

The Effect of Sparsity on k -Dominating Set and Related First-Order Graph Properties

Nick Fischer
Weizmann Institute of Science *

Marvin Künnemann
Karlsruhe Institute of Technology †

Mirza Redzic
Karlsruhe Institute of Technology †

Abstract

We revisit the classic k -Dominating Set problem. Besides its importance as perhaps the most natural $W[2]$ -complete problem, it is among the first problems for which a tight $n^{k-o(1)}$ conditional lower bound (for all sufficiently large k), based on the Strong Exponential Time Hypothesis (SETH), was shown (Pătraşcu and Williams, SODA 2007). Notably, however, the underlying reduction creates dense graphs, raising the question: how much does the sparsity of the graph affect its fine-grained complexity?

As our first result, we settle the fine-grained complexity of k -Dominating Set in terms of both the number of nodes n and number of edges m , up to resolving the matrix multiplication exponent ω . Specifically, on the hardness side, we show an $mn^{k-2-o(1)}$ lower bound based on SETH, for any dependence of m on n . On the algorithmic side, this is complemented by an $mn^{k-2+o(1)}$ -time algorithm for all sufficiently large k . For the smallest non-trivial case of $k = 2$, i.e., 2-Dominating Set, we give a randomized algorithm that employs a Bloom-filter inspired hashing to improve the state of the art of $n^{\omega+o(1)}$ to $m^{\omega/2+o(1)} = O(m^{1.187})$. If $\omega = 2$, this yields a conditionally tight bound for all $k \geq 2$.

To study whether k -Dominating Set is special in its sensitivity to sparsity, we study the effect of sparsity on very related problems:

- The k -Dominating Set problem belongs to a type of first-order definable graph properties that we call *monochromatic basic problems*. These problems are the canonical monochromatic variants of the basic problems that were proven complete for the class FOP of first-order definable properties (Gao, Impagliazzo, Kolokolova, and Williams, TALG 2019). We show that among the monochromatic basic problems, the k -Dominating Set property is the *only* property whose fine-grained complexity decreases in sparse graphs. Only for the special case of reflexive properties is there an additional basic problem that can be solved faster than $n^{k \pm o(1)}$ on sparse graphs.
- For the natural variant of distance- r k -dominating set, we obtain a hardness of $n^{k-o(1)}$ under SETH for every $r \geq 2$ already on sparse graphs, which is tight for sufficiently large k .

*This work is part of the project CONJEXITY that has received funding from the European Research Council (ERC) under the European Union’s Horizon Europe research and innovation programme (grant agreement No. 101078482).

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

1 Introduction

Consider an algorithmic graph problem whose best known algorithm runs in time $\tilde{\mathcal{O}}(n^c)$, where n denotes the number of vertices and $c \geq 2$ is some (usually small) constant. While improving this running time might resist intensive effort and even suffer from a conditional lower bound, it might still be possible to obtain substantial, *polynomial-factor* improvements by taking into account the sparsity of the given graph – after all, far from all interesting graphs are dense. A typical target to shoot for is a time bound of $\tilde{\mathcal{O}}(m^{c/2})$, where m denotes the number of edges in the graph, or (*edge*) *sparsity*. Such a running time is never worse than the known bound for dense graphs, i.e., $m = \Theta(n^2)$, but polynomially improves the running time for all sparser graphs, i.e., when $m = \mathcal{O}(n^{2-\epsilon})$.

On the algorithmic side, obtaining such an algorithm may be technically challenging (consider, e.g., the global min-cut problem [25, 24]) or even turn out to be conditionally impossible: E.g., turning the $n^{\omega+o(1)}$ -time state-of-the-art algorithm for All-Edge Triangle Detection to an $m^{\omega/2+o(1)}$ -time algorithm would refute the 3SUM and APSP hypotheses [31, 34].

While most works on the fine-grained complexity of graph problems analyze the time complexity either purely in n (we call this the *dense* case) or purely in m (we call this the *sparse* case), some recent works even specifically address the full trade-off between m and n , e.g. [1, 30]: Agarwal and Ramachandran [1] give several sparsity-preserving reductions from the shortest cycle problem, giving evidence of optimality of natural $\tilde{\mathcal{O}}(mn)$ -time algorithms. Subsequently, Lincoln et al. [30] even manage to prove that the weighted k -clique hypothesis implies optimality of the $\tilde{\mathcal{O}}(mn)$ -time bound for shortest cycle, specifically for $m = \Theta(n^\gamma)$ for infinitely many $1 \leq \gamma \leq 2$. This yields conditional optimality for several APSP-related problems, such as directed and undirected APSP, radius, replacement paths, and more. Further related work addresses, e.g., the influence of sparsity for (unweighted) k -cycle detection [13, 29].

In this paper we aim to advance this line of research by settling the effect of sparsity on interesting graph properties, most notably the k -Dominating Set problem.

1.1 The Effect of Sparsity on k -Dominating Set

We revisit the central graph problem k -Dominating Set for $k \geq 2$: Given an undirected graph $G = (V, E)$, determine whether there is a k -sized set S of vertices such that each vertex $v \in V$ is *dominated* by S (i.e., $v \in S$ or there exists $u \in S$ with $\{u, v\} \in E$). It is among the classic NP-hard problems, counts as perhaps *the* most natural $W[2]$ -complete problem [15] (see also [16, 12]) and suffers from strong fine-grained inapproximability results, see, e.g. [10, 26]. It gives rise to the notion of domination number in graph theory and has inspired a plethora of related problems (e.g., edge domination, total domination, partial domination, connected domination, capacitated domination and many more); see, e.g., [22] for a dedicated monograph.

In the dense setting, the fine-grained complexity of k -Dominating Set is well understood (up to resolution of fast matrix multiplication): Eisenbrand and Grandoni [19] show how to solve k -Dominating Set in time $n^{k+o(1)}$ for all $k \geq 8$. For $2 \leq k \leq 7$, the (small) polynomial overhead to the $\mathcal{O}(n^k)$ running time depends on the complexity of fast rectangular matrix multiplication. In particular, if $\omega = 2$, k -Dominating Set can be solved in time $n^{k+o(1)}$ for all $k \geq 2$.

On the lower bound side, Pătraşcu and Williams [32] show that an $\mathcal{O}(n^{k-\epsilon})$ algorithm for any $k \geq 3$ would refute the Strong Exponential Time Hypothesis (SETH).¹ Notably, this reduction creates dense graphs, and in particular does not give any lower bound for the case of $k = 2$. We thus ask:

¹In fact, the lower bound can be based on the k -OV hypothesis, see Section 2 for a definition.

Question 1: How does sparsity affect the time complexity of k -Dominating Set?

Perhaps surprisingly, it has been observed in [8, Footnote 5] that k -Dominating Set indeed admits faster than $n^{k \pm o(1)}$ algorithms in sparse graphs. The idea is simple: Any dominating set of size k must contain at least one member that dominates at least n/k vertices. Combining this idea with the algorithm of Eisenbrand and Grandoni [19], we obtain the following baseline, which depends on the optimal exponent $\omega(a, b, c)$ of multiplying a $n^a \times n^b$ matrix with a $n^b \times n^c$ matrix (and $\omega = \omega(1, 1, 1)$).

Proposition 1.1 (*k -Dominating Set Baseline*). *Let $k \geq 2$ and $1 \leq \gamma \leq 2$. The k -Dominating Set on graphs with n nodes and $m = \Theta(n^\gamma)$ edges can be solved in time $n^{\omega+o(1)} + n^{\omega(\lceil \frac{k-1}{2} \rceil, 1, \lfloor \frac{k-1}{2} \rfloor + \gamma - 1) + o(1)}$. If $\omega = 2$, this running time becomes $n^{2+o(1)} + mn^{k-2+o(1)}$.*

For $k \geq 3$ (and assuming that $\omega = 2$), this improves significantly over the running time for the dense case by a factor of $\Theta(n^2/m)$. However, under Pătraşcu and Williams' lower bound [32], we could hope for even better speed-ups – possibly even for an algorithm running in time $m^{k/2+o(1)}$. As our first contribution, we show that the $\Theta(n^2/m)$ -improvement by Proposition 1.1 is best-possible, assuming the k -Orthogonal Vectors Hypothesis, which is well-known to be implied by the Strong Exponential Time Hypothesis (see Section 2).

Theorem 1.2 (*k -Dominating Set Lower Bound*). *For all $k \geq 3$ and $\epsilon > 0$, there is no algorithm for k -Dominating Set in time $\mathcal{O}(mn^{k-2-\epsilon})$, unless the k -OV Hypothesis fails.*

Curiously, Theorem 1.2 leaves open an important special case, as it provides no non-trivial lower bound for $k = 2$. At the same time, Proposition 1.1 also does not improve over the best known upper bound of $n^{\omega+o(1)}$ due to Eisenbrand and Grandoni [19]. This brings us to an unclear situation: Can we improve the $n^{\omega+o(1)}$ -time algorithm for 2-Dominating Set on sparse graphs, or can the lower bound from Theorem 1.2 be strengthened?

Our main algorithmic result is that for 2-Dominating Set we indeed can improve upon Proposition 1.1:

Theorem 1.3 (*2-Dominating Set Algorithm*). *There is a randomized algorithm solving 2-Dominating Set in time $m^{\omega/2+o(1)}$.*

Note that if $\omega = 2$, this yields an almost-optimal $m^{1+o(1)}$ -time algorithm for 2-Dominating Set. More generally, if $\omega = 2$, our results conditionally establish that $mn^{k-2+o(1)}$ is the *optimal* running time for k -Dominating Set for all $k \geq 2$, up to subpolynomial factors.

We remark that if $\omega > 2$, our algorithm achieves an even better running time for very small graph densities ($m \leq n^{\frac{2+\omega}{4}} \leq n^{1.094}$). Specifically, our running time is never worse than $\tilde{\mathcal{O}}(m^2/n)$ and thus near-linear in for very sparse graphs with $m = \tilde{\mathcal{O}}(n)$. See Section 3 for more details.

1.2 Beyond k -Dominating Set: Monochromatic First-Order Graph Properties

The non-trivial influence of sparsity on the complexity of k -Dominating Set raises the question how general this phenomenon is:

Question 2: For which related graph problems does sparsity influence the time complexity?

To approach this question systematically, we observe that k -Dominating Set is a first-order definable property of the following form: Given an undirected graph $G = (V, E)$ with $|V| = n$ and $|E| = m$, decide if

$$\exists v_1 \in V \dots \exists v_k \in V : \forall w \in V : E(v_1, w) \vee \dots \vee E(v_k, w).$$

(In this formulation, we assume that the edge predicate is symmetric and reflexive, i.e., $E(v, v)$ for all $v \in V$.)

There are many interesting problems that may be formulated as such a first-order definable graph property, such as existence of a given k -vertex pattern, existence of a k -sized set of vertices sharing no common neighbor, the property of having a graph diameter 2, and many more. By allowing an even more general formulation², one arrives at the class FOP defined by Gao, Impagliazzo, Kolokolova and Williams [21]: For any first-order definable property ϕ , FOP contains the corresponding problem of deciding ϕ over a given relational structure (in many interesting cases, simply a graph).

General algorithmic results for this class have been obtained by Williams [35] for graph properties in the dense case (where we consider the universe size n as main parameter) and by [21] for general properties in the sparse case (where we consider the total size of the relational structure m as main parameter). Specifically, Williams showed that all $(k + 1)$ -quantifier first-order graph properties can be solved in time $\mathcal{O}(n^{k+o(1)})$ for $k \geq 8$ (which would even hold for all $k \geq 2$ if $\omega = 2$) and additionally gave a SETH-based lower bound of $n^{k-o(1)}$ for some properties. Gao et al. [21] show that all $(k + 1)$ -quantifier properties can be solved in time $\mathcal{O}(m^k)$, and for each $k \geq 2$, determine a list of $(k + 1)$ -quantifier problems, called *basic problems* of order k , to be *complete* for this class in the following sense: An $\mathcal{O}(m^{k-\epsilon})$ -time algorithm for any of these complete problems would give a polynomial improvement over the $\mathcal{O}(m^k)$ -time algorithm for all problems with $k' + 1$ quantifiers where $k' \geq k$. We shall call a problem that is complete in this sense an FOP_k -*complete* problem. For any $k \geq 2$, the basic problems of order k have the following form:

$$\exists v_1 \in V_1 \dots \exists v_k \in V_k \forall w \in W : \ell_1 \vee \dots \vee \ell_k,$$

where each literal ℓ_i is of the form $E(v_i, w)$ or $\overline{E(v_i, w)}$

Put differently, to obtain a basic problem, one must choose, for each $i \in [k]$, precisely one of $E(v_i, w)$ and its negation $\overline{E(v_i, w)}$. Note that this establishes the basic problems as fine-grained equivalent, *hardest* problems in FOP. Among these basic problems, we find the k -OV problem (see also [21] for a detailed discussion), and a problem that is usually not formulated in graph language: k -Set Cover, see below.

Monochromatic vs. Bichromatic: k -Dominating Set vs. k -Set Cover. In the k -Set Cover problem, the input consists of a set family \mathcal{S} over universe U , and the question is whether there are k sets from \mathcal{S} that cover U , i.e.,

$$\exists S_1 \in \mathcal{S} \dots \exists S_k \in \mathcal{S} : \forall u \in U : u \in S_1 \vee \dots \vee u \in S_k.$$

By introducing a set $S_v := N[v]$ for all $v \in V$, this problem generalizes the k -Dominating Set problem. In fact, k -Set Cover can be equivalently viewed as a *bichromatic* version of k -Dominating Set (also known as Red-Blue Dominating Set): Define a 2-partite graph $G = (\mathcal{S} \cup U, E)$ where for any $S_i \in \mathcal{S}, u \in U$, we have $\{S_i, u\} \in E$ if and only if $u \in S_i$. Then the task is to determine a set of k vertices S_1, \dots, S_k chosen from \mathcal{S} such that S_1, \dots, S_k dominate all vertices in U .

Pătraşcu and Williams observe that their $n^{k-o(1)}$ conditional lower bounds for k -Dominating Set extends to k -Set Cover. In fact, it is not difficult to see (and implicit in [32]) that the reduction for k -Dominating Set can be slightly simplified to establish hardness of k -Set Cover already for all $k \geq 2$ (rather than $k \geq 3$). Moreover, the hardness reduction produces sparse instances for which $\sum_{i=1}^n |S_i| = \mathcal{O}(n)$. Thus, our results in Section 1.1 *separate* k -Set Cover (the bichromatic

²Specifically, allowing an arbitrary number of relations of arbitrary constant arity, not just a single edge relation.

variant) from k -Dominating Set (the monochromatic variant), as the effect of sparsity differs for both problems. This leads to the natural question whether monochromatic versions are always easier to solve on sparse graphs than on general graphs.

Monochromatic Basic Problems. To address this question in some generality, we perform a comprehensive study on the *monochromatic basic problems*, i.e., the canonical *monochromatic* versions of the basic problems of FOP. Our monochromatic basic problems have the form:

$$\exists \text{ pairwise distinct } v_1, \dots, v_k \in V \forall w \in V : \ell_1 \vee \dots \vee \ell_k,$$

where each literal ℓ_i is of the form $E(v_i, w)$ or $\overline{E(v_i, w)}$

Note that here we introduce as additional requirement that the existentially quantified variables v_1, \dots, v_k are pairwise distinct. For monochromatic properties, this requirement is indeed the usually intended meaning – otherwise, most of the basic problems become trivial.³ Our monochromatic basic graph problems (of order k) contain several natural examples:

- *k-Dominating Set*: Is there a subset of k vertices dominating all vertices?
- *Neighborhood Containment*: Are there distinct $v_1, v_2 \in V$ such that $N(v_1) \subseteq N(v_2)$?
- *Neighborhood $(k - 1)$ -Covering*: Is there a vertex whose neighborhood can be covered by the neighbors of $k - 1$ other vertices?
- *k-Empty-Neighborhood-Intersection*: Are there k vertices that have no common neighbor?
- *$(k - 1)$ -Common Neighborhood*: Is there a vertex whose neighbors are common neighbors of $k - 1$ other vertices?

Although the (multichromatic) basic problems are all fine-grained equivalent (by being FOP_k -complete), we show that among the monochromatic basic problems, k -Dominating Set is surprisingly different:

Theorem 1.4 (Basic Problems Lower Bound). *Let $k \geq 2$ and $\epsilon > 0$. If any of the monochromatic basic problems except k -Dominating Set can be solved in time $\mathcal{O}(n^{k-\epsilon})$ on graphs with $n^{1+o(1)}$ edges, then the k -OV Hypothesis is false.*

This establishes that k -Dominating Set is the *only* monochromatic basic problem that becomes easier on sparse graphs, answering our driving Question 2.

Interestingly, this result also shows the fine-grained equivalence of almost all basic problems to their monochromatic versions – proving such multichromatic-to-monochromatic reductions for simpler first-order properties, such as detection of certain patterns of size k , would contradict established hardness assumptions in fine-grained complexity theory. As a case in point, one can show that the 4-chromatic version of 4-cycle detection conditionally requires $n^{\omega-o(1)}$ time under the triangle detection hypothesis, while the monochromatic version of 4-cycle detection is solvable in time $\mathcal{O}(n^2)$ [36].

Special Cases: Reflexivity vs. Irreflexivity. We highlight an additional reason why Theorem 1.4 appears surprising, as it does *not* hold for the special case of *reflexive* properties. Specifically, our definition of monochromatic basic properties allows for the existence of self-loops, i.e., $E(v, v)$ with $v \in V$, to be specified individually for each $v \in V$, as part of the input. It may be

³Specifically, whenever the basic problem contains both a positive disjunct $E(v_i, v)$ and a negative disjunct $\overline{E(v_j, v)}$, the property is trivially satisfied by choosing $v_i = v_j$.

reasonable to either disallow self-loops (i.e., require $\overline{E(v, v)}$ for all $v \in V$; we call this the *irreflexive case*) or enforce self-loops (i.e., require $E(v, v)$ for all $v \in V$; we call this the *reflexive case*). The reflexive special case generally expresses problems over closed neighborhoods and the irreflexive special case expresses problems over open neighborhoods.

Our proof of Theorem 1.4 establishes the same hardness for sparse graphs in the *irreflexive case*. For the *reflexive case*, however, it turns out that there exists an additional basic problem for which $n^{k \pm o(1)}$ time can be broken for sparse graphs: Closed Neighborhood $(k - 1)$ -Covering. In this problem the task is to detect distinct v_1, \dots, v_k vertices such that the closed neighborhood of v_1 is covered by the closed neighborhoods of v_2, \dots, v_k , i.e., $N[v_1] \subseteq N[v_2] \cup \dots \cup N[v_k]$. We design an algorithm running in time $m^{\frac{2\omega}{\omega+1}} n^{k-2} \leq \mathcal{O}(m^{1.41} n^{k-2})$. This beats running time $n^{k \pm o(1)}$ whenever $m \leq n^{1 + \frac{1}{\omega} - o(1)}$ (the exponent is roughly 1.42 under the current value of ω). To our surprise, we further establish that in the reflexive case, k -Dominating Set and Closed Neighborhood $(k - 1)$ -Covering are the *only* monochromatic basic problems that are influenced by sparsity.

Distance- r Domination. A popular generalization of k -Dominating Set is the Distance- r k -Dominating Set problem, see, e.g., [18, 28, 14]. In this variant, a vertex v is dominated by $S \subseteq V$, if there exists some $s \in S$ with distance at most r from v . In Section 5, we show that Distance- r k -Dominating Set is affected by sparsity if and only if $r = 1$: For $r = 1$, we obtain the usual k -Dominating Set problem with the complexity mn^{k-2} established in Section 1.1 if $\omega = 2$. For any $r \geq 2$, we prove a hardness of $n^{k-o(1)}$ under SETH already in sparse graphs, which is tight for all $k \geq 2$ if $\omega = 2$.

Related Work. Investigating the fine-grained complexity of classes of first-order definable problems has recently gained traction, see, e.g., [35, 21, 8, 6, 7, 5].

Establishing hardness results for monochromatic settings generally appears to be technically challenging: For additive problems, specifically 3-Linear Degeneracy Testing, [17] exploit involved constructions from additive combinatorics (k -sum-free sets) to establish the equivalence of monochromatic and multichromatic variants. Another example is the geometric setting of Closest Pair in the Euclidean Metric, for which a fine-grained equivalence between the bichromatic and monochromatic case could be shown [27].

1.3 Technical Overview

We give an outline of our most interesting technical ideas. Specifically, we sketch the algorithmic improvements for k -Dominating Set for small values of k , as well as our general reduction from multichromatic basic properties except k -Dominating Set to their monochromatic versions. All further contributions are detailed in their respective technical sections.

Algorithmic Contributions for k -Dominating Set. To exploit sparsity for k -Dominating Set, the first crucial observation is that in any k -dominating set $\{v_1, \dots, v_k\}$ there must exist a node v_i of degree at least $n/k - 1$. Let H denote the set of such nodes; we clearly have that $|H| \leq \frac{2m}{(n/k-1)} = \mathcal{O}(m/n)$. Thus, we may restrict our search for a dominating set to the search space $H \times V^{k-1}$ of size $\mathcal{O}(mn^{k-2})$.

However, naively testing each set in $H \times V^{k-1}$ still requires an overhead of $\mathcal{O}(n)$ per candidate solution. In this way we cannot beat running time $n^{k \pm o(1)}$, so we have to be more careful. It is only natural to try to adapt the approach of Eisenbrand and Grandoni [19]. Let us consider the case of $k = 2$: For any sets $S, T \subseteq V$, we denote by $A_{S,T}$ the adjacency matrix of G restricted to

$S \times T$ (i.e., $A_{S,T}$ is the 0-1 matrix whose rows are indexed by S , whose columns are indexed by T , and whose entries are defined by $A_{S,T}[s,t] = 1$ iff $\{s,t\} \in E$ or $s = t$). Furthermore, let \overline{A} denote the complement of A . Then it holds that

$$(\overline{A_{H,V}} \cdot \overline{A_{V,V}})[v_1, v_2] = 0 \text{ iff } N[v_1] \cup N[v_2] = V,$$

i.e., v_1, v_2 form a 2-dominating set. This reduces 2-dominating set to the multiplication of a rectangular $\mathcal{O}(m/n) \times n$ matrix $\overline{A_{H,V}}$ with a square $n \times n$ matrix $\overline{A_{V,V}}$.

Since already the input size for this matrix product is of size $\Theta(n^2)$, this approach again cannot directly achieve a $m^{1+o(1)}$ -time algorithm. To avoid this, one might hope to use techniques for sparse matrix multiplication (see, e.g., [3, 37]), since the adjacency matrix of a sparse graph has $\mathcal{O}(m)$ nonzeros. However, the *non*-adjacency matrix of a sparse graph as required here necessarily has $\Theta(n^2)$ nonzeros. Fortunately, we can still formulate the problem as a sparse matrix multiplication: Specifically, we have

$$(\overline{A_{H,V}} \cdot A_{V,V})[v_1, v_2] = \overline{\deg(v_1)} \text{ iff } N[v_1] \cup N[v_2] = V, \quad (1)$$

where $\overline{\deg(v)} = n - \deg(v)$ denotes the number of nonzeros of the row corresponding to v in $\overline{A_{H,V}}$. Note that the number of nonzeros of $\overline{A_{H,V}}$ (as an $\mathcal{O}(m/n) \times n$ matrix) and $A_{V,V}$ (as adjacency matrix of an m -edge graph) is $\mathcal{O}(m)$. Using sparse matrix multiplication/triangle counting, we can solve this problem in time $\mathcal{O}(m^{\frac{\omega}{2\omega+1}})$ [3]. Still, this does not yield linear-time complexity even if $\omega = 2$ – in particular, current sparse matrix multiplication techniques fail to beat $n^{2\pm o(1)}$ running time even for multiplying a $\sqrt{n} \times n$ matrix with a sparse $n \times n$ matrix containing $\mathcal{O}(n\sqrt{n})$ nonzeros [4].

Our crucial contribution is that, perhaps surprisingly, we can compute the special case outlined by (1) faster than computing the full matrix product $B := \overline{A_{H,V}} \cdot A_{V,V}$. To this end, we partition V into logarithmically many groups V_i ($0 \leq i \leq \lfloor \log n \rfloor$) consisting of all nodes with degree in $[2^i, 2^{i+1})$. For each $v_2 \in V_i$, we observe that it can only form a 2-dominating set with some $v_1 \in H$ if $\overline{\deg(v_1)} \leq \deg(v_2) < 2^{i+1}$. This simple observation allows us to employ a Bloom-filter-like approach: Let H_i denote set of nodes $v_1 \in H$ with $\overline{\deg(v_1)} \leq 2^{i+1}$. We construct hash functions $h_1, \dots, h_L : V \rightarrow [O(2^i)]$ with $L = \mathcal{O}(\log n)$ to reduce the inner dimension of the matrix multiplication $\overline{A_{H_i,V}} \cdot A_{V,V_i}$ to size $L \cdot \mathcal{O}(2^i)$, such that with high probability, any entry in the result matrix corresponding to $v_1 \in H, v_2 \in V_i$ is equal to the number of nonzeros in v_1 's row if and only if v_1, v_2 are a 2-dominating set. That this is possible is due to the special structure of (1) (and would fail for more general decision problems for sparse matrix products). In total, we perform $\mathcal{O}(\log n)$ multiplications – namely, for each $0 \leq i \leq \lfloor \log n \rfloor$, we multiply a $\mathcal{O}(m/n) \times \tilde{\mathcal{O}}(2^i)$ by a $\tilde{\mathcal{O}}(2^i) \times \min\{\frac{m}{2^i}, n\}$ matrix. This can be shown to take time $m^{\omega/2+o(1)}$. In particular, this algorithm runs in almost-linear time $m^{1+o(1)}$ if $\omega = 2$. We give all details in Section 3.

Hardness for Monochromatic Properties. Recall that for the class of k -Dominating Set-like problems, the monochromatic basic problems, we prove that any problem other than k -Dominating Set conditionally requires time $n^{k-o(1)}$. For the sketch of this proof, let us consider as a simple exemplary property, the Neighborhood Containment problem: \exists distinct $v_1, v_2 \in V : N(v_1) \subseteq N(v_2)$. Already for this simple problem, we face many technical challenges.

As for all monochromatic basic problems, there is a known hardness reduction (from the k -OV problem) to its multichromatic variant $\exists v_1 \in V_1 \exists v_2 \in V_2 \forall w \in W : E(v_1, w) \Rightarrow E(v_2, w)$. A natural attempt would be to add auxiliary vertices V_A that enforce any solution (v_1, v_2) with $N(v_1) \subseteq N(v_2)$ to be chosen from $V_1 \times V_2$. However, great care has to be taken for these auxiliary vertices: any node in V_A could be chosen itself as v_1 or v_2 . Furthermore, if *any* node in V_1 is

connected to some auxiliary node $z \in V_A$, then *all* nodes in V_2 need to be connected to z . To keep the whole instance sparse, this requires adding only a very small number of auxiliary nodes with carefully chosen connections. The situation gets more intricate for more complicated properties such as \exists pairwise distinct $x_1, \dots, x_4 \in V : N(v_1) \cap N(v_2) \subseteq N(v_3) \cup N(v_4)$, where no two auxiliary nodes z_1, z_2 may have disjoint neighborhoods, since otherwise we could set $v_1 = z_1, v_2 = z_2$ and get a trivial solution with any v_3, v_4 . For proving hardness, this shows that no gadget can act fully locally, but must take into account the full graph.

To nevertheless prove hardness for all monochromatic properties except k -Dominating Set, we proceed via an intermediate step of bichromatic properties. For any $\exists^k \forall$ -quantified first-order property ϕ , we distinguish between its *multichromatic*, *bichromatic* and *monochromatic* versions defined as follows:

- *Multichromatic*: $\exists v_1 \in V_1 \dots \exists v_k \in V_k \forall w \in W \phi(v_1, \dots, v_k, w)$.
- *Bichromatic*: \exists pairwise distinct $x_1, \dots, x_k \in X \forall y \in Y \phi(x_1, \dots, x_k, y)$.
- *Monochromatic*: \exists pairwise distinct $v_1, \dots, v_k \in V \forall v \in V \phi(v_1, \dots, v_k, v)$.

Step 1: From Multichromatic to Bichromatic. The first step is to reduce a multichromatic problem to its bichromatic version: We show how to construct, for any subset $S \subseteq X$, a *solution-excluding gadget* Y_S (with corresponding edges) such that no choice of pairwise distinct vertices $x_1, \dots, x_k \in S$ can satisfy $\forall y \in Y_S : \phi(x_1, \dots, x_k, y)$. Let us call a variable x_i a *positive* variable if $E(x_i, y)$ occurs in ϕ and a *negative* variable if $\overline{E}(x_i, y)$ occurs in ϕ . Furthermore, let $\phi(x)$ be a unique identifier for $x \in S$ consisting of $\mathcal{O}(\log n)$ bits. The main idea is that for every pair of a positive variable x_s and a negative variable x_j we can find a bit position $i_j^{(s)} \in [O(\log n)]$ where their identifiers $\phi(x_s)$ and $\phi(x_j)$ differ. For every guess of such bit positions and the corresponding bit values $\phi(x_j)[i_j^{(s)}]$, we introduce a corresponding node in Y_S and connect it in such a way to S such that for every choice of pairwise distinct x_1, \dots, x_k , the node for the correct guess is adjacent to all negative variables and non-adjacent to all positive variables. The number of nodes in Y_S is at most $\mathcal{O}(\log^k n) = \tilde{\mathcal{O}}(1)$.

Equipped with this tool, we can create a bichromatic instance as follows: We set $X = V_1 \cup \dots \cup V_k$. To obtain Y , we start with W , and include, for each subset $S_i = X \setminus V_i, i \in [k]$, a *solution-excluding gadget* Y_{S_i} with possibly additional edges to V_i , enforcing that the only way to satisfy all nodes in Y_{S_i} is to pick $x_i \in V_i$.

Step 2: From Bichromatic to Monochromatic. It remains to reduce the bichromatic to the monochromatic version, at least for the case that Y is of size $\tilde{\mathcal{O}}(1)$; note that the reduction from k -OV to the bichromatic setting as sketched above indeed maintains $|Y| = \tilde{\mathcal{O}}(1)$, so this is sufficient for our purposes. Interestingly, there is a rather simple randomized reduction based on the probabilistic method, and a technically more interesting derandomization. (This is a common phenomenon for fine-grained reductions involving coding-theory-like gadgets.) We proceed as follows: Given a bichromatic instance $G = (X \cup Y, E)$ with $Y = \{y_1, \dots, y_M\}$, we aim to construct an equivalent monochromatic instance $G' = (V', E')$. By a (*partial*) *solution* P , we understand a choice for (a subset of) the variables v_1, \dots, v_k (formally, it would be a mapping $\{v_1, \dots, v_k\} \rightarrow V \cup \{*\}$, where $*$ denotes an unspecified variable – by abuse of notation, we view it as a subset $P \subseteq V$, which hides that we need to distinguish between positive and negative variables). We say that P satisfies $v \in V$ if $\phi(v_1, \dots, v_k, v)$ is satisfied by at least one of the assigned variables in P .

The idea is to construct a graph H with $M^{f(k)} = \tilde{\mathcal{O}}(1)$ nodes such that (1) each node $v \in V(H)$ receives a label $\lambda(v) \in \{1, \dots, M\}$, and (2) for any solution P of size k and every node label $i \in \{1, \dots, M\}$, there exists a node $v \in V(H) \setminus P$ with $\lambda(v) = i$ that is *not satisfied* by $P \cap V(H)$.

Equipped with such an object, we can construct G' by setting $V' = X \cup V(H)$ and adding an edge (1) between $x \in X$ and $v \in V(H)$ iff x and $y_{\lambda(v)}$ were adjacent in G , and (2) between $v, v' \in V(H)$ iff v and v' were adjacent in H . Note that there are no edges within X .

We claim that any satisfying solution x_1, \dots, x_k in G yields a corresponding solution x_1, \dots, x_k in G' : Since ϕ has at least one negative literal, we have that $\phi(x_1, \dots, x_k, x)$ is satisfied for every $x \in X$.⁴ We claim that also $\phi(x_1, \dots, x_k, v)$ is satisfied for all $v \in V(H)$, since $\phi(x_1, \dots, x_k, v_{\lambda(v)})$ is satisfied in G .

Conversely, we claim that if G has no satisfying solution, then also G' has no satisfying solution. Consider any solution P in G' . Since G has no satisfying solution, the partial solution $X \cap P$ must leave at least some vertex $v_i \in Y$ in G unsatisfied. By construction of H , there exists some node $\tilde{v} \in V(H) \setminus P$ with label i such that $P \cap V(H)$ does not yet satisfy \tilde{v} . However, \tilde{v} also cannot be satisfied by $P \cap X$ by definition of i . Thus, \tilde{v} is left unsatisfied by the full solution P .

Constructing the graph H via a randomized algorithm is not too difficult using the probabilistic method. However, we are even able to give an explicit, deterministic construction: The rough idea is to identify the nodes of H with polynomials of degree d over the finite field of size p , for suitably chosen p, d . Any such polynomial f can be equivalently viewed by its evaluations $(f(1), f(2), \dots, f(d+1))$. We will use appropriate parameters $L < d < R$ (specifically, $L = d - k$ and $R = d + dk + 1$) and identify $(f(1), \dots, f(L))$ with the label $\lambda(f)$ and define an edge between f and g iff there exists some $L < x \leq R$ such that $f(x) = g(x)$. Now for any degree- d polynomials a_1, \dots, a_k and b_1, \dots, b_k , and any label (ℓ_1, \dots, ℓ_L) , we can prove existence of some degree- d polynomial f with $f(i) = \ell_i$ for all $i \in [L]$, as well as $\{f, a_j\} \in E$ for all $j \in [k]$ and $\{f, b_j\} \notin E$ for all $j \in [k]$. To do this, one must find $(d + 1) - L$ additional evaluations for f such that for each $j \in [k]$ there exists some $L < x \leq R$ such that $f(x) = a_j(x)$ while at the same time for each $j \in [k]$, we have $f(x) \neq b_j(x)$ for *all* $L < x \leq R$. We give all details in Section 4.

2 Preliminaries

Let n be a positive integer. We denote by $[n]$ the set $\{1, \dots, n\}$. If S is an n -element set and $0 \leq k \leq n$ is an integer, then $\binom{S}{k}$ denotes the set of all k -element subsets of S . We denote by $\mathcal{P}(S)$ the power set of S .

We use $\tilde{\mathcal{O}}$ notation, which hides the poly-logarithmic factors. In other words, $f(n) \in \tilde{\mathcal{O}}(g(n))$ if and only if there exists a $k \in \mathcal{O}(1)$ such that $f(n) \in \mathcal{O}(g(n) \cdot \log^k(n))$.

Let $\omega < 2.3729$ [2] denote the optimal exponent of multiplying two $n \times n$ matrices and $\omega(a, b, c)$ denote the optimal exponent for multiplying an $n^a \times n^b$ matrix by an $n^b \times n^c$ matrix.

Let G be a graph and $X \subseteq V(G)$. Then we denote by $G[X]$ the subgraph of G induced by X . If G, H are isomorphic, we write $G \cong H$. For any vertex $v \in V(G)$, the *neighbourhood* of v is the set of vertices adjacent to v , denoted $N(v)$. The *closed neighbourhood* of v , denoted $N[v]$ is defined as $N[v] := N(v) \cup \{v\}$. The *degree* of v denotes the size of its neighbourhood ($\deg(v) = |N(v)|$). For any two vertices $u, v \in V(G)$, we denote by $d_G(u, v)$ the length of the shortest path between u and v in G .

Hypotheses. Consider the k -Orthogonal Vectors problem (k -OV) that is stated as follows. Given k sets A_1, \dots, A_k of d -dimensional binary vectors, decide whether there exist vectors $a_1 \in A_1, \dots, a_k \in$

⁴This is the argument that crucially fails for the k -dominating set property.

A_k such that for all $t \in [d]$, it holds that $\prod_{i=1}^k a_i[t] = 0$. A simple brute force approach solves the k -OV in time $\mathcal{O}(d \cdot \prod_{i \in [k]} |A_i|)$.

On the other hand, it is well known that an algorithm solving k -OV with $|A_1| = \dots = |A_k| = n$ and $d = \log^2 n$ in time $\mathcal{O}(n^{k-\varepsilon})$ would refute SETH^5 , which follows by combining a split-and-list reduction [32] with the sparsification lemma [23], see [33] for details. This conjecture is known as (*low-dimensional*) k -OV Hypothesis. For the purpose of this paper, we consider a more general formulation where we allow the sets to be of different sizes and hence state the k -OVH as follows.

Conjecture 2.1 (k -OVH). *For no $\varepsilon > 0$ and for no $0 \leq \gamma_1, \dots, \gamma_k \leq 1$ is there an algorithm solving k -OV with $|A_1| = n^{\gamma_1}, \dots, |A_k| = n^{\gamma_k}$, $d = \log^2 n$ in time $\mathcal{O}(n^{(\sum_{i=1}^k \gamma_i) - \varepsilon})$.*

We refer to the setting of k -OVH with $\gamma_1 = \dots = \gamma_k = 1$ as *balanced k -OVH*.

These two hypotheses are known to be equivalent (see [9, Lemma II.1] for a proof for $k = 2$). Below we give a proof for general k .

Lemma 2.2. *Balanced k -OVH and k -OVH are equivalent.*

Proof. k -OVH implies balanced k -OVH trivially. Conversely, we show that refuting k -OVH refutes the balanced k -OVH. To this end, assume that for some $0 \leq \gamma_1, \dots, \gamma_k \leq 1$ there exists an algorithm \mathcal{A} solving the k -OV with $|A_1| = n^{\gamma_1}, \dots, |A_k| = n^{\gamma_k}$ in time $\mathcal{O}(n^{(\sum_{i=1}^k \gamma_i) - \varepsilon})$.

Then given an instance of balanced k -OV with $|A_1| = \dots = |A_k| = n$, we can partition each of the sets A_i into $s := \lceil n^{1-\gamma_i} \rceil$ subsets A_i^1, \dots, A_i^s and run the algorithm \mathcal{A} on each combination of subsets and return true if for at least one instance the algorithm \mathcal{A} returns true and false otherwise. Clearly, this yields a correct algorithm for the balanced k -OV and it runs in time $\mathcal{O}(n^{1-\gamma_1} \dots n^{1-\gamma_k} \cdot n^{(\sum_{i=1}^k \gamma_i) - \varepsilon}) = \mathcal{O}(n^{k-\varepsilon})$. \square

3 Algorithms and Hardness of k -Dominating Set

In this section we provide our algorithms and conditional hardness results for the k -Dominating Set problem. We start by recalling the baseline algorithm (Proposition 1.1) in Section 3.1. Then, in Section 3.2 we develop our improved algorithm for the 2-Dominating Set problem in sparse graphs, and in Section 3.3 we strengthen the conditional lower bounds for k -Dominating Set to match our algorithms.

3.1 The Baseline Algorithm

For the baseline algorithm (and also for our later improvements) we rely on the following simple observation:

Observation 3.1. *Given a graph G , let $S = \{v_1, \dots, v_k\}$ be a dominating set of G . Then, there exists $1 \leq i \leq k$, such that $\deg(v_i) \geq \frac{n}{k} - 1$.*

We call any such vertex $v \in V(G)$ with $\deg(v) \geq \frac{n}{k} - 1$ a *high-degree* vertex. Observe that in any graph G with n vertices and m edges, there are at most $\mathcal{O}(\frac{m}{n})$ high-degree vertices (assuming k is a fixed constant). For the rest of this section, let G denote a graph with n vertices and $m = n^\gamma$ edges for some $1 \leq \gamma \leq 2$ and let $H \subseteq V$ denote the set of high-degree vertices.

Proposition 1.1 (k -Dominating Set Baseline). *Let $k \geq 2$ and $1 \leq \gamma \leq 2$. The k -Dominating Set on graphs with n nodes and $m = \Theta(n^\gamma)$ edges can be solved in time $n^{\omega+o(1)} + n^{\omega(\lceil \frac{k-1}{2} \rceil, 1, \lfloor \frac{k-1}{2} \rfloor + \gamma - 1) + o(1)}$. If $\omega = 2$, this running time becomes $n^{2+o(1)} + mn^{k-2+o(1)}$.*

⁵ $d = \log^2 n$ can be replaced by any $d = \omega(\log n)$.

Proof. Let $\mathcal{S} \subseteq \mathcal{P}(V)$ and denote by $B_{\mathcal{S}}$ the binary matrix whose rows are indexed by \mathcal{S} and whose columns are indexed by V and the entry $B_{\mathcal{S}}[S, v] = 1$ if and only if S dominates v (i.e., $v \in S$ or there exists $s \in S$, such that $\{s, v\} \in E$). Similarly, for $\mathcal{T} \subseteq \mathcal{P}(V)$ let $C_{\mathcal{T}}$ denote the matrix whose rows are indexed by V and whose columns are indexed by \mathcal{T} and the entry $C_{\mathcal{T}}[v, T] = 1$ if and only if T dominates v . Clearly, we have $(\overline{B_{\mathcal{S}}} \cdot \overline{C_{\mathcal{T}}})[S, T] = 0$ if and only if $S \cup T$ dominate G .

Let $H \subseteq V$ be the set of all high degree vertices and let \mathcal{S} and \mathcal{T} be defined as follows:

$$\mathcal{S} = \left(\binom{V}{\lceil \frac{k-1}{2} \rceil} \right), \mathcal{T} = \left\{ T \in \left(\binom{V}{\lfloor \frac{k-1}{2} \rfloor + 1} \right) \mid T \cap H \neq \emptyset \right\}.$$

Recall that any k -dominating set S^* contains at least one high-degree vertex (Observation 3.1). We can therefore partition S^* into two subsets $S^* = S \cup T$, where S has size $\lceil \frac{k-1}{2} \rceil$, T has size $\lfloor \frac{k-1}{2} \rfloor + 1$ and T contains the high-degree vertex. It follows that the graph contains a k -dominating set if and only if the matrix product $\overline{B_{\mathcal{S}}} \cdot \overline{C_{\mathcal{T}}}$ contains a zero entry.

It remains to analyze the running time of computing this matrix product. Observe that

$$|\mathcal{S}| = \mathcal{O}(n^{\lceil \frac{k-1}{2} \rceil}), |\mathcal{T}| = \mathcal{O}\left(\frac{m}{n} \cdot n^{\lfloor \frac{k-1}{2} \rfloor}\right) = \mathcal{O}(n^{\gamma-1+\lfloor \frac{k-1}{2} \rfloor}).$$

Thus, $B_{\mathcal{S}}$ is an $\mathcal{O}(n^{\lceil \frac{k-1}{2} \rceil}) \times n$ matrix and $C_{\mathcal{T}}$ is an $n \times \mathcal{O}(n^{\gamma-1+\lfloor \frac{k-1}{2} \rfloor})$ matrices, and computing their product takes time $\mathcal{O}(n^{\omega+o(1)} + n^{\omega(\lceil \frac{k-1}{2} \rceil, 1, \lfloor \frac{k-1}{2} \rfloor + \gamma - 1) + o(1)})$ as claimed. \square

3.2 A Faster Algorithm for 2-Dominating Set

In this section we design our improved algorithm for the 2-Dominating Set for sparse graphs. Specifically, our goal is to prove the following theorem:⁶

Theorem 1.3 (2-Dominating Set Algorithm). *There is a randomized algorithm solving 2-Dominating Set in time $m^{\omega/2+o(1)}$.*

Our strategy is to phrase our 2-Dominating Set algorithm as an algorithmic *reduction* to the following intermediate problem:

Definition 3.2 (Max-Entry Matrix Product). *Consider 0-1-matrices B of size $\mathcal{O}(\frac{m}{n}) \times n$ and C of size $n \times n$, where C contains at most m nonzeros. The Max-Entry Matrix Product problem is to decide whether there exist i, j such that*

$$(B \cdot C)[i, j] = \sum_k B[i, k] \cdot C[k, j] = \sum_k B[i, k].$$

We remark that, when viewing B and C as the bi-adjacency lists of a tripartite graph, the Max-Entry Matrix Product problem asks whether there is a pair i, j of outer vertices such that every edge from i can be extended to a 2-path to j . Another equivalent formulation is in terms of the *Subset Query* problem: Given families \mathcal{S}, \mathcal{T} of subsets of some universe $[U]$, the goal is to test whether there exist sets $S \in \mathcal{S}$ and $T \in \mathcal{T}$ such that $S \subseteq T$. The Max-Entry Matrix Product problem is exactly the special case of Subset Query where $U = n$, $|\mathcal{S}| = \mathcal{O}(\frac{m}{n})$ and $\sum_{T \in \mathcal{T}} |T| = \mathcal{O}(m)$. While the (unrestricted) Subset Query problem has been studied in previous works [21, 11], we decided to stick to the matrix version from Definition 3.2 which is more in line with our view on the k -dominating set problem as explained in the overview.

⁶By employing the state-of-the-art fast rectangular matrix multiplication techniques (e.g. [20]), a more fine-grained analysis of the algorithm reveals that we can achieve even slightly better running time in very sparse graphs.

The Max-Entry Matrix Product problem can naively be solved in time $\mathcal{O}(m \cdot \frac{m}{n})$ (enumerate an index i and a nonzero entry $C[k, j]$) and in time $\text{MM}(\frac{m}{n}, n, n)$ by fast matrix multiplication. As we will prove later, by a more elaborate algorithm the Max-Entry Matrix Product problem can be solved in time $m^{\omega/2+o(1)}$.

Before, the first step towards our algorithm is to reduce 2-Dominating Set on sparse graphs to Max-Entry Matrix Product. The reduction is somewhat similar to the baseline algorithm in the previous section.

Lemma 3.3 (2-Dominating Set to Max-Entry Matrix Product). *If the Max-Entry Matrix Product problem can be solved in time $T(n, m)$, then the 2-Dominating Set problem can be solved in time $\mathcal{O}(T(n, m))$.*

Proof. Let $G = (V, E)$ be a given 2-Dominating Set instance. Let A be the adjacency matrix of G , where we understand that $A[i, i] = 1$ for all i . Moreover, let $H \subseteq V$ denote the subset of high-degree vertices (with degree at least $\frac{n}{2} - 1$), and let $\overline{A_{H,V}}$ denote the complement of the adjacency matrix restricted to $H \times V$. We claim that $\{u, v\}$ is a 2-dominating set of G if and only if $(\overline{A_{H,V}} \cdot A)[u, v] = \text{deg}_{\overline{G}}(u)$.

Indeed, note that $(\overline{A_{H,V}} \cdot A)[u, v] = \sum_{w \in V} \overline{A_{H,V}}[u, w] \cdot A[w, v]$ counts the number of vertices w that are not dominated by u , but are dominated by v . Furthermore, $u \in H$ and $v \in V$ form a 2-dominating set if and only if every node w that is not dominated by u is dominated by v . Since there are precisely $\text{deg}_{\overline{G}}(u)$ nodes that are non-dominated by u , we conclude that u and v form a 2-dominating set if and only if $(\overline{A_{H,V}} \cdot A)[u, v] = \text{deg}_{\overline{G}}(u) = \sum_w \overline{A_{H,V}}[u, w]$.

Picking $B = \overline{A_{H,V}}$ (which has size $\mathcal{O}(\frac{m}{n}) \times n$) and $C = A$ (which has size $n \times n$ and is m -sparse), we have therefore successfully reduced to an instance of Max-Entry Matrix Product. Constructing these matrices runs in linear time and is therefore negligible. \square

We proceed to the core of our algorithm.

Lemma 3.4 (Max-Entry Matrix Product to Rectangular Matrix Multiplication). *There is a randomized algorithm solving Max-Entry Matrix Product in time*

$$\tilde{\mathcal{O}} \left(\max_{1 \leq d \leq n} \text{MM} \left(\frac{m}{n}, d, \min \left\{ n, \frac{m}{d} \right\} \right) \right).$$

Proof. For an index j , the degree $\sum_k C[k, j]$ denotes the number of nonzero entries $C[k, j]$. As a first step, we will split C column-wise into $L = \lfloor \log n \rfloor$ many submatrices C_0, \dots, C_L such that all degrees in C_ℓ are in the range $[2^\ell, 2^{\ell+1})$. Each such submatrix still has size at most $n \times n$, and contains at most m nonzero entries. Even better: Since each column in C_ℓ contains at least 2^ℓ nonzero entries, there can be at most $\frac{m}{2^\ell}$ columns in C_ℓ , and thus C_ℓ has size at most $n \times \min\{n, \frac{m}{2^\ell}\}$. For the rest of the proof, fix any $0 \leq \ell \leq L$; we solve the Max-Entry Matrix Product problem on (B, C_ℓ) (for simplicity we will often omit the subscript from C_ℓ).

Our goal is to reduce Max-Entry Matrix Product to a regular matrix product. Unfortunately, while the outer dimensions are comparably small ($\frac{m}{n}$ and $\min\{n, \frac{m}{2^\ell}\}$), the inner dimension can be as large as n . The key idea is to apply a Bloom-filter-like construction to also compress the inner dimension. To this end, sample a uniformly random hash function $h : [n] \rightarrow [2^{\ell+2}]$ and consider

the matrices B' (of size $\mathcal{O}(\frac{m}{n}) \times 2^{\ell+2}$) and C' (of size $2^{\ell+2} \times \min\{n, \frac{m}{2^\ell}\}$) defined by

$$B'[i, k'] = \bigvee_{\substack{k \in [n] \\ h(k)=k'}} B[i, k],$$

$$C'[k', j] = \bigvee_{\substack{k \in [n] \\ h(k)=k'}} C[k, j].$$

We claim that these matrices preserve solutions in the following sense:

Claim 3.5. *If $(B \cdot C)[i, j] = \sum_k B[i, k]$, then $(B' \cdot C')[i, j] = \sum_{k'} B'[i, k']$.*

Proof. Assume that $(B \cdot C)[i, j] = \sum_k B[i, k] \cdot C[k, j] = \sum_k B[i, k]$, and take any k' with $B'[i, k'] = 1$. By construction, there is some $k \in [n]$ with $h(k) = k'$ and $B[i, k] = 1$. Our initial assumption implies that also $C[k, j] = 1$. But then our construction assigns $C'[k', j] = 1$. Since k' was arbitrary, it follows that indeed $(B' \cdot C')[i, j] = \sum_{k'} B'[i, k'] \cdot C[k', j] = \sum_{k'} B[i, k']$. \square

Claim 3.6. *If $(B \cdot C)[i, j] < \sum_k B[i, k]$, then $(B' \cdot C')[i, j] < \sum_{k'} B'[i, k']$ with probability at least $\frac{1}{2}$.*

Proof. From the assumption that $(B \cdot C)[i, j] < \sum_k B[i, k]$, it follows that there is some $k^* \in [n]$ with $B[i, k^*] = 1$ and $C[k^*, j] = 0$. We claim that with probability at least $\frac{1}{2}$, there is no index $k \neq k^*$ such that $h(k) = h(k^*)$ and $C[k, j] = 1$. Indeed, for any fixed $k \neq k^*$ the event $h(k) = h(k^*)$ happens with probability at most $\frac{1}{2^{\ell+2}}$. Moreover, in the j -th row of C there are at most $2^{\ell+1}$ nonzero entries $C[k, j]$, thus by a union bound the error probability is bounded by $\frac{2^{\ell+1}}{2^{\ell+2}} = \frac{1}{2}$. \square

This suggests the following algorithm: Repeat, for $R = 100 \log n$ iterations, the construction of B' and C' (with fresh randomness). We solve the Max-Entry Matrix Product problem on (B', C') using fast matrix multiplication. If there is a pair (i, j) such that across all repetitions we have $(B' \cdot C')[i, j] = \sum_{k'} B'[i, k']$, then we report “yes”. Otherwise, we report “no”. By Claim 3.5, we will always correctly report “yes” instances. In a “no” instance, the probability that we mistakenly report “yes” due to some false positive (i, j) is bounded by $2^{-R} \leq n^{-100}$ by Claim 3.6. Taking a union bound over the at most n^2 pairs, the total error probability is at most n^{-98} .

Let us finally consider the running time. For a fixed ℓ , we compute $\mathcal{O}(\log n)$ matrix products of size $\mathcal{O}(\frac{m}{n}) \times 2^{\ell+1} \times \min\{n, \frac{m}{2^\ell}\}$. This takes time $\tilde{\mathcal{O}}(\text{MM}(\frac{m}{n}, 2^\ell, \min\{n, \frac{m}{2^\ell}\}))$. It takes linear time to construct the matrices B' and C' , respectively, and this overhead is negligible in the running time. Summing over the $\log n$ levels $0 \leq \ell \leq L = \lceil \log n \rceil$, the total time is

$$\sum_{\ell=0}^{\lceil \log n \rceil} \tilde{\mathcal{O}}\left(\text{MM}\left(\frac{m}{n}, 2^\ell, \min\left\{n, \frac{m}{2^\ell}\right\}\right)\right) = \max_{1 \leq d \leq n} \tilde{\mathcal{O}}\left(\text{MM}\left(\frac{m}{n}, d, \min\left\{n, \frac{m}{d}\right\}\right)\right),$$

which is as claimed. \square

This completes the description of the Max-Entry Matrix Product algorithm; however, it remains to carefully analyze the complicated running time expression:

Corollary 3.7 (Max-Entry Matrix Product). *There is a randomized algorithm solving Max-Entry Matrix Product in time $m^{\omega/2+o(1)}$.*

Proof. We use the algorithm from Lemma 3.4. It runs in time $\tilde{\mathcal{O}}(\max_{1 \leq d \leq n} T(n, m, d))$ where

$$T(n, m, d) = \text{MM}\left(\frac{m}{n}, d, \min\left\{n, \frac{m}{d}\right\}\right),$$

and it remains to bound $T(n, m, d) \leq m^{\frac{\omega}{2}+o(1)}$ for any choice of $1 \leq d \leq n$. Throughout this proof, we will only use that $\text{MM}(N, N, N) \leq N^{\omega+o(1)}$ (by fast square matrix multiplication) and that trivially $\text{MM}(N_1 \cdot N'_1, N_2, N_3) = \mathcal{O}(N'_1) \cdot \text{MM}(N_1, N_2, N_3)$ (similarly for N_2 and N_3). We distinguish the following three cases, each of which can be proved by a calculation:

- *Case 1:* $n \leq \frac{m}{d}$:

$$\begin{aligned} T(n, m, d) &\leq \text{MM}\left(\frac{m}{n}, \frac{m}{n}, n\right) && (d \leq \frac{m}{n}) \\ &\leq \mathcal{O}\left(\frac{n^2}{m}\right) \cdot \text{MM}\left(\frac{m}{n}, \frac{m}{n}, \frac{m}{n}\right) && (n^2 \geq m) \\ &\leq \mathcal{O}\left(\frac{n^2}{m}\right) \cdot \left(\frac{m}{n}\right)^{\omega+o(1)} \\ &\leq \frac{m^{\omega-1+o(1)}}{n^{\omega-2}} \\ &\leq m^{\omega/2+o(1)}. && (n^2 \geq m) \end{aligned}$$

- *Case 2:* $n \geq \frac{m}{d}$ and $d \leq \frac{m}{d}$:

$$\begin{aligned} T(n, m, d) &\leq \text{MM}\left(d, d, \frac{m}{d}\right) && (n \geq \frac{m}{d}) \\ &\leq \mathcal{O}\left(\frac{m}{d^2}\right) \cdot \text{MM}\left(d, d, d\right) && (d \leq \frac{m}{d}) \\ &\leq \mathcal{O}\left(\frac{m}{d^2}\right) \cdot d^{\omega+o(1)} \\ &\leq m \cdot d^{\omega-2+o(1)} \\ &\leq m \cdot \sqrt{m}^{\omega-2+o(1)} && (d \leq \frac{m}{d}) \\ &\leq m^{\omega/2+o(1)}. \end{aligned}$$

- *Case 3:* $n \geq \frac{m}{d}$ and $d \geq \frac{m}{d}$:

$$\begin{aligned} T(n, m, d) &\leq \text{MM}\left(\frac{m}{d}, d, \frac{m}{d}\right) && (d \leq n, n \geq \frac{m}{d}) \\ &\leq \mathcal{O}\left(\frac{d^2}{m}\right) \cdot \text{MM}\left(\frac{m}{d}, \frac{m}{d}, \frac{m}{d}\right) && (d \geq \frac{m}{d}) \\ &\leq \mathcal{O}\left(\frac{d^2}{m}\right) \cdot \left(\frac{m}{d}\right)^{\omega+o(1)} \\ &\leq \frac{m^{\omega-1+o(1)}}{d^{\omega-2}} \\ &\leq \frac{m^{\omega-1+o(1)}}{\sqrt{m}^{\omega-2}} && (d \geq \frac{m}{d}) \\ &\leq m^{\omega/2+o(1)}. \end{aligned}$$

In all three cases, we have successfully bounded the running time by $m^{\omega/2+o(1)}$. This completes the proof. \square

We have assembled all parts of the proof of Theorem 1.3: We combine the reduction from Lemma 3.3 with the efficient Max-Entry Matrix Product algorithm from Corollary 3.7. It follows that the 2-Dominating Set problem is in time $m^{\omega/2+o(1)}$.

Recall that the baseline algorithm (Proposition 1.1) in combination with our 2-Dominating Set algorithm (Theorem 1.3) prove that for $\omega = 2$, the k -Dominating Set problem can be solved in time $mn^{k-2+o(1)}$ for all $k \geq 2$. In the next section, we complement this upper bound by a matching conditional lower bound.

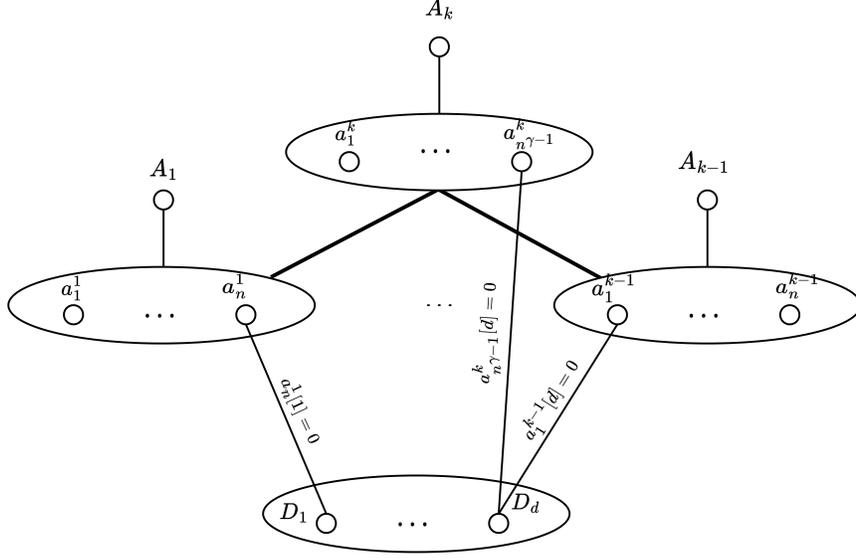


Figure 1: Schematic representation of the graph G as constructed in proof of Theorem 1.2. Bold edges represent bicliques between the corresponding sets.

3.3 A Matching Conditional Lower Bound

In this section our goal is to prove the following theorem:

Theorem 1.2 (k -Dominating Set Lower Bound). *For all $k \geq 3$ and $\epsilon > 0$, there is no algorithm for k -Dominating Set in time $\mathcal{O}(mn^{k-2-\epsilon})$, unless the k -OV Hypothesis fails.*

The idea is to adapt the reduction by Pătraşcu and Williams [32] to create a sparse instance. In particular, we reduce from the unbalanced variant of k -OV problem obtained by setting $|A_1| = \dots = |A_{k-1}| = n$ and $|A_k| = n^{\gamma-1}$ for the suitable $1 \leq \gamma \leq 2$, and then we use the vertices corresponding to vectors in A_k to dominate the remaining sets.

Proof. To prove this lower bound, we reduce from k -Orthogonal Vectors problem.

Let A_1, \dots, A_{k-1} denote sets of d -dimensional (assume $d = \mathcal{O}(\log^2 n)$) binary vectors of size n and A_k be a set of d -dimensional binary vectors of size $n^{\gamma-1}$ for $1 \leq \gamma \leq 2$. We construct a graph G with $\tilde{\mathcal{O}}(n)$ vertices and $\tilde{\mathcal{O}}(n^\gamma)$ edges that has dominating set of size k if and only if we can find vectors $a_1 \in A_1, \dots, a_k \in A_k$ such that $\prod_{i=1}^k a_i[j] = 0$ for every $j = 1, \dots, d$.

Consider the following construction illustrated in Figure 1. Let $V(G)$ consist of the k vertices corresponding to sets A_1, \dots, A_k , the $\mathcal{O}(n)$ vertices corresponding to the vectors, labeled $a_j^i \in A_i$, and the vertices labeled D_1, \dots, D_d , corresponding to the d dimensions. For each $1 \leq i \leq k$ add all edges between A_i and a_j^i for all j . For all $1 \leq i \leq k-1$ add an edge between every vertex corresponding to a vector contained in A_i (namely all the vertices a_j^i) and all the vertices corresponding to vectors in A_k . Add an edge between the vertices a_j^i and D_t if and only if $a_j^i[t] = 0$ for all i, j, t .

The graph G clearly has $\mathcal{O}(n + d) = \mathcal{O}(n)$ vertices and observe that there are $\mathcal{O}(n^{\gamma-1} + d) = \mathcal{O}(n^{\gamma-1})$ vertices with degree at most $\mathcal{O}(n)$ (in particular the vertices corresponding to vectors in A_i^k for all $i \in [n^{\gamma-1}]$, vertices D_1, \dots, D_d and the vertices A_i for all $i \in [k]$). All other vertices have degree at most $n^{\gamma-1}$, hence the total number of edges is at most $\mathcal{O}(n \cdot n^{\gamma-1} + dn) = \tilde{\mathcal{O}}(n^\gamma)$ and consequently, the graph G can be constructed in time $\mathcal{O}(n^\gamma d) = \tilde{\mathcal{O}}(n^\gamma)$.

It is now sufficient to argue that G has a dominating set of size k if and only if there exist vectors $a_1 \in A_1, \dots, a_k \in A_k$ such that $a_1[i] \cdot \dots \cdot a_k[i] = 0$ for every $i \in [d]$. To this end, suppose first that such vectors exist. Then we show that the vertices corresponding to a_1, \dots, a_k form a dominating set.

First observe that the vertex corresponding to a_i is adjacent to the vertex A_i , and hence the vertices A_1, \dots, A_k are dominated. Furthermore, since all vertices a_j^i (for $1 \leq i < k$) are adjacent to all the vertices corresponding to vectors in A_k , clearly all such vertices are dominated by a_k and symmetrically all vertices a_j^k are dominated by e.g. a_1 .

It is only left to verify that the vertices D_1, \dots, D_d are dominated. Consider arbitrary D_i . Now, since $a_1[i] \cdot \dots \cdot a_k[i] = 0$, there exists some $1 \leq j \leq k$, such that $a_j[i] = 0$ and consequently, by construction of G , a_j is adjacent to D_i .

Conversely, suppose that G assumes a dominating set $S = \{s_1, \dots, s_k\}$ of size k . Consider first the vertices A_1, \dots, A_k . Clearly, their closed neighbourhoods are disjoint, hence each vertex s_i from S is contained in $N[A_i]$ (without loss of generality). Furthermore, since $\{D_1, \dots, D_d\} \cap S = \emptyset$, and the vertices D_t are non-adjacent to the vertices A_1, \dots, A_k , there exists at least one $i \in [k]$, such that $s_i = a_j^i$ for some j .

We construct the solution of k orthogonal vectors as follows. For each s_i such that $s_i = A_i$, let a_i be an arbitrary vector from A_i . For the remaining vertices let a_i be the vector corresponding to s_i . We claim that $a_1 \in A_1, \dots, a_k \in A_k$ satisfy $a_1[i] \cdot \dots \cdot a_k[i] = 0$ for all $i \in [k]$. To see this, consider an arbitrary i . Observe that D_i is dominated by some s_j , and in particular $s_j \neq A_j$. Hence $a_j[i] = 0$ and the claim follows. \square

4 Hardness of Monochromatic Basic Problems

In this section, we will prove hardness of all monochromatic basic problems except k -Dominating Set in sparse graphs. We proceed in three steps: (1) First, we recall hardness established in [21] for all multichromatic basic problems, which already holds in sparse graphs. (2) We then reduce each multichromatic basic problem to its bichromatic version (where one of the parts maintains a size of $\tilde{O}(1)$). (3) Finally, we reduce each such bichromatic problem except bichromatic k -Dominating Set (i.e., k -Set Cover) to its monochromatic version.

For completeness, we begin with a proof that all Multichromatic Basic Problems are k -OV hard even in sparse graphs.

Proposition 4.1 (Implicit in [21]). *Let Φ be a basic property of order k . Given an instance $A_1, \dots, A_k \subseteq \{0, 1\}^d$ of k -OV, we can construct in time $\tilde{O}(n)$ an equivalent instance $G = (V_1, \dots, V_k, W, E)$ of the Multichromatic Basic Problem for Φ with at most $\tilde{O}(n)$ edges and $|W| = \tilde{O}(1)$.*

Proof. Without loss of generality, assume that $\Phi = E(v_1, w) \vee \dots \vee E(v_\ell, w) \vee \overline{E(v_{\ell+1}, w)} \vee \dots \vee \overline{E(v_k, w)}$. We call a vertex v_i *positive* (resp. *negative*) if Φ contains the literal $E(v_i, w)$ (resp. $\overline{E(v_i, w)}$). For each dimension i , add a vertex w_i to W and let $V_i = A_i$ for all $i \in [k]$. Now for every vertex $v \in V_1, \dots, V_\ell$, add an edge between v and w_i if and only if the vector corresponding to v has i -th entry equal to zero. Similarly, for every vertex $v \in V_{\ell+1}, \dots, V_k$, add an edge between v and w_i if and only if the vector corresponding to v has i -th entry equal to one.

If A_1, \dots, A_k is a yes-instance of k -OV, then we can find vectors $a_1 \in A_1, \dots, a_k \in A_k$ such that for any dimension j , some vector a_i satisfies $a_i[j] = 0$. We proceed to show that choosing $v_1 \in V_1, \dots, v_k \in V_k$ such that v_i is the vertex corresponding to a_i satisfies Φ . If $1 \leq i \leq \ell$, then Φ contains the literal $E(v_i, w)$ and the corresponding vertex v_i is by construction adjacent to w_j

satisfying Φ . Otherwise, Φ contains the literal $\overline{E(v_i, w)}$ and the corresponding vertex from v_i is non-adjacent to w_j , again satisfying Φ .

On the other hand, if A_1, \dots, A_k is a no-instance of k -OV, then for every selection of vectors $a_1 \in A_1, \dots, a_k \in A_k$ there exists a dimension j , such that $a_i[j] = 1$. In particular, if we choose any $v_1 \in V_1, \dots, v_k \in V_k$, then considering the corresponding vectors yields a vertex $w_j \in W$ that is non-adjacent to all v_1, \dots, v_ℓ and adjacent to all $v_{\ell+1}, \dots, v_k$, thus leaving $\Phi(v_1, \dots, v_k, w_j)$ unsatisfied.

Note that $|W| = d = \tilde{O}(1)$. Since every edge in G connects $v \in V_i$ with $w \in W$, the total number of edges is at most $\mathcal{O}((\sum_{i=1}^k |V_i|) \cdot |W|) = \mathcal{O}(nd) = \tilde{O}(n)$. \square

4.1 Multichromatic to Bichromatic

In order to prove the hardness of Monochromatic Basic Problems with at least one negative literal in the sparse graphs, we introduce the intermediate class of *Bichromatic Basic Problems* defined as follows. Let k be a fixed constant and for every $i \in [k]$, let $L_i : X \times Y \rightarrow \{0, 1\}$ be either defined as $L_i(x, y) = E(x, y)$, or $L_i(x, y) = \overline{E(x, y)}$. Now all the Bichromatic Basic Problems can be stated in the following way. Given a bipartite graph $G = (X, Y, E)$ decide if there is a set of pairwise distinct vertices $x_1, \dots, x_k \in X$, such that for all $y \in Y$ it holds that $L_1(x_1, y) \vee \dots \vee L_k(x_k, y)$?

We proceed to prove that all the Bichromatic Basic Problems are k -OV hard already in sparse graphs, by reducing from the corresponding Multichromatic Basic Problems. To this end, for any subset $S \subseteq X$, we first show how to construct an *identifier gadget* J_S , such that if we add corresponding edges between S and J_S then for every $x_1, \dots, x_k \in S$ there exists $y \in J_S$ such that $\phi(x_1, \dots, x_k, y)$ is not satisfied. In order to do this, we first need to introduce some tools.

Given a set S with n elements, let $\phi : S \rightarrow \{0, 1\}^{\log(n)}$ be an injective function called the *identifier* of S . Let x be a binary vector with d dimensions and define

$$f(x) = \left\{ (i_1^{(1)}, \dots, i_k^{(1)}, \dots, i_1^{(k)}, \dots, i_k^{(k)}), (b_1^{(1)}, \dots, b_k^{(1)}, \dots, b_1^{(k)}, \dots, b_k^{(k)}) \mid \bigwedge_{s=1}^k \left(\bigvee_{t=1}^k x[i_t^{(s)}] = b_t^{(s)} \right) \right\}$$

Let S be a set of n elements and let

$$T = \left\{ (x, I, B) \mid x \in S, (I, B) \in f(\phi(x)) \right\}$$

For the rest of this section, let S, T, f, ϕ be as above. The following lemma shows that for any subset S' consisting of at most $2k$ elements from S , by labeling at most k distinct elements in S' *positive* and at most k distinct elements in S' *negative* (such that no element is labeled both positive and negative), we can always find the the tuple of corresponding indices I and bits B , such that (x, I, B) is in T for each positive element x and (y, I, B) is not in T for any negative element y . This will be a key observation allowing us to build the gadget that has the desired properties.

Lemma 4.2. *For every subset $\{x_1, \dots, x_k, y_1, \dots, y_k\} \subseteq S$, there exist indices I , and bits B such that $(x_i, I, B) \in T$ and $(y_i, I, B) \notin T$ for all i .*

Proof. Intuitively, for any fixed y_s , we want to find indices $i_1^{(s)}, \dots, i_k^{(s)}$ and bits $b_1^{(s)}, \dots, b_k^{(s)}$ that leave the clause $\bigvee_{t=1}^k \phi(y_s)[i_t^{(s)}] = b_t^{(s)}$ unsatisfied, and observe that this is sufficient to show that for arbitrary selection of the remaining indices and bits, (y_s, I, B) is not in T . Simultaneously, we want to make sure that for every x_t the clause $\bigvee_{t=1}^k \phi(x_t)[i_t^{(s)}] = b_t^{(s)}$ is satisfied, by assuring that $\phi(x_t)[i_t^{(s)}] = b_t^{(s)}$ for every t . To this end, we exploit the injectiveness of ϕ . In particular, since ϕ is

injective, we can find an index $i_1^{(1)}$, such that $\phi(x_1)[i_1^{(1)}] \neq \phi(y_1)[i_1^{(1)}]$. Similarly, for all $2 \leq j \leq k$, find indices $i_2^{(1)}, \dots, i_k^{(1)}$, such that $\phi(x_j)[i_j^{(1)}] \neq \phi(y_1)[i_j^{(1)}]$.

Let $b_t^{(1)} = \phi(x_t)[i_t^{(1)}]$ for all $1 \leq t \leq k$ and observe that for all $t \in [k]$, $\phi(y_1)[i_t^{(1)}] \neq b_t^{(1)}$, and hence the formula $\bigvee_{t=1}^k \phi(y_1)[i_t^{(1)}] = b_t^{(1)}$ is not satisfied. On the other hand, clearly for any j , the formula $\bigvee_{t=1}^k \phi(x_j)[i_t^{(1)}] = b_t^{(1)}$ is satisfied since $\phi(x_j)[i_j^{(1)}] = b_j^{(1)}$. Similarly, for all $2 \leq s \leq k$, one can find indices $i_1^{(s)}, \dots, i_k^{(s)}$, such that $\phi(x_j)[i_j^{(s)}] \neq \phi(y_s)[i_j^{(s)}]$ and symmetrically setting $b_t^{(s)} = \phi(x_t)[i_t^{(s)}]$ leaves the formula $\bigvee_{t=1}^k \phi(y_s)[i_t^{(s)}] = b_t^{(s)}$ unsatisfied, while satisfying $\bigvee_{t=1}^k \phi(x_j)[i_t^{(s)}] = b_t^{(s)}$.

Finally, we observe that consequently the formula $\bigwedge_{s=1}^k (\bigvee_{t=1}^k \phi(x_j)[i_t^{(s)}] = b_t^{(s)})$ is satisfied for every j and $\bigwedge_{s=1}^k (\bigvee_{t=1}^k \phi(y_j)[i_t^{(s)}] = b_t^{(s)})$ is not satisfied for any j . The desired result follows. \square

We proceed to use the result from the previous lemma to construct our gadget. In particular, the vertices in our gadget will correspond to the indices and the bits as above and we will add an edge between a vertex x from the set $S \subseteq X$ and the vertex (I, B) in the gadget if and only if (x, I, B) is in T . By the previous lemma, for any selection of at most k positive and at most k negative vertices, we can find a vertex in the gadget that is adjacent to all negative vertices and non-adjacent to all positive vertices, thus appending the gadget vertices to the set Y implies that no complete solution is contained inside S .

Lemma 4.3 (Identifier gadget). *Let $G = (X, Y, E)$ be a bipartite graph and k a fixed constant. If $S \subseteq X$, then one can add $\tilde{\mathcal{O}}(1)$ vertices to Y and connect those vertices to S so that for any k vertices $x_1, \dots, x_k \in S$ and for any k vertices $x'_1, \dots, x'_k \in X$ there exists a vertex $y \in Y$ such that $y \in N(x_1) \cap N(x_2) \cap \dots \cap N(x_k)$ and $y \notin N(x'_1) \cup \dots \cup N(x'_k)$.*

Proof. Let $I = \lceil \log(|S|) \rceil^{k^2}$ and let $B = \{0, 1\}^{k^2}$. Now define $J_S := I \times B$. Observe that $|J_S| = \mathcal{O}(2^{k^2} \log^{k^2} n) = \tilde{\mathcal{O}}(1)$.

Let $Y' = Y \cup J_S$ and add edge between $x \in X$ and $(I, B) \in J_S$ if and only if $(x, I, B) \in T$. Now fix any tuple of vertices $x_1, \dots, x_k \in S$. By the previous lemma, for any $x'_1, \dots, x'_k \in S$, we can find a vertex $y \in J_S$ corresponding to some tuple of indices I and bits B such that $y \in N(x_i)$ and $y \notin N(x'_i)$ for all i . Observe that since y is adjacent only to the vertices in S , by choosing a vertex $x'_i \in X \setminus S$, this property remains preserved. \square

Given a bipartite graph $G = (X, Y, E)$ and a subset S , the set of $\tilde{\mathcal{O}}(1)$ vertices as described in the last lemma is called *identifier gadget* and is denoted by J_S . Constructing a graph $G' = (X, Y \cup J_S, E)$ from G by adding the edges as described above will be referred to as *attaching* the identifier gadget to S . If the identifier J_S is attached to S , we schematically represent this by adding a directed edge from S to J_S .

We proceed to give the reduction from any Multichromatic Basic Problem to the corresponding Bichromatic Basic Problem and then using the result from the beginning of the section, we conclude that all the Bichromatic Basic Problems are hard under SETH.

Proposition 4.4. *Let Φ be any basic property. Given an instance $G = (V_1, \dots, V_k, W)$ of the Multichromatic Basic Problem corresponding to Φ with $\mathcal{O}(n)$ vertices with $|W| = \tilde{\mathcal{O}}(1)$, we can construct in time $\tilde{\mathcal{O}}(n)$ an equivalent instance $G' = (X, Y, E)$ of the corresponding Bichromatic Basic Problem with $|Y| = \tilde{\mathcal{O}}(|W|)$.*

Proof. Without loss of generality, assume that $\Phi = E(v_1, w) \vee \dots \vee E(v_\ell, w) \vee \overline{E(v_{\ell+1}, w)} \vee \dots \vee \overline{E(v_k, w)}$ and let $\ell' := k - \ell$ denote the number of negative literals in Φ . Consider the following construction of the graph $G' = (X, Y, E)$. For all $S' \subseteq \{V_1, \dots, V_k\}$ of size $k - 1$, let $S = \bigcup_{s \in S'} s$

and attach an identifier gadget J_S to S . If for some $1 \leq i \leq \ell$, $V_i \notin S'$, add all edges between V_i and J_S . Let $X = V_1 \cup \dots \cup V_k$ and let Y consist of W and all the gadgets J_S as constructed above. This reduction is depicted schematically on Figure 2.

Claim 4.5. *If there are vertices $x_1, \dots, x_k \in X$ such that $N(x_{\ell+1}) \cap \dots \cap N(x_k) \subseteq N(x_1) \cup \dots \cup N(x_\ell)$, then each of the sets V_1, \dots, V_k contains exactly one of the vertices x_1, \dots, x_k .*

Proof. Assume that for some $\ell + 1 \leq i \leq k$ the set V_i contains none of the vertices x_1, \dots, x_k . Observe that the identifier gadget $J_{X \setminus V_i}$ is attached to the whole set X except V_i , hence contradicting Lemma 4.3.

Assume now that for some $1 \leq i \leq \ell$, V_i contains none of the vertices x_1, \dots, x_k . By Lemma 4.3, there exists a vertex $v \in J_{X \setminus V_i}$ adjacent to all $x_{\ell+1}, \dots, x_k$ and to no other vertex in $X \setminus V_i$. In particular, it is non-adjacent to x_j for any $j \in [\ell]$, yielding a contradiction. Hence, each set V_i contains at least one vertex and since we are placing k vertices in k sets, the upper bound is also trivially satisfied. \square

Claim 4.6. *If there are vertices $x_1, \dots, x_k \in X$ such that $N(x_{\ell+1}) \cap \dots \cap N(x_k) \subseteq N(x_1) \cup \dots \cup N(x_\ell)$, then for every $1 \leq i \leq \ell$, there is a $1 \leq j \leq \ell$ such that $x_i \in V_j$.*

Proof. Assume for contradiction that there exists some index i such that $x_i \in X \setminus \bigcup_{1 \leq j \leq \ell} V_j$. Then by the pigeonhole principle, there exists a set V_j (for $1 \leq j \leq \ell$) that does not contain any of the vertices x_1, \dots, x_ℓ and by the previous claim, V_j contains x_k (without loss of generality).

Consider the identifier gadget $J_{X \setminus V_j}$. By construction, every vertex in this gadget is adjacent to x_k . Hence, by the assumption that $N(x_{\ell+1}) \cap \dots \cap N(x_k) \subseteq N(x_1) \cup \dots \cup N(x_\ell)$, it must also hold that $N(x_{\ell+1}) \cap \dots \cap N(x_{k-1}) \cap J_{X \setminus V_j} \subseteq N(x_1) \cup \dots \cup N(x_\ell)$, thus contradicting Lemma 4.3. \square

Now assume V_1, \dots, V_k, W is a yes-instance of the multichromatic basic problem. Namely, we can find $v_1 \in V_1, \dots, v_k \in V_k$, such that for every $w \in W$, the formula $E(v_1, w) \vee \dots \vee E(v_\ell, w) \vee \overline{E(v_{\ell+1}, w)} \vee \dots \vee \overline{E(v_k, w)}$ is satisfied. We claim that in G' , for every $y \in Y$, the formula $E(v_1, y) \vee \dots \vee E(v_\ell, y) \vee \overline{E(v_{\ell+1}, y)} \vee \dots \vee \overline{E(v_k, y)}$ is satisfied. Let $v \in Y$ be arbitrary. If $v \in W$, then it is trivially satisfied. Consider now $v \in Y \setminus W$. Clearly $v \in J_{X \setminus V_i}$ for some i . If $1 \leq i \leq \ell$, then by construction, all edges between V_i and $J_{X \setminus V_i}$ are present and in particular v is adjacent to the positive vertex v_i . On the other hand if $\ell + 1 \leq i \leq k$, then there are no edges between V_i and $J_{X \setminus V_i}$ and in particular v is nonadjacent to the negative vertex v_i . In both cases, v is satisfied.

Conversely, assume that there are vertices $x_1, \dots, x_k \in X$ such that for every y in Y the formula $E(x_1, y) \vee \dots \vee E(x_\ell, y) \vee \overline{E(x_{\ell+1}, y)} \vee \dots \vee \overline{E(x_k, y)}$ is satisfied. By the two claims above (without loss of generality), $x_1 \in V_1, \dots, x_k \in V_k$. And now it is sufficient to observe that since $W \subseteq Y$ and every vertex in Y is satisfied by x_1, \dots, x_k , every vertex in W is also trivially satisfied by x_1, \dots, x_k . \square

The following proposition now follows directly.

Proposition 4.7. *Let Φ be a basic property with at least one negative literal. For any $\varepsilon > 0$, the Bichromatic Basic Problem corresponding to Φ cannot be solved faster than $\mathcal{O}(n^{k-\varepsilon})$ even on graphs with $\tilde{\mathcal{O}}(n)$ edges, unless SETH fails.*

4.2 Bichromatic to Monochromatic

We now proceed to prove the hardness of Monochromatic Basic Problems in sparse graphs with at least one negative literal in the corresponding basic property by reducing from the corresponding Bichromatic Problem.

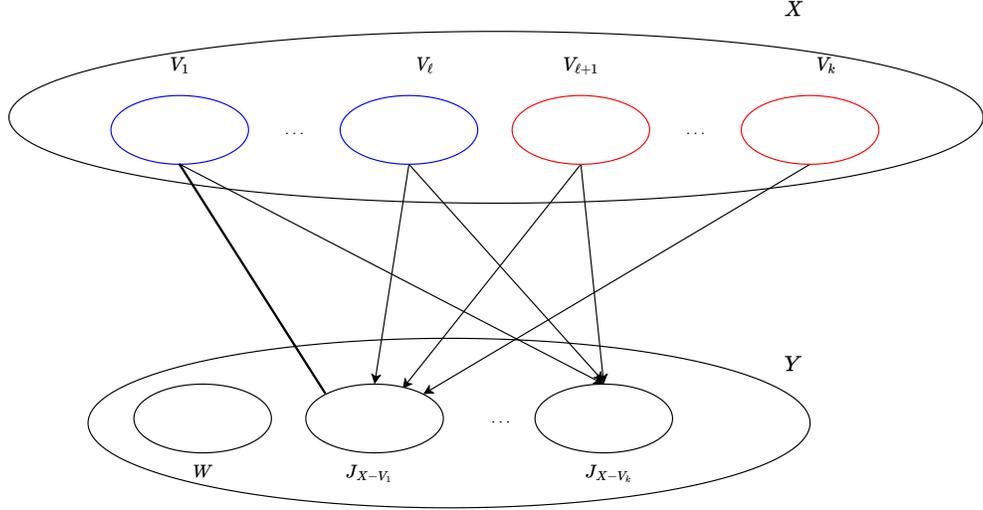


Figure 2: Schematical representation of the reduction from Multichromatic Basic Problem to Bichromatic Basic Problem. Blue sets contain positive vertices, red sets contain negative vertices, bold edges represent bicliques between the corresponding sets and directed edge from set V_i to the identifier gadget J_S represents attaching the gadget to the set. The edges from V_i to W are inherited from the multichromatic instance.

Lemma 4.8. *Given any positive integer N and a constant k , there exists a graph G with at most $\tilde{O}(N)$ vertices satisfying the following conditions.*

- *Every vertex v is labelled by an integer $\lambda(v) \in \{1, \dots, N\}$.*
- *For all $a_1, \dots, a_k, b_1, \dots, b_k \in V(G)$ and for every $i \in 1, \dots, N$, there exists a vertex $x \in V(G) \setminus \{a_1, \dots, a_k, b_1, \dots, b_k\}$ such that $\lambda(x) = i$ and x is adjacent to all a_j and nonadjacent to all b_j for $j \in [k]$.*
- *We can compute G in deterministic polynomial time in N .*

Before proving this lemma, we first show how one can use this result to obtain a reduction from a sparse Bichromatic Basic Problem to the corresponding Monochromatic Basic Problem.

Proposition 4.9. *Let Φ be a basic property with k variables and at least one negative literal. Given an instance $G = (X, Y, E)$ of the Bichromatic Basic Problem corresponding to Φ with n vertices and assume $Y = \tilde{O}(1)$, we can construct an equivalent instance $G' = (V, E)$ of the corresponding Monochromatic Basic Problem in time $\tilde{O}(n)$ consisting of at most $\tilde{O}(n)$ edges.*

Proof. Without loss of generality, assume that $\Phi = E(x_1, y) \vee \dots \vee E(x_\ell, y) \vee \overline{E(x_{\ell+1}, y)} \vee \dots \vee \overline{E(x_k, y)}$, for some $\ell < k$, and let $k - \ell = \ell' \geq 1$. Consider the graph G' constructed as follows.

Let $N = |Y|$ and let H be a graph satisfying the properties of Lemma 4.8. Without loss of generality, assume that $Y = \{y_1, \dots, y_N\}$ and label the vertex y_i by i . Let $V(G') = V(H) \cup X$. Let $H \cong G[V(H)]$, and add an edge between $x \in X$ and $h \in H$ if and only if $\lambda(h) = i$ and there is an edge between x and a vertex in Y labeled i in G .

First assume G is a yes-instance of the Bichromatic Basic Problem. Namely, assume that we can find $x_1, \dots, x_k \in X$ such that for every $y \in Y$, the formula $E(x_1, y) \vee \dots \vee E(x_\ell, y) \vee \overline{E(x_{\ell+1}, y)} \vee \dots \vee \overline{E(x_k, y)}$ is satisfied. We claim that also $\forall v \in V(G')$, it holds that $E(x_1, v) \vee \dots \vee E(x_\ell, v) \vee$

$\overline{E(x_{\ell+1}, v)} \vee \dots \vee \overline{E(x_k, v)}$. To see this assume first that $v \in X$. Since there are no edges inside X , clearly each vertex is nonadjacent to e.g. $x_{\ell+1}$ hence satisfying the desired formula.

Now assume that $v \in V(H)$. Then $\lambda(v) = i$ for some $1 \leq i \leq N$. Consider the vertex y in $Y \subseteq V(G)$ that is labelled i . Since G is a yes-instance, this vertex is either adjacent to x_j for some $j \in [\ell]$, or nonadjacent to x_j for some $\ell + 1 \leq j \leq k$. Now, since v and y have the same label in the respective graph, by construction of G' , if $E(x_j, y)$, then also $E(x_j, v)$ and similarly if $\overline{E(x_j, y)}$, then $\overline{E(x_j, v)}$. Therefore the desired formula is satisfied for v as well. We have thus covered all the vertices from $V(G')$.

On the other hand, assume G is a no-instance of the Bichromatic Basic Problem. In particular, assume that for every selection of x_1, \dots, x_k , there exists $y \in Y$ that is non-adjacent to all x_1, \dots, x_ℓ and adjacent to all $x_{\ell+1}, \dots, x_k$. Consider any collection of vertices $v_1, \dots, v_k \in V(G')$. Let S_X be the subset of v_1, \dots, v_k consisting of all vertices v_j contained in X . Clearly, we can find a vertex $y_i \in Y$ that is not satisfied by S_X . By construction of G' it follows that any vertex $w \in V(H)$ with $\lambda(w) = i$ is not satisfied by S_X .

Moreover, by Lemma 4.8, there exists a vertex $w \in V(H)$ with $\lambda(w) = i$ such that w is not satisfied by any vertex v_i in $\{v_1, \dots, v_k\} \setminus S_X$. In particular, it follows that such w is not satisfied by v_1, \dots, v_k and since the vertices v_1, \dots, v_k were selected arbitrarily, G' is also a no-instance. \square

We now proceed to prove the Lemma 4.8. But before that, we state a few basic well known facts about polynomials that will be useful.

Lemma 4.10. *For any two polynomials f, g with coefficients in a field \mathbb{F} of degree at most d , there are at most d points x in \mathbb{F} such that $f(x) = g(x)$.*

Lemma 4.11. *Given $d + 1$ distinct points $x_1, \dots, x_{d+1} \in \mathbb{F}$ and $d + 1$ values $y_1, \dots, y_{d+1} \in \mathbb{F}$, there is a unique polynomial $f \in \mathbb{F}[X]_{\leq d}$ satisfying $f(x_i) = y_i$ for all i .*

Proof (of Lemma 4.8). Let p be the smallest prime number strictly larger than $k^2(\log(N) + k + 2)$ and let \mathbb{F}_p denote the finite field of order p . Let $d = \lceil \log_p(N) \rceil + k$. Let $V(G) = \mathbb{F}_p[X]_{\leq d}$, that is each vertex in G corresponds to a polynomial with coefficients in \mathbb{F}_p of degree at most d . Let $L = \lceil \log_p(N) \rceil$ and for $f \in V(G)$ let $\lambda(f) = i$ if and only if $\psi(f(1), \dots, f(L)) = i$ be the labeling of the vertices in G for a bijective function $\psi : \mathbb{F}_p^L \rightarrow [p^L]$ (if $\psi(f(1), \dots, f(L)) > N$, then we relabel f to N). We add the edges to our graph as follows. Let $R = L + dk + k + 1$ and for any $f, g \in V(G)$, let $\{f, g\} \in E(G)$ if and only if there exists an x such that $L < x \leq R$ and $f(x) = g(x)$. We observe that $p > R$, hence the elements $1, \dots, R$ are all distinct in \mathbb{F}_p .

We proceed to show that given any distinct polynomials $a_1, \dots, a_k, b_1, \dots, b_k$ and any label i , we can find a polynomial f with $\lambda(f) = i$ such that for all $j \in [k]$ there exists $L + 1 \leq x \leq R$ satisfying $a_j(x) = f(x)$, and for no $j \in [k]$ is there an $L + 1 \leq x \leq R$ satisfying $b_j(x) = f(x)$. In particular, note that the vertex corresponding to such polynomial would be adjacent to all the vertices corresponding to a_j and non-adjacent to all vertices corresponding to b_j and would thus yield the desired property.

Claim 4.12. *For any fixed set of polynomials $\{a, b_1, \dots, b_k\} \subseteq V(G)$ and a $k - 1$ element set $A \subseteq \{L + 1, \dots, R\}$ there exists a point $x \in \{L + 1, \dots, R\} \setminus A$ such that $a(x) \notin \{b_1(x), \dots, b_k(x)\}$*

Proof. By Lemma 4.10, a agrees with b_i in at most d points. Hence, in total, there are at most dk points x such that $a(x) \in \{b_1(x), \dots, b_k(x)\}$.

Consequently, there are at least $R - (L + 1) - dk = k$ points x such that $a(x) \notin \{b_1(x), \dots, b_k(x)\}$ thus proving the desired. \square

Let $a_1, \dots, a_k, b_1, \dots, b_k$ be arbitrary vertices. By the claim above, we can find distinct elements x_1, \dots, x_k such that $a_i(x_i) \notin \{b_1(x_i), \dots, b_k(x_i)\}$ for any $1 \leq i \leq k$. Let ℓ be an arbitrary fixed label such that $\psi^{-1}(\ell) = (\ell_1, \dots, \ell_L)$. Consider the set S consisting of polynomials f satisfying $f(1) = \ell_1, \dots, f(L) = \ell_L$ and $f(x_i) = a_i(x_i)$ for every $1 \leq i \leq k$. Since $d = L + k$, by Lemma 4.11, if we fix a point $t \in \{L + 1, \dots, R\} \setminus \{x_1, \dots, x_k\}$, there is a unique polynomial f in S , such that $f(t) = b_j(t)$ for any $1 \leq j \leq k$. Hence, the set

$$T = \{f \in S \mid \exists L + 1 \leq t \leq R \exists 1 \leq j \leq k \text{ such that } f(t) = b_j(t)\}$$

contains at most $(R - L - k - 1)k$ elements.

Similarly, by Lemma 4.11, there are at least p distinct polynomials in S . Notice that it is sufficient to show that the set $S \setminus T$ is non-empty, as this would prove the existence of a vertex adjacent to all a_i and non-adjacent to all b_j for all i, j . To this end, we compute the value of

$$|S \setminus T| \geq p - (R - L - k - 1)k > p - R > 0.$$

Construction of such a graph can be implemented in polynomial time by evaluating each of the $\mathcal{O}(p^d)$ polynomials in at most $\mathcal{O}(p)$ points and then iterating through $\mathcal{O}(p^{2d})$ pairs of vertices and checking in time $\mathcal{O}(p)$ if they match in the corresponding coordinates. The total time complexity of the construction is thus at most

$$\mathcal{O}(dp^{d+1} + dp^{2d}) \leq \mathcal{O}(p^{3 \log_p(N)}) = \mathcal{O}(N^3).$$

Finally, we have to argue that $|V(G)| \leq \tilde{\mathcal{O}}(N)$. This, however follows from the simple observation that there are at most p^{d+1} distinct polynomials in $\mathbb{F}_p[X]_{\leq d}$ and

$$p^{d+1} \leq p^{\log_p(N) + k + 2} = Np^{k+2} \leq \tilde{\mathcal{O}}(N).$$

□

Combining the results from the last two subsections, we obtain the main result of this chapter.

Theorem 4.13. *Let Φ be a basic property with at least one negative literal. For any $\varepsilon > 0$, the Monochromatic Basic Problem corresponding to Φ cannot be solved faster than $\mathcal{O}(n^{k-\varepsilon})$ even on graphs with $\tilde{\mathcal{O}}(n)$ edges, unless SETH fails.*

5 Algorithms and Hardness of Distance- r k -Dominating Set

Let $r \geq 1$ be a fixed integer. We consider the *Distance- r k -Dominating Set* problem as a natural generalization of k -Dominating Set problem, where a vertex u dominates a vertex v if and only if there exists a path from u to v of length at most r . Clearly, if $r = 1$, we obtain the usual k -Dominating Set problem.

Formally, Distance- r k -Dominating Set problem can be stated as follows. Given an undirected graph $G = (V, E)$ with $|V| = n$ and $|E| = m$, decide if

$$\exists v_1 \in V \dots \exists v_k \in V : \forall w \in V : \min\{d_G(v_1, w), \dots, d_G(v_k, w)\} \leq r.$$

Let $B_r(v) = \{x \in V(G) \mid d(x, v) \leq r\}$.

In this section, we prove that sparsity does not affect the fine-grained complexity of Distance- r k -Dominating Set problem. In fact, we prove that a lower bound $n^{k \pm o(1)}$ holds already for graphs with $m = \tilde{\mathcal{O}}(n)$ edges, assuming the k -OV hypothesis. Furthermore, we prove that if $\omega = 2$, or k is sufficiently large, we can construct an algorithm that runs in $\tilde{\mathcal{O}}(n^{k+o(1)})$, thus matching the lower bound.

Theorem 5.1. *For any $\varepsilon > 0$, there exists no algorithm solving Distance- r k -Dominating Set problem in $\mathcal{O}(n^{k-\varepsilon})$ for any $k \geq 2, r \geq 2$, even in sparse graphs, unless SETH fails.*

Proof. To prove this lower bound, we reduce from k -OV. Let A_1, \dots, A_k be an instance of k orthogonal vectors problem. In particular, let A_1, \dots, A_k denote sets of d -dimensional (assume $d = \mathcal{O}(\log n)$) binary vectors of size n . Construct the graph G as follows. Start with an empty graph. For each set A_i , add one vertex for every vector in A_i , denoted a_j^i and two additional vertices A_i, A'_i . Add an edge between A_i and a_j^i and between A'_i and a_j^i for every i, j . Add an edge between A_i and A'_i and subdivide this edge $r - 2$ times and label the new vertices by $s_1(A_i, A'_i), \dots, s_{r-2}(A_i, A'_i)$. Add $2d$ vertices $D_1, \dots, D_d, D'_1, \dots, D'_d$ and add an edge between a_j^i and D_t if and only if $a_j^i[t] = 0$ for all i, j, t . Add an edge between D_t and D'_t for all t , and subdivide this edge $r - 2$ times. Label the new vertices by $s_1(D_t, D'_t), \dots, s_{r-2}(D_t, D'_t)$. The construction of such a graph is schematically depicted on Figure 3.

Clearly the constructed graph G has $\tilde{\mathcal{O}}(n)$ vertices and $\tilde{\mathcal{O}}(n)$ edges. We proceed to show that G contains k vertices v_1, \dots, v_k such that $\bigcup_{i=1}^k B_r(v_i) = V(G)$ if and only if there are vectors $a_1 \in A_1, \dots, a_k \in A_k$ such that $\prod_{i=1}^k a_i[j] = 0$ holds for all $1 \leq j \leq d$.

Suppose first that there are vectors $a_1 \in A_1, \dots, a_k \in A_k$ such that $\prod_{i=1}^k a_i[j] = 0$ for all $1 \leq j \leq d$. Then we claim that every vertex in G is at distance at most r from at least one a_i . Consider first the vertices A'_i and A_i . By construction A_i is adjacent to a_i and A'_i is at distance exactly r from a_i . Consequently also $s_1(A_i, a_i), \dots, s_{r-1}(A_i, a_i)$ are covered. Furthermore, since any vertex a_j^i is adjacent to A_i , $d(a_i, a_j^i) \leq 2$ for all i, j and since $r \geq 2$, also vertices a_j^i are covered. Finally, by construction, for any $1 \leq t \leq d$ there is an edge from some a_i to D_t , therefore since $d(D_t, D'_t) = r - 1$, we conclude that also all the vertices along the (unique) path from D_t to D'_t are covered and thus a_1, \dots, a_k cover all the vertices in G .

Conversely assume that there are vertices v_1, \dots, v_k such that $\bigcup_{i=1}^k B_r(v_i) = V(G)$. Consider vertices A'_1, \dots, A'_k . Clearly $B_r(A'_i)$ consists only of those vertices along the paths from A'_i to a_j^i for all j . In particular, since for all $i \neq j$ $B_r(A'_i)$ and $B_r(A'_j)$ are pairwise disjoint, the only possible solution is $v_i \in B_r(A'_i)$ (w.l.o.g.). Furthermore, for any $1 \leq t \leq d$, $B_r(D'_t)$ consists only of the vertices along the path from D'_t to some a_j^i , such that a_j^i is adjacent to D_i . Considering also the last constraint, we can conclude that for all t , there is a vertex in $\{v_1, \dots, v_k\}$ that is adjacent to D_t .

Hence, for every $v_i = a_j^i$ (for some j), take a_i to be the vector that corresponds a_j^i , and for v_i that lies along the path from A_i to A'_i , take a_i that corresponds to a_1^i , then we claim that $a_1 \in A_1, \dots, a_k \in A_k$ and for every $1 \leq t \leq d$ $\prod_{i=1}^k a_i[t] = 0$. By selection of the vectors a_i , clearly $a_i \in A_i$ and also, in G , if v_i is adjacent to D_t , then $v_i = a_j^i$ for some j and furthermore, $a_i[t] = 0$. But as we argued above, for every t such a vertex v_i exists and so $a_i[t] = 0$. The desired result follows directly. \square

Proposition 5.2. *Let G be a graph with n vertices and m edges and let $k \geq 2, r \geq 1$ be fixed constants. For sufficiently large k , or $\omega = 2$, we can solve Distance- r k -Dominating Set in time $\mathcal{O}(n^{k+o(1)})$.*

Proof. Without loss of generality, we may assume that $r > 1$, as otherwise we have an instance of k -dominating set problem and we already know that this statement holds. Let A be the adjacency matrix of G . Compute the matrix $(A + I)^r$. We can compute this matrix in $\mathcal{O}(n^{\omega+o(1)} \log(r)) = \mathcal{O}(n^{\omega+o(1)})$ time (by repeated squaring of matrix $A + I$). The entry i, j in the matrix $(A + I)^r$ is non-zero if and only if there is a walk of length at most r between i and j in G . Observe that every path is a walk, so if there exists a path of length at most r between i and j , the entry $(A + I)^r[i, j]$

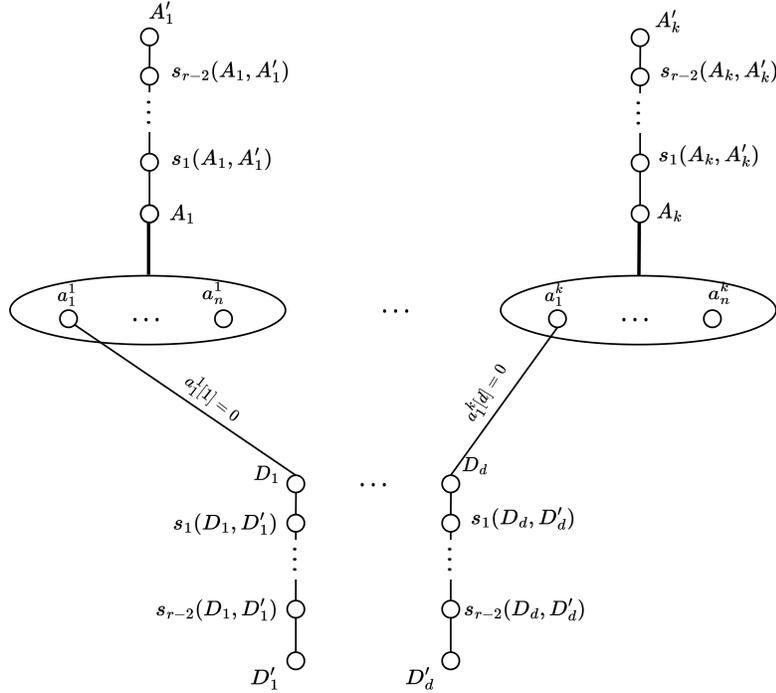


Figure 3: Schematic representation of reduction from k -OV to Distance- r k -Dominating Set problem. Bold edges represent bicliques between the corresponding sets.

will be non-zero. Furthermore, if there exists a walk of length at most r between i and j , then there exists a path of length at most r between them as well (obtained by removing all the cycles from the walk).

Let B be a matrix obtained by normalizing $(A + I)^r$. In particular, $B[i, j] = 1$ if and only if $(A + I)^r[i, j] > 0$, otherwise, $B[i, j] = 0$. and let G' be a graph with adjacency matrix $B - I_n$. Note that $V(G) = V(G')$. We claim that G' has dominating set at most k if and only if G has a dominating set of size k at distance r .

In particular, assume first that G has a Distance- r k -Dominating Set, $S = \{v_1, \dots, v_k\}$. Fix a vertex $v \in V(G)$. We want to show that v is dominated by some v_i in G' . If $v \in S$, we are done, so assume this is not the case. Then for some $1 \leq i \leq k$, there exists a (non-trivial) path of length $t \leq r$ between v_i and v in G . Hence, the entry corresponding to v_i, v in $(A + I)^r$ will be non-zero and since $v_i \neq v$, this entry will be equal to 1 in $B - I_n$ and therefore v_i is adjacent to v in G' . Since v was an arbitrary vertex, we may conclude that S dominates G' .

Conversely, assume that $S = \{v_1, \dots, v_k\}$ is a k -dominating set in G' and fix a vertex $v \in V(G)$. Then v is either contained in S , or adjacent to a vertex in S in G . If v was contained in S , then there exists a path of length $0 < r$ between v and a vertex in S and v is dominated by S . Otherwise, assume that v is adjacent to a vertex $v_i \in S$. Then there exists a walk of length $t \leq r$ in G and in particular, a path of length at most t . Hence, v is dominated by v_i and S dominates G as desired.

Observe that we can compute B in $\mathcal{O}(n^{\omega+o(1)})$, and as we proved in section 3, we can check if a graph G' has a dominating set of size k in $\mathcal{O}(n^{k+o(1)})$, assuming $k \geq 8$, or $k \geq 2$ and $\omega = 2$. \square

6 Closed Neighbourhood Containment and $(k - 1)$ -Covering

In this section we show that if we assume that the edge relation is reflexive (i.e., $E(v, v)$ for all $v \in V$), then the basic problem

$$\exists \text{pairwise distinct } v_1, \dots, v_k : \forall w \in V : \overline{E(v_1, w)} \vee E(v_2, w) \vee \dots \vee E(v_k, w)$$

becomes easier on sparse graphs. For $k = 2$, we call this problem *Closed Neighborhood Containment* (are there distinct $v_1, v_2 \in V$ such that $N[v_1] \subseteq N[v_2]$?). For general $k \geq 2$, we call it *Closed Neighborhood $(k - 1)$ -Covering* (are there pairwise distinct $v_1, \dots, v_k \in V$ such that $N[v_1] \subseteq N[v_2] \cup \dots \cup N[v_k]$?).

Closed Neighborhood Containment. To obtain faster algorithms on sparse graphs for $k = 2$, we exploit the following simple, yet crucial observation:

Observation 6.1. *Given a graph G , if there exists a pair of vertices x, y , such that $N[x] \subseteq N[y]$, then $E(x, y)$.*

We can use this fact to reduce the problem to the problem All-Edge Triangle Counting (Given an m -edge graph, for each edge $e \in E$, output the number of triangles containing e .), which can be solved in time $\mathcal{O}(m^{\frac{2\omega}{\omega+1}})$ [3].

Lemma 6.2. *Given a graph G , there exists a pair of vertices x, y , such that $N[x] \subseteq N[y]$ if and only if $E(x, y)$ and the number of triangles containing the edge $\{x, y\}$ is equal to $\deg(x) - 1$.*

Proof. Suppose first that $N[x] \subseteq N[y]$. Then by the previous lemma $E(x, y)$. Clearly, the number of triangles containing the edge $\{x, y\}$ is at most $\deg(x) - 1$, so it is sufficient to show that the lower bound holds as well. Assume for contradiction that this number is strictly smaller than $\deg(x) - 1$. Then there exists $z \in N(x)$ such that $z \notin N(y)$, contradicting the assumption that $N[x] \subseteq N[y]$.

Conversely, assume that for some pair of adjacent vertices x, y the number of triangles containing the edge $\{x, y\}$ is equal to $\deg(x) - 1$. Then clearly every vertex $z \neq y$ that is adjacent to x is contained in this triangle. In particular every $z \in N(x) \setminus \{y\}$ is adjacent to y as desired. \square

Theorem 6.3. *Closed Neighborhood Containment can be solved in $\mathcal{O}(m^{\frac{2\omega}{\omega+1}})$.*

Proof. By computing, for each $e \in E$, the number of triangles containing e in total time $\mathcal{O}(m^{\frac{2\omega}{\omega+1}})$, we can use the last lemma to decide any given instance in the same running time. \square

Closed Neighborhood $(k - 1)$ -Covering. The above algorithm easily generalizes to $k > 2$ by brute-forcing over $k - 2$ variables as follows:

Given a graph G with n vertices and m edges, fix distinct x_1, \dots, x_{k-2} arbitrarily. Construct a tripartite graph G' as follows. Let $V_1 = V_2 = V(G) \setminus \{x_1, \dots, x_{k-2}\}$ and let $V_3 = V(G) \setminus (N[x_1] \cup \dots \cup N[x_{k-2}])$. Let $V(G') = V_1 \cup V_2 \cup V_3$. Finally, add edges in G' as follows. If a pair $u \neq v$ is in $E(G)$, add the edge between the vertex corresponding to $u \in V_1$ and the vertex corresponding to $v \in V_2$. Add the edges between V_3 and V_1 , and V_3 and V_2 similarly. Finally, for any vertex v in V_3 , add the edge between the copy of v in V_3 and the copy of v in V_1 and V_2 . For each vertex $x \in V_1 \cup V_2$ store the number $\pi(x)$ of neighbours of x in V_3 . Observe first that G' has $\mathcal{O}(n)$ vertices and $\mathcal{O}(m)$ edges.

Lemma 6.4. *Given a graph G , and a set of distinct vertices x_1, \dots, x_{k-2} , there exists a pair of distinct vertices x_{k-1}, x_k , such that $N[x_k] \subseteq \bigcup_{i=1}^{k-1} N[x_i]$ and $E(x_{k-1}, x_k)$ if and only if in the graph G' as constructed above the number of triangles containing $x_{k-1} \in V_1$ and $x_k \in V_2$ is equal to $\pi(x_k)$.*

Proof. Suppose first that there exists a pair of distinct vertices x_{k-1}, x_k , such that $N[x_k] \subseteq \bigcup_{i=1}^{k-1} N[x_i]$ and $E(x_{k-1}, x_k)$. Consider the copy of x_{k-1} in V_1 and the copy of x_k in V_2 .

Fix any vertex v in V_3 and consider two cases.

- v corresponds to the same vertex as x_{k-1} , or x_k in G .
- v, x_{k-1}, x_k are all distinct vertices in G .

Notice that these are the only possibilities, since the endpoints of the edges going between V_1 and V_2 by construction correspond to distinct vertices in G .

If $v \in \{x_{k-1}, x_k\}$, since x_{k-1} and x_k are distinct vertices, adjacent in G , in G' the vertices $\{x_{k-1} \in V_1, x_k \in V_2, v \in V_3\}$ form a triangle.

Assume now that v, x_{k-1}, x_k are all distinct vertices in G . Since in G , v is not in $\bigcup_{i=1}^{k-2} N[x_i]$, it is either in $N(x_{k-1})$, or is not in $N(x_k)$. But by construction of G' this property is preserved and hence if $v \in N(x_k)$, then $v \in N(x_{k-1})$, or in other words for any vertex $v \in V_3$ that is adjacent to x_k , it holds that v, x_{k-1}, x_k form a triangle, giving us the lower bound. The upper bound is trivial from the fact that G' is tripartite.

Conversely, assume that for a pair of vertices $x_{k-1} \in V_1$ and $x_k \in V_2$, the number of triangles in G' containing these two vertices is equal to $\pi(x_k)$. Let v be an arbitrary vertex in G . If v is not in V_3 in G' , then it is contained in $\bigcup_{i=1}^{k-2} N[x_i]$ in G . On the other hand, if it is contained in V_3 , in G' it is either adjacent to $x_{k-1} \in V_1$, or nonadjacent to $x_k \in V_2$. In G , this means that v is either equal to x_k or x_{k-1} , or adjacent to x_{k-1} , or nonadjacent to x_k . In other words, if $v \in V_3$ in G' , then in G it is either contained in $N[x_{k-1}]$, or not contained in $N[x_k]$, as desired. \square

We can observe that the last lemma gives us an algorithm running in time $\mathcal{O}(n^{k-2}m^{\frac{2\omega}{\omega+1}})$, beating the $\Omega(n^k)$ running time for sufficiently sparse graphs.

Theorem 6.5. *Closed Neighborhood $(k-1)$ -Covering can be solved in time $\mathcal{O}(n^{k-2}m^{\frac{2\omega}{\omega+1}})$.*

Hardness of All Remaining Reflexive Properties. We now argue that for every monochromatic basic problem

$$\exists x_1 \in V, \dots, \exists x_{\ell} \in V, \exists x_{\ell+1} \in V, \dots, \exists x_k \in V \forall v \in V \left(\bigvee_{i=1}^{\ell} E(x_i, v) \right) \vee \left(\bigvee_{i=\ell+1}^k \overline{E(x_i, v)} \right),$$

the reduction given in Section 4 can be adapted for the reflexive case if the number of negative literals is $\ell' \geq 2$). Note that for $\ell' = 0$ (k -Dominating Set) and $\ell' = 1$ (Closed Neighborhood $(k-1)$ -Covering), we have shown faster algorithms in sparse graphs, so $\ell' \geq 2$ is indeed the only case remaining.

For $\ell' \geq 2$ we construct the reduction from a sparse Bichromatic Basic Problem to the corresponding Monochromatic Basic Problem similarly as in Section 4.

Lemma 6.6. *Let Φ be a basic property with k variables and at least two negative literals. Given an instance $G = (X, Y, E)$ of the Bichromatic Basic Problem corresponding to Φ with n vertices and assume $Y = \tilde{\mathcal{O}}(1)$, we can construct an equivalent instance $G' = (V, E)$ of the corresponding Monochromatic Basic Problem in reflexive setting in time $\tilde{\mathcal{O}}(n)$ consisting of at most $\tilde{\mathcal{O}}(n)$ edges.*

Proof. Consider the same construction of the graph G' as described in Section 4. If G is a no-instance, one can observe that the identical arguments show that G' is also a no-instance.

On the other hand, if G is a yes-instance, then we can find $x_1, \dots, x_k \in X$ such that for every $y \in Y$, the formula $E(x_1, y) \vee \dots \vee E(x_\ell, y) \vee \overline{E(x_{\ell+1}, y)} \vee \dots \vee \overline{E(x_k, y)}$ is satisfied. We claim that also $\forall v \in V(G')$, it holds that $E(x_1, v) \vee \dots \vee E(x_\ell, v) \vee \overline{E(x_{\ell+1}, v)} \vee \dots \vee \overline{E(x_k, v)}$. If $v \in V(H)$, then again the identical arguments as before show that v is satisfied. Thus, we may assume that $v \in X$. For every $v \in X \setminus \{x_{\ell+1}\}$, we observe that v is non-adjacent to $x_{\ell+1}$ and hence satisfied. However, since we are in the reflexive case, $x_{\ell+1}$ is adjacent to itself, and since there are no (non-trivial) edges inside X , $x_{\ell+1}$ has to be satisfied by another negative vertex (this is a subtle, but crucial reason why the reduction fails in the reflexive setting for $\ell' \leq 1$). Since $\ell' > 1$, notice that $x_{\ell+1}$ is satisfied by $x_{\ell+2}$. \square

We conclude this section by stating the consequential hardness result.

Theorem 6.7. *Let Φ be a basic property with at least two negative literals. For any $\varepsilon > 0$, the Monochromatic Basic Problem in the reflexive setting corresponding to Φ cannot be solved faster than $\mathcal{O}(n^{k-\varepsilon})$, even on graphs with $\tilde{\mathcal{O}}(n)$ edges, unless SETH fails.*

References

- [1] Udit Agarwal and Vijaya Ramachandran. Fine-grained complexity for sparse graphs. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 239–252. ACM, 2018. doi:10.1145/3188745.3188888.
- [2] Josh Alman and Virginia Vassilevska Williams. A refined laser method and faster matrix multiplication. In Dániel 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. doi:10.1137/1.9781611976465.32.
- [3] Noga Alon, Raphael Yuster, and Uri Zwick. Finding and counting given length cycles. *Algorithmica*, 17(3):209–223, 1997. doi:10.1007/BF02523189.
- [4] Rasmus Resen Amossen and Rasmus Pagh. Faster join-projects and sparse matrix multiplications. In Ronald Fagin, editor, *Database Theory - ICDT 2009, 12th International Conference, St. Petersburg, Russia, March 23-25, 2009, Proceedings*, volume 361 of *ACM International Conference Proceeding Series*, pages 121–126. ACM, 2009. doi:10.1145/1514894.1514909.
- [5] Haozhe An, Mohit Gurumukhani, Russell Impagliazzo, Michael Jaber, Marvin Künnemann, and Maria Paula Parga Nina. The fine-grained complexity of multi-dimensional ordering properties. *Algorithmica*, 84(11):3156–3191, 2022. doi:10.1007/s00453-022-01014-x.
- [6] Karl Bringmann, Alejandro Cassis, Nick Fischer, and Marvin Künnemann. Fine-grained completeness for optimization in P. In Mary Wootters and Laura Sanità, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2021, August 16-18, 2021, University of Washington, Seattle, Washington, USA (Virtual Conference)*, volume 207 of *LIPICs*, pages 9:1–9:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPICs.APPROX/RANDOM.2021.9.

- [7] Karl Bringmann, Alejandro Cassis, Nick Fischer, and Marvin Künnemann. A structural investigation of the approximability of polynomial-time problems. In Mikolaj Bojanczyk, Emanuela Merelli, and David P. Woodruff, editors, *49th International Colloquium on Automata, Languages, and Programming, ICALP 2022, July 4-8, 2022, Paris, France*, volume 229 of *LIPICs*, pages 30:1–30:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICs.ICALP.2022.30.
- [8] Karl Bringmann, Nick Fischer, and Marvin Künnemann. A fine-grained analogue of schaefer’s theorem in P: dichotomy of exists^k-forall-quantified first-order graph properties. In Amir Shpilka, editor, *34th Computational Complexity Conference, CCC 2019, July 18-20, 2019, New Brunswick, NJ, USA*, volume 137 of *LIPICs*, pages 31:1–31:27. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPICs.CCC.2019.31.
- [9] Karl Bringmann and Marvin Künnemann. Quadratic conditional lower bounds for string problems and dynamic time warping. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 79–97, 2015. doi:10.1109/FOCS.2015.15.
- [10] Parinya Chalermsook, Marek Cygan, Guy Kortsarz, Bundit Laekhanukit, Pasin Manurangsi, Danupon Nanongkai, and Luca Trevisan. From gap-exponential time hypothesis to fixed parameter tractable inapproximability: Clique, dominating set, and more. *SIAM J. Comput.*, 49(4):772–810, 2020. doi:10.1137/18M1166869.
- [11] Moses Charikar, Piotr Indyk, and Rina Panigrahy. New algorithms for subset query, partial match, orthogonal range searching, and related problems. In Peter Widmayer, Francisco Triguero Ruiz, Rafael Morales Bueno, Matthew Hennessy, Stephan J. Eidenbenz, and Ricardo Conejo, editors, *Automata, Languages and Programming, 29th International Colloquium, ICALP 2002, Malaga, Spain, July 8-13, 2002, Proceedings*, volume 2380 of *Lecture Notes in Computer Science*, pages 451–462. Springer, 2002. doi:10.1007/3-540-45465-9_39.
- [12] Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.
- [13] Søren Dahlgaard, Mathias Bæk Tejs Knudsen, and Morten Stöckel. Finding even cycles faster via capped k-walks. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 112–120. ACM, 2017. doi:10.1145/3055399.3055459.
- [14] Mark de Berg, Hans L. Bodlaender, Sándor Kisfaludi-Bak, Dániel Marx, and Tom C. van der Zanden. A framework for exponential-time-hypothesis-tight algorithms and lower bounds in geometric intersection graphs. *SIAM J. Comput.*, 49(6):1291–1331, 2020. doi:10.1137/20M1320870.
- [15] Rodney G. Downey and Michael R. Fellows. Fixed-parameter tractability and completeness I: basic results. *SIAM J. Comput.*, 24(4):873–921, 1995. doi:10.1137/S0097539792228228.
- [16] Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013. doi:10.1007/978-1-4471-5559-1.

- [17] Bartłomiej Dudek, Paweł Gawrychowski, and Tatiana Starikovskaya. All non-trivial variants of 3-ldt are equivalent. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 974–981. ACM, 2020. doi:10.1145/3357713.3384275.
- [18] Kord Eickmeyer, Archontia C. Giannopoulou, Stephan Kreutzer, O-joung Kwon, Michał Pilipczuk, Roman Rabinovich, and Sebastian Siebertz. Neighborhood complexity and kernelization for nowhere dense classes of graphs. In Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl, editors, *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland*, volume 80 of *LIPICs*, pages 63:1–63:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPICs.ICALP.2017.63.
- [19] 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.
- [20] Francois Le Gall and Florent Urrutia. Improved rectangular matrix multiplication using powers of the coppersmith-winograd tensor. 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 1029–1046. SIAM, 2018. doi:10.1137/1.9781611975031.67.
- [21] Jiawei Gao, Russell Impagliazzo, Antonina Kolokolova, and Ryan Williams. Completeness for first-order properties on sparse structures with algorithmic applications. *ACM Trans. Algorithms*, 15(2):23:1–23:35, 2019. doi:10.1145/3196275.
- [22] Teresa W. Haynes, Stephen T. Hedetniemi, and Peter J. Slater. *Fundamentals of domination in graphs*, volume 208 of *Pure and applied mathematics*. Dekker, 1998.
- [23] Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001. doi:10.1006/jcss.2001.1774.
- [24] David R. Karger. Minimum cuts in near-linear time. *J. ACM*, 47(1):46–76, 2000. doi:10.1145/331605.331608.
- [25] David R. Karger and Clifford Stein. A new approach to the minimum cut problem. *J. ACM*, 43(4):601–640, 1996. doi:10.1145/234533.234534.
- [26] 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.
- [27] Karthik C. S. and Pasin Manurangsi. On closest pair in euclidean metric: Monochromatic is as hard as bichromatic. *Comb.*, 40(4):539–573, 2020. doi:10.1007/s00493-019-4113-1.
- [28] Stephan Kreutzer, Roman Rabinovich, Sebastian Siebertz, and Grischa Weberstädt. Structural properties and constant factor-approximation of strong distance- r dominating sets in sparse directed graphs. In Heribert Vollmer and Brigitte Vallée, editors, *34th Symposium on Theoretical Aspects of Computer Science, STACS 2017, March 8-11, 2017, Hannover, Germany*, volume 66 of *LIPICs*, pages 48:1–48:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPICs.STACS.2017.48.

- [29] Andrea Lincoln and Nikhil Vyas. Algorithms and lower bounds for cycles and walks: Small space and sparse graphs. In Thomas Vidick, editor, *11th Innovations in Theoretical Computer Science Conference, ITCS 2020, January 12-14, 2020, Seattle, Washington, USA*, volume 151 of *LIPICs*, pages 11:1–11:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICs.ITCS.2020.11.
- [30] 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. doi:10.1137/1.9781611975031.80.
- [31] 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.
- [32] Mihai Pătraşcu and Ryan Williams. On the possibility of faster SAT algorithms. In Moses Charikar, editor, *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17-19, 2010*, pages 1065–1075. SIAM, 2010. doi:10.1137/1.9781611973075.86.
- [33] Virginia Vassilevska Williams. On some fine-grained questions in algorithms and complexity. In *Proceedings of the International Congress of Mathematicians, ICM '18*, pages 3447–3487, 2018.
- [34] 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.
- [35] Ryan Williams. Faster decision of first-order graph properties. In Thomas A. Henzinger and Dale Miller, editors, *Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), CSL-LICS '14, Vienna, Austria, July 14 - 18, 2014*, pages 80:1–80:6. ACM, 2014. doi:10.1145/2603088.2603121.
- [36] Raphael Yuster and Uri Zwick. Finding even cycles even faster. In Serge Abiteboul and Eli Shamir, editors, *Automata, Languages and Programming, 21st International Colloquium, ICALP94, Jerusalem, Israel, July 11-14, 1994, Proceedings*, volume 820 of *Lecture Notes in Computer Science*, pages 532–543. Springer, 1994. doi:10.1007/3-540-58201-0_96.
- [37] Raphael Yuster and Uri Zwick. Fast sparse matrix multiplication. *ACM Trans. Algorithms*, 1(1):2–13, 2005. doi:10.1145/1077464.1077466.