathcal{O}\left(\frac{m}{n} \cdot \left(\sqrt{m}\right)^{k-1+o(1)}\right) = \left(\frac{m^{\frac{k+1}{2}}}{n}\right)^{1+o(1)} \qquad \blacktriangleleft
\end{aligned}$$

The second case for basic patterns is that the decomposition into S, $N(S)$ and R as in Definition 1.1 yields $S \neq \emptyset$. Intuitively this case is simpler than the one above, since we can use the sets $S, N(S)$ to simulate $\beta > 0$, even if $\beta = 0$ and "balance" the sets $V(B_1)$ and $V(B_2)$. We only sketch the proof, as the details are quite similar to the ones above.

---

[6] Hence in G we will be able to enumerate all subgraphs isomorphic to P_3 that fulfill this degree requirement in time $\mathcal{O}(m^{3/2})$.

▶ **Lemma 3.8.** *Let P be a basic pattern and $S, N(S), R$ be a partition of $V(P)$ as defined in Definition 1.1, such that $S \neq \emptyset$. If the basic pattern P has at least 16 vertices, then there exists an algorithm solving Dominating P-Pattern in time $\left(n^{|S|-|N(S)|}m^{|N(S)|+\frac{|R|}{2}}\right)^{1+o(1)} = t_P(n,m)^{1+o(1)}$. If $\omega = 2$, this time complexity can be achieved for all basic patterns P.*

Proof (sketch). Similarly as above, construct the sets B_1 and B_2 by finding a spanning subgraph of $P[R]$ that consists of union of independent P_3's and K_2's and distributing the connected components as equally as possible. Clearly, we have that $|V(B_1)| + |V(B_2)| = |R|$, and (without loss of generality) $0 \leq |V(B_1)| - |V(B_2)| \leq 3$. Since $S \neq \emptyset$, also $N(S) \neq \emptyset$ (by definition S contains no isolated vertices), hence we can distribute the edges from a perfect matching between the set $N(S)$ and some subset of S of size $|N(S)|$ (such perfect matching always exists by Hall's marriage theorem) between the sets B_1 and B_2 similarly as in previous lemma to obtain graphs B_1', B_2'. Clearly, it holds that $|V(B_1')| + |V(B_2')| = 2|N(S)| + |R|$ and (without loss of generality) $0 \leq |V(B_1')| - |V(B_2')| \leq 2$. We then define the graphs Q_1, Q_2 that consist of $\lfloor \frac{|S|-|N(S)|}{2} \rfloor$ and $\lceil \frac{|S|-|N(S)|}{2} \rceil$ isolated vertices respectively. Similarly as above, we can enumerate all subgraphs of G isomorphic to $B_i' \cup Q_i$ in time $\mathcal{O}\left(n^{|Q_i|}m^{\frac{|B_i'|}{2}}\right)$. Also, by similar arguments as above, we have that for $k \geq 16$, the matrices are "thin enough" that rectangular matrix multiplication techniques can be used to obtain the tight running time of

$$\mathrm{MM}\left(n^{|Q_1|}m^{\frac{|B_1'|}{2}}, n, n^{|Q_2|}m^{\frac{|B_2'|}{2}}\right) \leq \left(n^{|Q_1|+|Q_2|}m^{(|B_1'|+|B_2'|)/2}\right)^{1+o(1)} \qquad (k \geq 16)$$
$$\leq \left(n^{|S|-|N(S)|}m^{\frac{2|N(S)|+|R|}{2}}\right)^{1+o(1)}$$ ◀

3.2 Small Cliques

In this section, we deal with the remaining small patterns, i.e., the cliques of size at most 4. We note that in [30] the K_2 was already settled, so it remains to settle the cliques of size 3 and 4. In particular, this section is dedicated to proving the following two main theorems.

▶ **Theorem 3.9.** *For a Dominating Triangle, there exists a deterministic $\left(\frac{m^{1+\frac{2\omega}{\omega+1}}}{n}\right)^{1+o(1)}$-time algorithm and a randomized $m^{\frac{\omega+1}{2}+o(1)}$-time algorithm.*

▶ **Theorem 3.10.** *There is a randomized algorithm solving K_4-Dominating Set in time $\left(\frac{m^{(\omega+3)/2}}{n}\right)^{1+o(1)}$. If $\omega = 2$, this is $\left(\frac{m^{5/2}}{n}\right)^{1+o(1)} = t_{K_4}(n,m)^{1+o(1)}$.*

We start with the simple deterministic algorithm for Dominating Triangle problem that uses a reduction to the *All-Edges-Triangle-Counting* problem. This problem asks for a given tripartite graph $G = (V_1 \cup V_2 \cup V_3, E)$ to determine, for every edge e in $(V_1 \times V_2) \cap E$, how many triangles in G contain e. It is well known that this problem can be solved in time $m^{\frac{2\omega}{\omega+1}+o(1)}$ (see e.g. [4]). We remark that any $\mathcal{O}(m^{\frac{4}{3}-\varepsilon})$-time algorithm would refute the 3-SUM and the APSP Hypothesis [35, 40], giving matching lower bounds if $\omega = 2$.

▶ **Lemma 3.11** (All-Edge Triangle Counting, see [4]). *There is an algorithm solving All-Edges-Triangle-Counting in time $m^{\frac{2\omega}{\omega+1}+o(1)}$.*

By using this lemma, we are able to solve the problem in $\mathcal{O}(t_{K_3}(n,m)) \cdot m^{\frac{\omega-1}{\omega+1}+o(1)}$, which evaluates to $\mathcal{O}(t_{K_3}(n,m)) \cdot m^{\frac{1}{3}+o(1)}$ if $\omega = 2$. By a more careful approach, we are able to construct a randomized algorithm that uses a bloom-filter inspired approach similar to the one in [21], to achieve $\mathcal{O}\left(t_{K_3}(n,m)\right) \cdot \frac{n}{\sqrt{m}}$, which is better whenever $m \in \Omega(n^{1.10283})$ with current value of ω and $m \in \Omega(n^{1.2})$ if $\omega = 2$, giving us the second part of the Theorem 3.9. We remark that although this value can be slightly improved by using a more careful analysis, it seems that in order to match the lower bound, a new technique will be required. We now sketch the approach of the deterministic algorithm and give the details to both algorithms in the full version of the paper.

▶ **Proposition 3.12.** *There exists a deterministic algorithm solving Dominating Triangle in time* $\left(\frac{m^{1+\frac{2\omega}{\omega+1}}}{n}\right)^{1+o(1)}$. *If* $\omega = 2$, *this equals* $\left(\frac{m^{7/3}}{n}\right)^{1+o(1)}$.

Proof (sketch). We first guess a heavy vertex w (a vertex satisfying $\deg(w) \geq n/3 - 1$). We then run a breadth-first-search from w and let $(X_0 = \{w\}, X_1, \ldots, X_n)$ be the obtained BFS layering. Note that for any $\ell \geq 3$, if X_ℓ is non-empty, we can conclude that there is no dominating clique that contains w and we can proceed with the next choice of the heavy vertex. Otherwise, finding any dominating triangle in G that contains w is equivalent to finding an edge u, v in X_1 that dominates all vertices in X_2. For any vertex u, we write $N_2(u) := N(u) \cap X_2$. Using the principle of inclusion-exclusion, we can write $|N_2(u) \cup N_2(v)| = |N_2(u)| + |N_2(v)| + |N_2(u) \cap N_2(v)|$. We note that for any pair of vertices $u, v \in X_1$ the value $|N_2(u) \cap N_2(v)|$ corresponds to the number of triangles containing the edge uv. Hence, by employing the algorithm from Lemma 3.11, we obtain the desired time. ◀

On the other hand, by coming up with a more involved argument, we are able to construct a randomized algorithm that matches the lower bound (up to resolving the matrix multiplication exponent ω) for the remaining small pattern, namely Dominating 4-Clique. The algorithm starts similarly as above. That is, we first guess a heavy vertex w in time $\mathcal{O}\left(\frac{m}{n}\right)$, and then we partition the vertices into two sets X_1 and X_2 based on the distance from w and thus reduce to detecting a triangle in X_1 that dominates X_2. We then consider two cases based on the size of X_2. If $|X_2| \leq 3\sqrt{m}$, this turns out to be quite easy, and boils down to using standard matrix multiplication algorithms (with a bit more careful analysis). If on the other hand $|X_2| > 3\sqrt{m}$, we then either reduce to counting cliques (triangles and K_4's) in a certain restricted setting, or use the randomized bloom-filter based approach of [21] (with a finer-grained analysis) to achieve the desired running time. This approach gives us the proof of Theorem 3.10 and we provide the detailed proof in the full version of the paper.

3.3 Handling Isolated Vertices

In the last two subsections we considered the graphs that have no isolated vertices. It remains to prove that we can obtain a tight classification even for patterns with isolated vertices. In particular, we prove the following theorem.

▶ **Theorem 3.13.** *Let* P *be any pattern graph with* k *vertices and* $1 \leq r \leq k$ *isolated vertices. There is a randomized algorithm that enumerates all dominating* P*-patterns in time* $t_P(n,m) \cdot m^{\frac{\omega-2}{2}+o(1)}$ *with high probability. If* $\omega = 2$, *this is* $t_P(n,m)^{1+o(1)}$.

We again consider two cases separately, namely when $r \geq 2$ and $r = 1$. In the first (simpler) case of $r \geq 2$, we aim to reduce to the Maximal r-Independent Set problem, which is known to have efficient algorithm in sparse graphs (see [30]). We enumerate all subgraphs of G that correspond to the induced subgraph of P that does not contain isolated vertices, and then after removing the closed neighborhood of the enumerated subgraph, we run the Maximal

r-Independent Set algorithm on the remaining part. By a careful analysis, we obtain the desired time. We provide a sketch of the proof here, for the full proof, see the full version of the paper.

▶ **Lemma 3.14.** *Let P be a pattern with $2 \le r \le k$ isolated vertices. Then there exists a randomized algorithm solving P-Dominating Set in time $\left(t_P(n,m) \cdot m^{(\omega-2)/2}\right)^{1+o(1)}$ with high probability. If $\omega = 2$, this is equal to $(t_P(n,m))^{1+o(1)}$.*

Proof (sketch). If $r = k$, then this problem is precisely the Maximal k-Independent Set problem, and by [30] can be solved in the time bounded by $\left(\frac{m^{(k-1)/2+\omega/2}}{n}\right)^{1+o(1)}$, as desired. Hence, we may assume that $2 \le r \le k-2$ (if P contains at least one edge, then at least two vertices have degree ≥ 1). Let $X := \{x \in V(P) : \deg_P(x) \ge 1\}$ and $Y := V(P) \setminus X$ be the set of non-isolated and isolated vertices in P respectively. Let $S, N(S), R$ be the partition of $V(P)$ as in Definition 1.1. We can enumerate all induced subgraphs of G isomorphic to $P[X]$ in some time $\mathcal{O}\left(t^{(1)}_{P[X]}(n,m)\right)$ (the ratio of the size of this set to $t_{P[X]}(n,m)$ depends only on whether the set S is empty or not). For each enumerated subgraph H, we appeal to the randomized Dominating r-Independent Set algorithm from [30] on the graph $G - N[H]$ that runs in $\mathcal{O}\left(t^{(2)}_r(n,m)\right)$ (depends on which vertex from P is mapped to a heavy vertex). It is straightforward to see that this algorithm is correct. We conclude the proof by showing that the product of $t^{(1)}_{P[X]}(n,m)$ and $t^{(2)}_r(n,m)$ is in all cases bounded by $t_P(n,m)$ (see the full version of the paper for details). ◀

Things get slightly more complicated when dealing with patterns that only have one isolated vertex. In particular, the preprocessing part of the algorithm above takes in the worst case $\Theta\left(t_{P[X]}(n,m) \cdot m\right)$, while for pattern graphs with only one isolated vertex, we have $t_P(n,m) = \mathcal{O}\left(t_{P[X]}(n,m) \cdot \sqrt{m}\right)$. To circumvent this overhead, we employ a more careful analysis and use a hashing-based approach similar to that in Section 3.2. We first provide an algorithm for two special cases of some of the smallest patterns that exhibit this structure and then the idea is to reduce any pattern to one of these special cases.

▶ **Lemma 3.15.** *Let G be a graph with n vertices and m edges, and let $Y \subseteq V(G)$ be a subset of vertices of G. Then we can construct an algorithm that w.h.p. enumerates all choices of three vertices u, v, w in time $m^{(\omega+1)/2+o(1)}$, such that the following is satisfied.*

- *The vertex w is contained in Y and satisfies $\deg(w) < \sqrt{m}$.*
- *Vertices u and v are adjacent in G and both of them are non-adjacent to w in G (i.e. the induced subgraph $G[\{u,v,w\}]$ is isomorphic to an edge and an isolated vertex).*
- *$Y \subseteq N[u] \cup N[v] \cup N[w]$.*

Proof. Without loss of generality, assume that $\deg(u) \ge \deg(v)$. Let t be such that $t \le \deg(u) \le 2t$. Assume first that $t \le \sqrt{m}$. Then each vertex u, v, w has degree bounded by $\mathcal{O}(\sqrt{m})$ and so, if there exists a valid triple that dominates Y, the size of Y must also be bounded by $\mathcal{O}(\sqrt{m})$. Hence by Lemma 3.1 we can enumerate all valid triples in time $\mathrm{MM}(m, \sqrt{m}, \sqrt{m}) \le m^{(\omega+1)/2+o(1)}$, as desired.[7] For the rest of this proof we assume that $t \ge \sqrt{m}$. We now proceed to enumerate all valid triples u, v, w such that $\deg(v) \le \sqrt{m}$. To this end, we run the following routine. For each of the $\mathcal{O}(m/t)$ valid choices for the vertex u, construct the tripartite graph $H_u = (V_u^{(1)} \cup V_u^{(2)} \cup V_u^{(3)}, E_u)$, where $V_u^{(1)}$ corresponds to a

[7] Since vertices u, v are adjacent, there are at most m choices for this pair, and w is by assumption in Y, and $|Y| \le \mathcal{O}(\sqrt{m})$

copy of $N(u)$, while $V_u^{(2)}$ and $V_u^{(3)}$ each correspond to a copy of the set $Y - N(u)$. The set of edges E_u is constructed naturally, that is, for each pair $x \in V_u^{(i)}, y \in V_u^{(j)}$ for $i \neq j$ we add an edge xy if and only if the edge between the corresponding vertices is present in G (we assume that each vertex is adjacent to itself in G). It is easy to see that for any fixed u, any valid choice of v, w corresponds precisely to a choice of $v \in V_u^{(1)}$ and $w \in V_u^{(2)}$ such that the following two conditions are satisfied: 1) $V_u^{(3)} \subseteq N(w) \cup N(v)$ and 2) $vw \notin E(H_u)$. Now, since we are only enumerating triples u, v, w where both v, w have degrees at most $\sqrt{m}$, if for a fixed u the set $V_u^{(3)}$ contains more than $2\sqrt{m}$ vertices, we can conclude that no pair v, w satisfies both conditions and proceed with the next choice of u. On the other hand, if $|V_u^{(2)}| = |V_u^{(3)}| \leq 2\sqrt{m}$, by Lemma 3.1, for any fixed u, we can enumerate all the valid pairs of non-adjacent vertices v, w that dominate $Y - N(u)$ in time

$$\begin{aligned}
T_u(n,m) &\leq \mathrm{MM}\left(V_u^{(1)}, V_u^{(3)}, V_u^{(2)}\right) \\
&\leq \mathrm{MM}\left(t, \sqrt{m}, \sqrt{m}\right) \\
&\leq \mathcal{O}\left(\frac{t}{\sqrt{m}}\right) \mathrm{MM}\left(\sqrt{m}, \sqrt{m}, \sqrt{m}\right) && (t \geq \sqrt{m}) \\
&\leq t \cdot m^{\frac{\omega-1}{2}+o(1)}.
\end{aligned}$$

Repeating this for each of the possible $\mathcal{O}(m/t)$ many choices of vertex u yields the desired time. It remains to enumerate the triples of vertices u, v, w where $\deg(v) \geq \sqrt{m}$. Recall also that $2t \geq \deg(u) \geq \deg(v)$ and $t \geq \sqrt{m}$. Hence, since by the first condition of the Lemma, we are only interested in the triples u, v, w with $\deg(w) \leq \sqrt{m} \leq t$, if the set Y contains more than $5t$ vertices, we can conclude that no triple u, v, w satisfies $Y \subseteq N[u] \cup N[v] \cup N[w]$. Hence, we can assume that $Y \leq 5t$. Let B be a matrix whose rows correspond to adjacent pairs of vertices u, v with $t \leq \deg(u) \leq 2t$ and $\deg(v) \geq \sqrt{m}$ and whose columns correspond to the set Y. We set the entry $B[(u,v), y]$ to 0 if either u or v are adjacent to y and to 1 otherwise. Similarly, we construct a matrix C whose columns correspond to the set of vertices $w \in Y$ such that $\deg(w) \leq \sqrt{m}$ and the rows correspond to Y. Set the entry $B[y, w]$ to 1 if w and y are adjacent (we understand that each vertex is adjacent to itself) and 0 otherwise. Notice that u, v, w dominate Y if and only if $(B \cdot C)[(u,v), w] = \sum_k B[w,k]$ (each vertex that is not dominated by the pair (u, v) is dominated by w). Also notice that the matrix B is an $\mathcal{O}\left(\frac{m\sqrt{m}}{t}\right) \times \mathcal{O}(t)$ and C is an $\mathcal{O}(t) \times \mathcal{O}(t)$ matrix with at most m non-zeros. Hence by a Lemma from [21] (see full version for details), we can enumerate with high probability all valid triples u, v, w in time $\sqrt{m} \cdot m^{\omega/2+o(1)} = m^{\frac{\omega+1}{2}+o(1)}$, as desired. We remark that we enumerate all valid triples for any fixed t in the claimed time. To enumerate all the valid triples, simply run the previous algorithm $\mathcal{O}(\log(n))$ times, once for each $\ell \in 1, \ldots, \lfloor \log(n) \rfloor$, setting $t = 2^\ell$. ◀

An almost identical argument as in previous lemma can also be used to show that we can efficiently enumerate the solutions that are isomorphic to $C_{2r+1} + K_1$.

▶ **Lemma 3.16.** *Let G be a graph with n vertices and m edges, and let $Y \subseteq V(G)$ be a subset of vertices of G. Then we can construct an algorithm that w.h.p. enumerates all choices of $2r+2$ vertices $x_1, \ldots, x_{2r+1}, w$ in time $m^{r+\omega/2+o(1)}$, such that the following is satisfied.*

- *The vertex w is contained in Y and satisfies $\deg(w) < \sqrt{m}$.*
- *Vertices $x_1, \ldots, x_{2r+1}$ form a cycle in G (not necessarily induced) and w is not adjacent to any of the vertices x_i in G.*
- *$Y \subseteq N[w] \cup N[x_1] \cup \cdots \cup N[x_{2r+1}]$.*

The idea for any pattern with precisely one isolated vertex v is to decompose the pattern into sets $S, N(S), R$ as in Definition 1.1, and then depending on whether the set S is empty or not, we either directly reduce to the setting of Lemma 3.15, or first apply Lemma 3.4 to decompose R into edges and odd cycles, and then reduce to either the setting of Lemma 3.15, or that of Lemma 3.16. For the lack of space, we give the detailed proof in the the full version of the paper.

▶ **Proposition 3.17.** *Let P be a pattern with one isolated vertex. Then there exists a randomized algorithm solving P-Dominating Set in time $\mathcal{O}\left(t_P(n,m)\right) \cdot m^{\frac{\omega-2}{2}+o(1)}$.*

References

1 Amir Abboud, Arturs Backurs, and Virginia Vassilevska Williams. If the current clique algorithms are optimal, so is Valiant's parser. *SIAM Journal on Computing*, 47(6):2527–2555, 2018. doi:10.1137/16M1061771.

2 Josh Alman, Ran Duan, Virginia Vassilevska Williams, Yinzhan Xu, Zixuan Xu, and Renfei Zhou. More asymmetry yields faster matrix multiplication. In Yossi Azar and Debmalya Panigrahi, editors, *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2025, New Orleans, LA, USA, January 12-15, 2025*, pages 2005–2039. SIAM, 2025. doi:10.1137/1.9781611978322.63.

3 Noga Alon. On the number of subgraphs of prescribed type of graphs with a given number of edges. *Israel Journal of Mathematics*, 38(1):116–130, March 1981. doi:10.1007/BF02761855.

4 Noga Alon, Raphael Yuster, and Uri Zwick. Finding and counting given length cycles. *Algorithmica*, 17(3):209–223, 1997. doi:10.1007/BF02523189.

5 Alkida Balliu, Sebastian Brandt, Fabian Kuhn, and Dennis Olivetti. Distributed maximal matching and maximal independent set on hypergraphs. In Nikhil Bansal and Viswanath Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 2632–2676. SIAM, 2023. doi:10.1137/1.9781611977554.CH100.

6 Parinya Chalermsook, Marek Cygan, Guy Kortsarz, Bundit Laekhanukit, Pasin Manurangsi, Danupon Nanongkai, and Luca Trevisan. From gap-eth to fpt-inapproximability: Clique, dominating set, and more. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 743–754. IEEE Computer Society, 2017. doi:10.1109/FOCS.2017.74.

7 Victor Chepoi. A note on r-dominating cliques. *Discrete mathematics*, 183(1-3):47–60, 1998. doi:10.1016/S0012-365X(97)00076-9.

8 Eun-Kyung Cho, Ilkyoo Choi, Hyemin Kwon, and Boram Park. A tight bound for independent domination of cubic graphs without 4-cycles. *Journal of Graph Theory*, 104(2):372–386, 2023. doi:10.1002/JGT.22968.

9 Eun-Kyung Cho, Ilkyoo Choi, and Boram Park. On independent domination of regular graphs. *Journal of Graph Theory*, 103(1):159–170, 2023. doi:10.1002/JGT.22912.

10 Eun-Kyung Cho, Jinha Kim, Minki Kim, and Sang-il Oum. Independent domination of graphs with bounded maximum degree. *Journal of Combinatorial Theory, Series B*, 158:341–352, 2023. doi:10.1016/J.JCTB.2022.10.004.

11 Gérard Cornuéjols and William R. Pulleyblank. Critical graphs, matching and tours or a hierarchy of relaxations for the traveling salesman problem. *Comb.*, 3(1):35–52, 1983. doi:10.1007/BF02579340.

12 Margaret B Cozzens and Laura L Kelleher. Dominating cliques in graphs. *Discrete Mathematics*, 86(1-3):101–116, 1990. doi:10.1016/0012-365X(90)90353-J.

13 Mina Dalirrooyfard and Virginia Vassilevska Williams. Induced cycles and paths are harder than you think. In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022, Denver, CO, USA, October 31 - November 3, 2022*, pages 531–542. IEEE, 2022. doi:10.1109/FOCS54457.2022.00057.

14 Mina Dalirrooyfard, Thuy-Duong Vuong, and Virginia Vassilevska Williams. Graph pattern detection: Hardness for all induced patterns and faster noninduced cycles. *SIAM J. Comput.*, 50(5):1627–1662, 2021. doi:10.1137/20M1335054.

15 Feodor F Dragan and Andreas Brandstädt. Dominating cliques in graphs with hypertree structure. In *STACS 94: 11th Annual Symposium on Theoretical Aspects of Computer Science Caen, France, February 24–26, 1994 Proceedings 11*, pages 735–746. Springer, 1994. doi:10.1007/3-540-57785-8_186.

16 Friedrich Eisenbrand and Fabrizio Grandoni. On the complexity of fixed parameter clique and dominating set. *Theor. Comput. Sci.*, 326(1-3):57–67, 2004. doi:10.1016/J.TCS.2004.05.009.

17 Yibin Fang and Liming Xiong. Circumference of a graph and its distance dominating longest cycles. *Discrete Mathematics*, 344(2):112196, 2021. doi:10.1016/j.disc.2020.112196.

18 Yibin Fang and Liming Xiong. On the dominating (induced) cycles of iterated line graphs. *Discrete Applied Mathematics*, 325:43–51, 2023. doi:10.1016/j.dam.2022.09.025.

19 Jill R. Faudree, Ralph J. Faudree, Ronald J. Gould, Paul Horn, and Michael S. Jacobson. Degree sum and vertex dominating paths. *J. Graph Theory*, 89(3):250–265, 2018. doi:10.1002/JGT.22249.

20 Ralph J. Faudree, Ronald J. Gould, Michael S. Jacobson, and Douglas B. West. Minimum degree and dominating paths. *J. Graph Theory*, 84(2):202–213, 2017. doi:10.1002/JGT.22021.

21 Nick Fischer, Marvin Künnemann, and Mirza Redzic. The effect of sparsity on k-dominating set and related first-order graph properties. In David P. Woodruff, editor, *Proc. 2024 ACM-SIAM Symposium on Discrete Algorithms (SODA 2024)*, pages 4704–4727. SIAM, 2024. doi:10.1137/1.9781611977912.168.

22 Teresa W Haynes and Peter J Slater. Paired-domination in graphs. *Networks: An International Journal*, 32(3):199–206, 1998. doi:10.1002/(SICI)1097-0037(199810)32:3\%3C199::AID-NET4\%3E3.0.CO;2-F.

23 Karthik C. S. and Subhash Khot. Almost polynomial factor inapproximability for parameterized k-clique. In Shachar Lovett, editor, *37th Computational Complexity Conference, CCC 2022, July 20-23, 2022, Philadelphia, PA, USA*, volume 234 of *LIPIcs*, pages 6:1–6:21. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICS.CCC.2022.6.

24 Karthik C. S., Bundit Laekhanukit, and Pasin Manurangsi. On the parameterized complexity of approximating dominating set. *J. ACM*, 66(5):33:1–33:38, 2019. doi:10.1145/3325116.

25 Ton Kloks, Dieter Kratsch, and Haiko Müller. Finding and counting small induced subgraphs efficiently. *Inf. Process. Lett.*, 74(3-4):115–121, 2000. doi:10.1016/S0020-0190(00)00047-8.

26 Ekkehard Köhler. Connected domination and dominating clique in trapezoid graphs. *Discret. Appl. Math.*, 99(1-3):91–110, 2000. doi:10.1016/S0166-218X(99)00127-4.

27 Dieter Kratsch. Finding dominating cliques efficiently, in strongly chordal graphs and undirected path graphs. *Discrete mathematics*, 86(1-3):225–238, 1990. doi:10.1016/0012-365X(90)90363-M.

28 Dieter Kratsch, Peter Damaschke, and Anna Lubiw. Dominating cliques in chordal graphs. *Discrete Mathematics*, 128(1-3):269–275, 1994. doi:10.1016/0012-365X(94)90118-X.

29 Kirsti Kuenzel and Douglas F Rall. On independent domination in direct products. *Graphs and Combinatorics*, 39(1):7, 2023. doi:10.1007/S00373-022-02600-0.

30 Marvin Künnemann and Mirza Redzic. Fine-grained complexity of multiple domination and dominating patterns in sparse graphs. In Édouard Bonnet and Pawel Rzazewski, editors, *Proc. 19th International Symposium on Parameterized and Exact Computation (IPEC 2024)*, volume 321 of *LIPIcs*, pages 9:1–9:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPICS.IPEC.2024.9.

31 Bingkai Lin, Xuandi Ren, Yican Sun, and Xiuhan Wang. Improved hardness of approximating k-clique under ETH. In *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023, Santa Cruz, CA, USA, November 6-9, 2023*, pages 285–306. IEEE, 2023. doi:10.1109/FOCS57990.2023.00025.

32 Dániel Marx. Can you beat treewidth? *Theory Comput.*, 6(1):85–112, 2010. doi:10.4086/TOC.2010.V006A005.

33 Jaroslav Nešetřil and Svatopluk Poljak. On the complexity of the subgraph problem. *Commentationes Mathematicae Universitatis Carolinae*, 26(2):415–419, 1985.

34 Shiwei Pan, Yiming Ma, Yiyuan Wang, Zhiguo Zhou, Jinchao Ji, Minghao Yin, and Shuli Hu. An improved master-apprentice evolutionary algorithm for minimum independent dominating set problem. *Frontiers of Computer Science*, 17(4):174326, 2023. doi:10.1007/S11704-022-2023-7.

35 Mihai Pătraşcu. Towards polynomial lower bounds for dynamic problems. In Leonard J. Schulman, editor, *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pages 603–610. ACM, 2010. doi:10.1145/1806689.1806772.

36 Mihai Pătraşcu and Ryan Williams. On the possibility of faster SAT algorithms. In Moses Charikar, editor, *Proc. 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2010)*, pages 1065–1075. SIAM, 2010. doi:10.1137/1.9781611973075.86.

37 Justin Southey and Michael A Henning. A characterization of graphs with disjoint dominating and paired-dominating sets. *Journal of combinatorial optimization*, 22(2):217–234, 2011. doi:10.1007/S10878-009-9274-1.

38 Daniel S Studer, Teresa W Haynes, and Linda M Lawson. Induced-paired domination in graphs. *Ars Combinatoria*, 57:111–128, 2000.

39 Virginia Vassilevska Williams, Joshua R. Wang, Richard Ryan Williams, and Huacheng Yu. Finding four-node subgraphs in triangle time. In Piotr Indyk, editor, *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 1671–1680. SIAM, 2015. doi:10.1137/1.9781611973730.111.

40 Virginia Vassilevska Williams and Yinzhan Xu. Monochromatic triangles, triangle listing and APSP. In Sandy Irani, editor, *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 786–797. IEEE, 2020. doi:10.1109/FOCS46700.2020.00078.

41 Virginia Vassilevska Williams, Yinzhan Xu, Zixuan Xu, and Renfei Zhou. New bounds for matrix multiplication: from alpha to omega. In David P. Woodruff, editor, *Proceedings of the 2024 ACM-SIAM Symposium on Discrete Algorithms, SODA 2024, Alexandria, VA, USA, January 7-10, 2024*, pages 3792–3835. SIAM, 2024. doi:10.1137/1.9781611977912.134.

42 Henk Jan Veldman. Existence of dominating cycles and paths. *Discret. Math.*, 43(2-3):281–296, 1983. doi:10.1016/0012-365X(83)90165-6.