# Tight (Double) Exponential Bounds for Identification Problems: Locating-Dominating Set and Test Cover


**Dipayan Chakraborty**
Université Clermont Auvergne, CNRS, Mines Saint-Étienne, Clermont Auvergne INP, LIMOS, 63000 Clermont-Ferrand, France
Department of Mathematics and Applied Mathematics, University of Johannesburg, South Africa

**Florent Foucaud**
Université Clermont Auvergne, CNRS, Mines Saint-Étienne, Clermont Auvergne INP, LIMOS, 63000 Clermont-Ferrand, France

**Diptapriyo Majumdar**
Indraprastha Institute of Information Technology Delhi, New Delhi, India

**Prafullkumar Tale**
Indian Institute of Science Education and Research Bhopal, India



## Abstract

Foucaud et al. [ICALP 2024] demonstrated that some problems in NP can admit (tight) double-exponential lower bounds when parameterized by treewidth or vertex cover number. They showed these first-of-their-kind results by proving conditional lower bounds for certain graph problems, in particular, the metric-based identification problems (Strong) Metric Dimension. We continue this line of research and highlight the usefulness of this type of problems, to prove relatively rare types of (tight) lower bounds. We investigate fine-grained algorithmic aspects of classical (non-metric based) identification problems in graphs, namely Locating-Dominating Set, and in set systems, namely Test Cover. In the first problem, an input is a graph $G$ on $n$ vertices and an integer $k$, and the objective is to decide whether there is a subset $S$ of $k$ vertices such that any two distinct vertices not in $S$ are dominated by distinct subsets of $S$. In the second problem, an input is a set of items $U$, a collection of subsets $\mathcal{F}$ of $U$ called *tests*, and an integer $k$, and the objective is to select a set $S$ of at most $k$ tests such that any two distinct items are contained in a distinct subset of tests of $S$.

For our first result, we adapt the techniques introduced by Foucaud et al. [ICALP 2024] to prove similar (tight) lower bounds for these two problems.

- Locating-Dominating Set (respectively, Test Cover) parameterized by the treewidth of the input graph (respectively, the natural auxiliary graph) does not admit an algorithm running in time $2^{2^{o(\text{tw})}} \cdot \text{poly}(n)$ (respectively, $2^{2^{o(\text{tw})}} \cdot \text{poly}(|U| + |\mathcal{F}|)$), unless the ETH fails.

This augments the short list of NP-Complete problems that admit tight double-exponential lower bounds when parameterized by treewidth, and shows that "local" (non-metric-based) problems can also admit such bounds. We show that these lower bounds are tight by designing treewidth-based dynamic programming schemes with matching running times.

Next, we prove that these two problems also admit "exotic" (and tight) lower bounds, when parameterized by the solution size $k$. We prove that unless the ETH fails,

- Locating-Dominating Set does not admit an algorithm running in time $2^{o(k^2)} \cdot \text{poly}(n)$, nor a polynomial-time kernelization algorithm that reduces the solution size and outputs a kernel with $2^{o(k)}$ vertices, and
- Test Cover does not admit an algorithm running in time $2^{2^{o(k)}} \cdot \text{poly}(|U| + |\mathcal{F}|)$ nor a kernel with $2^{2^{o(k)}}$ vertices.

Again, we show that these lower bounds are tight by designing (kernelization) algorithms with matching running times. To the best of our knowledge, Locating-Dominating Set is the first known problem which is FPT when parameterized by solution size $k$, where the optimal running time has a quadratic function in the exponent. These results also extend the (very) small list of problems that admit an ETH-based lower bound on the number of vertices in a kernel, and (for




35th International Symposium on Algorithms and Computation (ISAAC 2024).
Editors: Julián Mestre and Anthony Wirth; Article No. 19; pp. 19:1–19:18
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Test Cover) a double-exponential lower bound when parameterized by the solution size. Whereas it is the first example, to the best of our knowledge, that admit a double exponential lower bound for the number of vertices.

**2012 ACM Subject Classification** Theory of computation → Fixed parameter tractability

**Keywords and phrases** Identification Problems, Locating-Dominating Set, Test Cover, Double Exponential Lower Bound, ETH, Kernelization Lower Bounds

**Digital Object Identifier** 10.4230/LIPIcs.ISAAC.2024.19

**Related Version** *Full Version*: `https://arxiv.org/abs/2402.08346`

**Funding** *Dipayan Chakraborty*: International Research Center "Innovation Transportation and Production Systems" of the I-SITE CAP 20-25.
*Florent Foucaud*: ANR project GRALMECO (ANR-21-CE48-0004), French government IDEX-ISITE initiative 16-IDEX-0001 (CAP 20-25), International Research Center "Innovation Transportation and Production Systems" of the I-SITE CAP 20-25.
*Diptapriyo Majumdar*: Supported by Science and Engineering Research Board (SERB) grant SRG/2023/001592.
*Prafullkumar Tale*: Supported by INSPIRE Faculty Fellowship DST/INSPIRE/04/2021/00314.

## 1 Introduction

The article aims to study the algorithmic properties of certain identification problems in discrete structures. In identification problems, one wishes to select a solution substructure of an input structure (a subset of vertices, the coloring of a graph, etc.) so that the solution substructure uniquely identifies each element. Some well-studied examples are, for example, the problems Test Cover for set systems and Metric Dimension for graphs (Problems [SP6] and [GT61] in the book by Garey and Johnson [39], respectively). This type of problem has been studied since the 1960s both in the combinatorics community (see e.g. Rényi [60] or Bondy [9]), and in the algorithms community since the 1970s [7, 10, 25, 55]. They have multiple practical and theoretical applications, such as network monitoring [59], medical diagnosis [55], bioinformatics [7], coin-weighing problems [62], graph isomorphism [4], games [19], machine learning [18] etc. An online bibliography on the topic with over 500 entries as of 2024 is maintained at [46].

In this article, we investigate fine-grained algorithmic aspects of identification problems in graphs, namely Locating-Dominating Set, and in set systems, namely Test Cover. Like most other interesting and practically motivated computational problems, identification problems also turned out to be NP-hard, even in very restricted settings. See, for example, [20] and [39], respectively. We refer the reader to "Related Work" towards the end of this section for a more detailed overview on their algorithmic complexity.

To cope with this hardness, these problems have been studied through the lens of parameterized complexity. In this paradigm, we associate each instance $I$ with a parameter $\ell$, and are interested to know whether the problem admits a *fixed parameter tractable* (FPT) algorithm, i.e., an algorithm with the running time $f(\ell) \cdot |I|^{\mathcal{O}(1)}$, for some computable function $f$. A parameter can either originate from the formulation of the problem itself or can be a property of the input. If a parameter originates from the formulation of the problem itself, then that is called a *natural parameter*. Otherwise, the parameters that are properties of the input graph are called the *structural parameters*. One of the most well-studied structural parameters is 'treewidth' (which, informally, quantifies how close the input graph is to a tree, and is denoted by tw). We refer readers to [26, Chapter 7] for a formal definition. Courcelle's celebrated theorem [21] states that the class of graph problems expressible in Monadic Second-Order Logic (MSOL) of constant size admit an algorithm running in time $f(\mathsf{tw}) \cdot \mathsf{poly}(n)$. Hence, a large class of problems admit an FPT algorithm when parameterized by the treewidth. Unfortunately, the function $f$ is a tower of exponents whose height depends roughly on the size of the MSOL formula. Hence, this result serves as a starting point to obtain an (usually impractical) FPT algorithm.

Over the years, researchers have searched for more efficient problem-specific algorithms when parameterized by the treewidth. There is a rich collection of problems that admit an FPT algorithm with single- or almost-single-exponential dependency with respect to treewidth, i.e., of the form $2^{\mathcal{O}(\mathsf{tw})} \cdot n^{\mathcal{O}(1)}$ or $2^{\mathcal{O}(\mathsf{tw}\log(\mathsf{tw}))} \cdot n^{\mathcal{O}(1)}$, (see, for example, [26, Chapter 7]). There are a handful of graph problems that only admit FPT algorithms with double- or triple-exponential dependence in the treewidth [8, 30, 31, 32, 42, 54]. In the respective articles, the authors prove that this double- (respectively, triple-) dependence in the treewidth cannot be improved unless the Exponential Time Hypothesis (ETH)[1] fails.

All the double- (or triple-) exponential lower bounds in treewidth mentioned in the previous paragraph are for problems that are #NP-complete, $\Sigma_2^p$-complete, or $\Pi_2^p$-complete. Indeed, until recently, this type of lower bounds were known only for problems that are complete for levels that are higher than NP in the polynomial hierarchy. Foucaud et al. [36] recently proved for the first time, that it is not necessary to go to higher levels of the polynomial hierarchy to achieve double-exponential lower bounds in the treewidth. The authors studied three NP-complete *metric-based graph problems* viz Metric Dimension, Strong Metric Dimension, and Geodetic Set. They proved that these problems admit double-exponential lower bounds in tw (and, in fact in the size of minimum vertex cover size vc for the second problem) under the ETH. The first two of these three problems are identification problems.

In this article, we continue this line of research and highlight the usefulness of identification problems to prove relatively rare types of lower bounds, by investigating fine-grained algorithmic aspects of Locating-Dominating Set and Test Cover, two classical (non-metric-based) identification problems. This also shows that this type of bounds can hold for "local" (i.e., non-metric-based) problems (the problems studied in [36] were all metric-based). Apart from serving as examples for double-exponential dependence on treewidth, these problems are of interest in their own right, and possess a rich literature both in the algorithms and discrete mathematics communities, as highlighted in "Related Work".

Locating-Dominating Set
**Input:** A graph $G$ on $n$ vertices and an integer $k$.
**Question:** Does there exist a locating-dominating set of size $k$ in $G$, that is, a set $S$ of $V(G)$ of size at most $k$ such that for any two different vertices $u, v \in V(G) \setminus S$, their neighbourhoods in $S$ are different, i.e., $N(u) \cap S \neq N(v) \cap S$ and non-empty?

Test Cover
**Input:** A set of items $U$, a collection $\mathcal{F}$ of subsets of $U$ called *tests*, and an integer $k$.
**Question:** Does there exist a collection of at most $k$ tests such that for each pair of items, there is a test that contains exactly one of the two items?

[1] The ETH roughly states that $n$-variable 3-SAT cannot be solved in time $2^{o(n)}n^{\mathcal{O}(1)}$. See [26, Chapter 14].

As Test Cover is defined over set systems, for structural parameters, we define an *auxiliary graph* in the natural way: A bipartite graph $G$ on $n$ vertices with bipartition $\langle R, B\rangle$ of $V(G)$ such that sets $R$ and $B$ contain a vertex for every set in $\mathcal{F}$ and for every item in $U$, respectively, and $r \in R$ and $b \in B$ are adjacent if and only if the set corresponding to $r$ contains the element corresponding to $b$.

The Locating-Dominating Set problem is also a *graph domination problem.* In the classical Dominating Set problem, an input is an undirected graph $G$ and an integer $k$, and the objective is to decide whether there is a subset $S \subseteq V(G)$ of size $k$ such that for any vertex $u \in V(G) \setminus S$, at least one of its neighbours is in $S$. It can also be seen as a local version of Metric Dimension[2] in which the input is the same and the objective is to determine a set $S$ of $V(G)$ such that for any two vertices $u, v \in V(G) \setminus S$, there exists a vertex $s \in S$ such that $dist(u, s) \neq dist(v, s)$.

We demonstrate the applicability of the techniques from [36] to Locating-Dominating Set and Test Cover. We adopt the main technique developed in [36] to our setting, namely, the *bit-representation gadgets* and *set representation gadget* to prove the following result.

▶ **Theorem 1.** *Unless the ETH fails, Locating-Dominating Set (respectively, Test Cover) parameterized by the treewidth of the input graph (respectively, the natural auxiliary graph) does not admit an algorithm running in time* $2^{2^{o(\mathsf{tw})}} \cdot \text{poly}(n)$.

We remark that the algorithmic lower bound of Theorem 1 holds true even with respect to treedepth (and hence with respect to pathwidth), a parameter larger than treewidth. In contrast, Dominating Set admits an algorithm running in time $\mathcal{O}(3^{\mathsf{tw}} \cdot n^2)$ [67, 52]. In the full version of the paper, we prove that both Locating-Dominating Set and Test Cover admit an algorithm with matching running time, by nontrivial dynamic programming schemes on tree decompositions.

Theorem 1 adds Locating-Dominating Set and Test Cover to the short list of NP-Complete problems that admit (tight) double-exponential lower bounds for treewidth. Using the techniques mentioned in [36], two more problems, viz. Non-Clashing Teaching Map and Non-Clashing Teaching Dimension, from learning theory were recently shown in [14] to admit similar lower bounds.

Next, we prove that Locating-Dominating Set and Test Cover also admit "exotic" lower bounds, when parameterized by the solution size $k$. First, note that both problems are trivially FPT when parameterized by the solution size. Indeed, as any solution must have size at least logarithmic in the number of elements/vertices (assuming no redundancy in the input), the whole instance is a trivial single-exponential kernel for Locating-Dominating Set, and double-exponential in the case of Test Cover. To see this, note that in both problems, any two vertices/items must be assigned a distinct subset from the solution set. Hence, if there are more than $2^k$ of them, we can safely reject the instance. Thus, for Locating-Dominating Set, we can assume that the graph has at most $2^k + k$ vertices, and for Test Cover, at most $2^k$ items. Moreover, for Test Cover, one can also assume that every test is unique (otherwise, delete any redundant test), in which case there are at most $2^{2^k}$ tests. Hence, Locating-Dominating Set admits a kernel with size $\mathcal{O}(2^k)$, and an FPT algorithm running in time $2^{\mathcal{O}(k^2)}$ (See Proposition 7). We prove that both of these bounds are optimal.

[2] Note that Metric Dimension is also an identification problem, but it is inherently non-local in nature, and indeed was studied together with two other non-local problems in [36], where the similarities between these non-local problems were noticed.

▶ **Theorem 2.** *Unless the* ETH *fails,* Locating-Dominating Set*, parameterized by the solution size $k$, does not admit*

- *an algorithm running in time $2^{o(k^2)} \cdot n^{\mathcal{O}(1)}$, nor*
- *a polynomial time kernelization algorithm that reduces the solution size and outputs a kernel with $2^{o(k)}$ vertices.*

To the best of our knowledge, Locating-Dominating Set is the first known problem to admit such an algorithmic lower bound, with a matching upper bound, when parameterized by the solution size. The only other problems known to us, admitting similar lower bounds, are for structural parameterizations like vertex cover [1, 14, 36] or pathwidth [58, 61]. The second result is also quite rare in the literature. The only results known to us about ETH-based conditional lower bounds on the number of vertices in a kernel when parameterized by the solution size are for Edge Clique Cover [27] and Biclique Cover [15][3]. Theorem 2 also improves upon a "no $2^{\mathcal{O}(k)}n^{\mathcal{O}(1)}$ algorithm" bound from [5] (under $\mathsf{W}[2] \neq \mathsf{FPT}$) and a $2^{o(k \log k)}$ ETH-based lower bound recently proved in [11].

Now, consider the case of Test Cover. As mentioned before, it is safe to assume that $|\mathcal{F}| \leq 2^{|U|}$ and $|U| \leq 2^k$. By Bondy's celebrated theorem [9], which asserts that in any feasible instance of Test Cover, there is always a solution of size at most $|U| - 1$, we can also assume that $k \leq |U| - 1$. Hence, the brute-force algorithm that enumerates all the sub-collections of tests of size at most $k$ runs in time $|\mathcal{F}|^{\mathcal{O}(|U|)} = 2^{\mathcal{O}(|U|^2)} = 2^{2^{\mathcal{O}(k)}}$. Our next result proves that this simple algorithm is again optimal.

▶ **Theorem 3.** *Unless the* ETH *fails,* Test Cover *does not admit*

- *an algorithm running in time $2^{2^{o(k)}} \cdot (|U| + |\mathcal{F}|)^{\mathcal{O}(1)}$, nor*
- *a polynomial time kernelization algorithm that reduces the solution size and outputs a kernel with $2^{2^{o(k)}}$ vertices.*

This result adds Test Cover to the relatively rare list of NP-complete problems that admit such double-exponential lower bounds when parameterized by the solution size and the matching algorithm. The only other examples that we know of are Edge Clique Cover [27], Distinct Vectors Problem [57], and Telephone Broadcast [66]. For double-exponential algorithmic lower bounds with respect to structural parameters, please see [33, 42, 45, 47, 48, 50, 51, 53].

The second result in the theorem is a simple corollary of the first result. Assume that the problem admits a kernel with $2^{2^{o(k)}}$ vertices. Then, the brute-force enumerating all the possible solutions works in time $\binom{2^{2^{o(k)}}}{k} \cdot (|U| + |\mathcal{F}|)^{\mathcal{O}(1)}$, which is $2^{k \cdot 2^{o(k)}} \cdot (|U| + |\mathcal{F}|)^{\mathcal{O}(1)}$, which is $2^{2^{o(k)}} \cdot (|U| + |\mathcal{F}|)^{\mathcal{O}(1)}$, contradicting the first result. To the best of our knowledge, Test Cover is the first problem that admit a double exponential kernelization lower bound for the number of vertices when parameterized by solution size, or by any natural parameter.

**Related Work.** Locating-Dominating Set was introduced by Slater in the 1980s [63, 64]. The problem is NP-complete [20], even for special graph classes such as planar unit disk graphs [56], planar bipartite subcubic graphs, chordal bipartite graphs, split graphs and co-bipartite graphs [35], interval and permutation graphs of diameter 2 [38]. By a straightforward application of Courcelle's theorem [22], Locating-Dominating Set is FPT for parameter treewidth and even cliquewidth [23]. Explicit polynomial-time algorithms were given for trees [63], block graphs [3], series-parallel graphs [20], and cographs [37]. Regarding the approximation complexity of Locating-Dominating Set, see [35, 40, 65].

[3] Additionally, Point Line Cover does not admit a kernel with $\mathcal{O}(k^{2-\epsilon})$ *vertices*, for any $\epsilon > 0$, unless $\mathsf{NP} \subseteq \mathsf{coNP}/poly$ [49].

It was shown in [5] that Locating-Dominating Set cannot be solved in time $2^{o(n)}$ on bipartite graphs, nor in time $2^{o(\sqrt{n})}$ on planar bipartite graphs or on apex graphs, assuming the ETH. Moreover, they also showed that Locating-Dominating Set cannot be solved in time $2^{\mathcal{O}(k)} n^{\mathcal{O}(1)}$ on bipartite graphs, unless $\mathsf{W}[2] = \mathsf{FPT}$. Note that the authors of [5] have designed a complex framework with the goal of studying a large class of identification problems related to Locating-Dominating Set and similar problems.

In [12], structural parameterizations of Locating-Dominating Set were studied. It was shown that the problem admits a linear kernel for the parameter max-leaf number, however (under standard complexity assumptions) no polynomial kernel exists for the solution size, combined with either the vertex cover number or the distance to a clique. They also provide a double-exponential kernel for the parameter distance to the cluster. In the full version [11] of [12], the same authors show that Locating-Dominating Set does neither admit a $2^{o(k \log k)} n^{\mathcal{O}(1)}$-time nor an $n^{o(k)}$-time algorithm, assuming the ETH.

Test Cover was shown to be NP-complete by Garey and Johnson [39, Problem SP6] and it is also hard to approximate within a ratio of $(1 - \epsilon) \ln n$ [10] (an approximation algorithm with ratio $1 + \ln n$ exists by reduction to Set Cover [7]). As any solution has size at least $\log_2(n)$, the problem admits a trivial kernel of size $2^{2^k}$, and thus Test Cover is FPT parameterized by solution size $k$. Test Cover was studied within the framework of "above/below guarantee" parameterizations in [6, 24, 25, 41] and kernelization in [6, 24, 41]. These results have shown an intriguing behavior for Test Cover, with some nontrivial techniques being developed to solve the problem [6, 25]. Test Cover is FPT for parameters $n-k$, but $W[1]$-hard for parameters $m-k$ and $k-\log_2(n)$ [25]. However, assuming standard assumptions, there is no polynomial kernel for the parameterizations by $k$ and $n-k$ [41], although there exists a "partially polynomial kernel" for parameter $n-k$ [6] (i.e. one with $O((n-k)^7)$ elements, but potentially exponentially many tests). When the tests have all a fixed upper bound $r$ on their size, the parameterizations by $k$, $n-k$ and $m-k$ all become FPT with a polynomial kernel [24, 41].

The problem Discriminating Code [16] is very similar to Test Cover (with the distinction that the input is presented as a bipartite graph, one part representing the elements and the other, the tests, and that every element has to be covered by some solution test), and has been shown to be NP-complete even for planar instances [17].

**Organization.** Due to the space constraints, we present overviews of the reductions in this extended abstract. Formal proofs for the arguments can be found in the full version of the paper. We use the Locating-Dominating Set problem to demonstrate key technical concepts regarding our lower bounds and algorithms. We present an overview of the arguments about Locating-Dominating Set in Sections 3 and 4. The arguments regarding Test Cover follow the identical line, and an overview is presented in Section 5. We conclude with an open problem in Section 6.

## 2 Preliminaries

For a positive integer $q$, we denote the set $\{1, 2, \ldots, q\}$ by $[q]$. We use $\mathbb{N}$ to denote the collection of all non-negative integers.

**Graph theory.** We use standard graph-theoretic notation, and we refer the reader to [28] for any undefined notation. For an undirected graph $G$, sets $V(G)$ and $E(G)$ denote its set of vertices and edges, respectively. We denote an edge with two endpoints $u, v$ as $uv$. Unless otherwise specified, we use $n$ to denote the number of vertices in the input graph $G$ of the problem under consideration. Two vertices $u, v$ in $V(G)$ are *adjacent* if there is an edge $uv$ in $G$. The *open neighborhood* of a vertex $v$, denoted by $N_G(v)$, is the set of vertices adjacent to $v$. The *closed neighborhood* of a vertex $v$, denoted by $N_G[v]$, is the set $N_G(v) \cup \{v\}$. We say that a vertex $u$ is a *pendant vertex* if $|N_G(v)| = 1$. We omit the subscript in the notation for neighborhood if the graph under consideration is clear. For a subset $S$ of $V(G)$, we define $N[S] = \bigcup_{v \in S} N[v]$ and $N(S) = N[S] \setminus S$. For a subset $S$ of $V(G)$, we denote the graph obtained by deleting $S$ from $G$ by $G - S$. We denote the subgraph of $G$ induced on the set $S$ by $G[S]$.

**Locating-Dominating Sets.** A subset of vertices $S$ in graph $G$ is called its *dominating set* if $N[S] = V(G)$. A dominating set $S$ is said to be a *locating-dominating set* if for any two different vertices $u, v \in V(G) \setminus S$, we have $N(u) \cap S \neq N(v) \cap S$. In this case, we say vertices $u$ and $v$ are *distinguished* by the set $S$. We say a vertex $u$ is *located* by set $S$ if for any vertex $v \in V(G) \setminus \{u\}$, $u$ and $v$ are distinguished by $S$ (equivalently $N(u) \cap S \neq N(v) \cap S$). Note that, if $S$ locates $u$, then any superset $S' \supset S$ also locates $u$. By extension, a set $X$ is *located* by $S$ if all vertices in $X$ are located by $S$. We note the following simple observation (see also [13, Lemma 5]).

▶ **Observation 4.** *If $S$ is a locating-dominating set of a graph $G$, then there exists a locating-dominating set $S'$ of $G$ such that $|S'| \leq |S|$ and that contains all vertices that are adjacent with a pendant vertices (i.e. vertices of degree 1) in $G$.*

**Proof.** Let $u$ be a pendant vertex which is adjacent with a vertex $v$ of $G$. We now look for a locating dominating set $S'$ of $G$ such that $|S'| \leq |S|$ and contains the vertex $v$. As $S$ is a (locating) dominating set, we have $\{u, v\} \cap S \neq \emptyset$. If $v \in S$, then take $S' = S$. Therefore, let us assume that $u \in S$ and $v \notin S$. Define $S' = (S \cup \{v\}) \setminus \{u\}$. It is easy to see that $S'$ is a dominating set. If $S'$ is not a locating-dominating set, then there exists $w$, apart from $u$, in the neighbourhood of $v$ such that both $u$ and $w$ are adjacent with *only* $v$ in $S'$. As $u$ is a pendant vertex and $v$ its unique neighbour, $w$ is not adjacent to $u$. Hence, $w$ was not adjacent with any vertex in $S' \setminus \{v\} = S \setminus \{u\}$. This, however, contradicts the fact that $S$ is a (locating) dominating set. Hence, $S'$ is a locating-dominating set and $|S'| = |S|$. Thus, the result follows from repeating this argument for each vertex of $G$ adjacent to a pendant vertex. ◀

**Parameterized complexity.** An instance of a parameterized problem Π consists of an input $I$, which is an input of the non-parameterized version of the problem, and an integer $k$, which is called the *parameter*. Formally, $\Pi \subseteq \Sigma^* \times \mathbb{N}$. A problem Π is said to be *fixed-parameter tractable*, or FPT, if given an instance $(I, k)$ of Π, we can decide whether $(I, k)$ is a YES-instance of Π in time $f(k) \cdot |I|^{\mathcal{O}(1)}$. Here, $f : \mathbb{N} \mapsto \mathbb{N}$ is some computable function depending only on $k$. A parameterized problem Π is said to admit a *kernelization* if given an instance $(I, k)$ of Π, there is an algorithm that runs in time polynomial in $|I| + k$ and constructs an instance $(I', k')$ of Π such that (i) $(I, k) \in \Pi$ if and only if $(I', k') \in \Pi$, and (ii) $|I'| + k' \leq g(k)$ for some computable function $g : \mathbb{N} \mapsto \mathbb{N}$ depending only on $k$. If $g(\cdot)$ is a polynomial function, then Π is said to admit a *polynomial* kernelization. For a detailed introduction to parameterized complexity and related terminologies, we refer the reader to the recent books by Cygan et al. [26] and Fomin et al. [34].

## 3 Locating-Dominating Set Parameterized by Treewidth

We first present a bird's eye overview of the dynamic programming algorithm. Let $\mathcal{T} = (T, \{X_t\}_{t \in V(T)})$ be a nice tree decomposition of $G$. For every node $t \in V(T)$, consider the subtree $T_t$ of $T$ rooted at $t$. Let $G_t$ denote the subgraph of $G$ that is induced by the vertices that are present in the bags of $T_t$. For every node $t \in T$, we define a subproblem (or DP-state) using a tuple $[t, (Y, W), (\mathcal{A}, \mathcal{D}), \mathcal{B}]$. Consider a partition $(Y, W, X_t \setminus (Y \cup W))$ of $X_t$. The first part denotes the vertices in the partial solution $S$. The second part denotes the vertices in $X_t$ that are dominated (but need not be located) by the solution vertices in $G_t$ but that are outside $X_t$. To extend this partial solution, we need to keep track of vertices that are adjacent to a unique subset in $Y$. For example, suppose there is vertex $u \in V(G_t) \setminus (S \cup X_t)$ such that $N_{G_t}(u) \cap S = A$ for some subset $A \subseteq Y$. Then $u$ still needs to be located by $S$. It means that there should not be a vertex, say $v$, in $V(G) \setminus V(G_t)$ such that $N_{G_t}(v) \cap S' = A$, where $S'$ is an extension of the partial solution $S$. Hence, we need to keep track of all such vertices by keeping track of the neighbourhood of all such vertices. We define $\mathcal{A}$, which is a subset of the power set of $Y$, to store all such sets that are the neighborhoods of vertices in $V(G_t) \setminus X_t$. Similarly, we define $\mathcal{D}$ to store all such sets with respect to vertices that are in $X_t$. Finally, we define $\mathcal{B}$ to store the pairs of vertices that need to be resolved by the extension of the partial solution. We formalise these ideas in the full version of the paper to prove the following theorem.

▶ **Theorem 5.** *Locating-Dominating Set, parameterized by the treewidth* $\mathsf{tw}$ *of the input graph admits an algorithm running in time* $2^{2^{\mathcal{O}(\mathsf{tw})}} \cdot n^{\mathcal{O}(1)}$.

In the remainder of this section, we prove the lower bound mentioned in Theorem 1 by presenting a reduction from a variant of 3-SAT called $(3,3)$-SAT. In this variation, an input is a boolean satisfiability formula $\psi$ in conjunctive normal form such that each clause contains *at most* 3 variables, and each variable appears at most 3 times. Using the ETH [43], the sparcification lemma [44], and a simple reduction from 3-SAT, we have the following result.

▶ **Proposition 6.** $(3,3)$*-SAT, with $n$ variables and $m$ clauses, does not admit an algorithm running in time* $2^{o(m+n)}$*, unless the* ETH *fails.*

We highlight that every variable appears positively and negatively at least once. Otherwise, if a variable appears only positively (respectively, only negatively) then we can assign it `True` (respectively, `False`) and safely reduce the instance by removing the clauses containing this variable. Hence, instead of the first, second, or third appearance of the variable, we use the first positive, first negative, second positive, or second negative appearance of the variable.

**Reduction.** The reduction takes as input an instance $\psi$ of $(3,3)$-SAT with $n$ variables and outputs an instance $(G, k)$ of Locating-Dominating Set such that $\mathsf{tw}(G) = \mathcal{O}(\log(n))$. Suppose $X = \{x_1, \ldots, x_n\}$ is the collection of variables and $C = \{C_1, \ldots, C_m\}$ is the collection of clauses in $\psi$. Here, we consider $\langle x_1, \ldots, x_n \rangle$ and $\langle C_1, \ldots, C_m \rangle$ to be arbitrary but fixed orderings of variables and clauses in $\psi$. For a particular clause, the first order specifies the first, second, or third (if it exists) variable in the clause in a natural way. The second ordering specifies the first/second positive/negative appearance of variables in $X$ in a natural way. The reduction constructs a graph $G$ as follows.

- To construct a variable gadget for $x_i$, it starts with two claws $\{\alpha_i^0, \alpha_i^1, \alpha_i^2, \alpha_i^3\}$ and $\{\beta_i^0, \beta_i^1, \beta_i^2, \beta_i^3\}$ centered at $\alpha_i^0$ and $\beta_i^0$, respectively. It then adds four vertices $x_i^1, \neg x_i^1, x_i^2, \neg x_i^2$, and the corresponding edges, as shown in Figure 1. Let $A_i$ be the collection of these twelve vertices and we define $A = \cup_{i=1}^n A_i$. Define $X_i := \{x_i^1, \neg x_i^1, x_i^2, \neg x_i^2\}$.

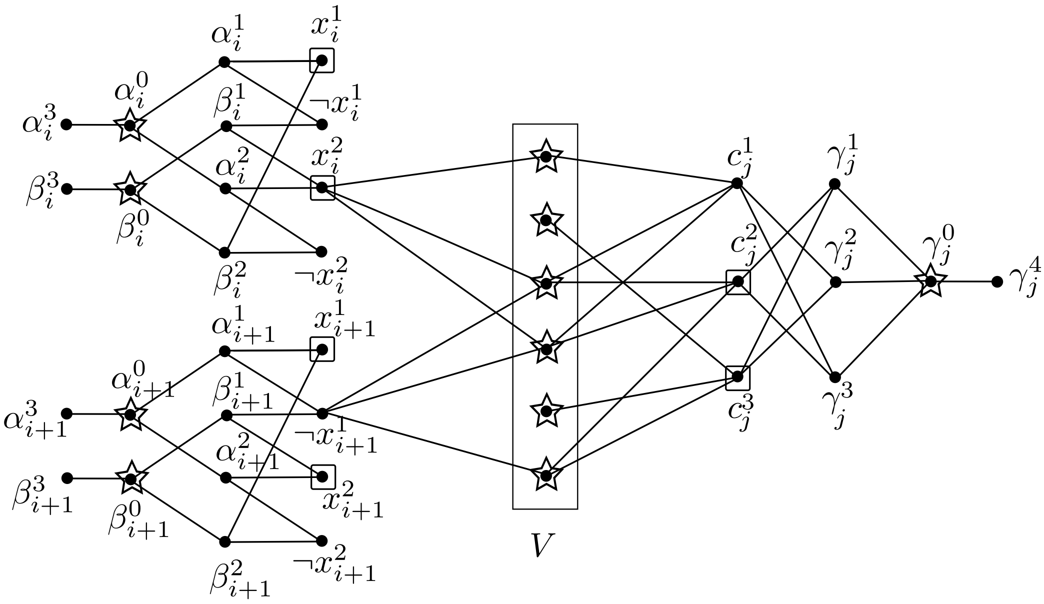


**Figure 1** For the sake of clarity, we do not explicitly show the pendant vertices adjacent to vertices in $V$. The variable and clause gadgets are on the left-side and right-side of $V$, respectively. In this example, we consider a clause $C_j = x_i \vee \neg x_{i+1} \vee x_{i+2}$. Moreover, suppose this is the second positive appearance of $x_i$ and the first negative appearance of $x_{i+1}$, and $x_i$ corresponds to $c_j^1$ and $x_{i+1}$ corresponds to $c_j^2$. Suppose $V$ contains 6 vertices indexed from top to bottom, and the set corresponding to these two appearances are $\{1, 3, 4\}$ and $\{3, 4, 6\}$ respectively. The star boundary denote the vertices that we can assume to be in any locating-dominating set, without loss of generality. The square boundary corresponds to selection of other vertices in $S$. In the above example, it corresponds to setting both $x_i$ and $x_{i+1}$ to `True`. On the clause side, the selection corresponds to selecting $x_i$ to satisfy the clause $C_j$.

- To construct a clause gadget for $C_j$, the reduction starts with a star graph centered at $\gamma_j^0$ and with four leaves $\{\gamma_j^1, \gamma_j^2, \gamma_j^3, \gamma_j^4\}$. It then adds three vertices $c_j^1, c_j^2, c_j^3$ and the corresponding edges shown in Figure 1. Let $B_j$ be the collection of these eight vertices and $B = \cup_{j=1}^m B_j$.
- Let $p$ be the smallest positive integer such that $4n \leq \binom{2p}{p}$. Define $\mathcal{F}_p$ as the collection of subsets of $[2p]$ that contains exactly $p$ integers (such a collection $\mathcal{F}_p$ is called a *Sperner family*). Define $\texttt{set-rep} : \bigcup_{i=1}^n X_i \to \mathcal{F}_p$ as an injective function by arbitrarily assigning a set in $\mathcal{F}_p$ to a vertex $x_i^\ell$ or $\neg x_i^\ell$, for every $i \in [n]$ and $\ell \in [2]$. In other words, every appearance of a literal is assigned a distinct subset in $\mathcal{F}_p$.
- The reduction adds a *connection portal* $V$, which is a clique on $2p$ vertices $v_1, v_2, \ldots, v_{2p}$. For every vertex $v_q$ in $V$, the reduction adds a pendant vertex $u_q$ adjacent to $v_q$.
- For each vertex $x_i^\ell \in X$ where $i \in [n]$ and $\ell \in [2]$, the reduction adds edges $(x_i^\ell, v_q)$ for every $q \in \texttt{set-rep}(x_i^\ell)$. Similarly, it adds edges $(\neg x_i^\ell, v_q)$ for every $q \in \texttt{set-rep}(\neg x_i^\ell)$.
- For a clause $C_j$, suppose variable $x_i$ appears positively for the $\ell^{th}$ time as the $r^{th}$ variable in $C_j$. For example, $x_i$ appears positively for the second time as the third variable in $C_j$. Then, the reduction adds edges across $B$ and $V$ such that the vertices $c_j^r$ and $x_i^\ell$ have the same neighbourhood in $V$, namely, the set $\{v_q : q \in \texttt{set-rep}(x_i^\ell)\}$. Similarly, it adds edges for the negative appearance of the variables.

This concludes the construction of $G$. The reduction sets $k = 4n + 3m + 2p$ and returns $(G, k)$ as the reduced instance of Locating-Dominating Set.

We now provide an overview of the proof of correctness in the reverse direction. The crux of the correctness is: Without loss of generality, all the vertices in the connection portal $V$ are present in any locating-dominating set $S$ of $G$. Consider a vertex, say $x_i^1$, on the "variable-side" of $S$ and a vertex, say $c_j^1$, on the "clause-side" of $S$. If both of these vertices have the same neighbors in the connection portal and are not adjacent to the vertices in $S \setminus V$, then at least one of $x_i^1$ or $c_j^1$ must be included in $S$.

More formally, suppose $S$ is a locating-dominating set of $G$ of size at most $k = 4n + 3m + 2p$. Then, we prove that $S$ must have exactly 4 vertices from each variable gadget and exactly 3 vertices from each clause gadget. Further, $S$ contains either $\{\alpha_0^i, \beta_0^i, x_i^1, x_i^2\}$ or $\{\alpha_0^i, \beta_0^i, \neg x_i^1, \neg x_i^2\}$, but no other combination of vertices in the variable gadget corresponding to $x_i$. For a clause gadget corresponding to $C_j$, $S$ contains either $\{\gamma_j^0, c_j^2, c_j^3\}$, $\{\gamma_j^0, c_j^1, c_j^3\}$, or $\{\gamma_j^0, c_j^1, c_j^2\}$, but no other combination. These choices imply that $c_j^1$, $c_j^2$, or $c_j^3$ are not adjacent to any vertex in $S \setminus V$. Consider the first case and suppose $c_j^1$ corresponds to the second positive appearance of variable $x_i$. By the construction, the neighborhoods of $x_i^2$ and $c_j^1$ in $V$ are identical. This forces a selection of $\{\alpha_0^i, \beta_0^i, x_i^1, x_i^2\}$ in $S$ from the variable gadget corresponding to $x_i$, which corresponding to setting $x_i$ to `True`. Hence, a locating dominating set $S$ of size at most $k$ implies a satisfying assignment of $\psi$.

**Sketch of Proof of Theorem 1.** Note that since each component of $G - V$ is of constant order, the tree-width of $G$ is $\mathcal{O}(|V|)$. By the asymptotic estimation of the central binomial coefficient, $\binom{2p}{p} \sim \frac{4^p}{\sqrt{\pi \cdot p}}$ [2]. To get the upper bound of $2p$, we scale down the asymptotic function and have $4n \leq \frac{4^p}{2^p} = 2^p$. Since we choose the value of $p$ as small as possible such that $2^p \geq 4n$, we choose $p = \log(n) + 3$. This ensures us that $p = \mathcal{O}(\log(n))$. And hence, $|V| = \mathcal{O}(\log(n))$ which implies $\mathsf{tw}(G) = \mathcal{O}(\log n)$. The remaining arguments are standard for proving the conditional lower bound under ETH. ◀

## 4 Locating-Dominating Set Parameterized by the Solution Size

In this section, we study the parameterized complexity of Locating-Dominating Set when parameterized by the solution size $k$. First, we formally prove that the problem admits a kernel with $\mathcal{O}(2^k)$ vertices, and hence a simple FPT algorithm running in time $2^{\mathcal{O}(k^2)}$. Next, we prove that both results mentioned above are optimal under the ETH.

▶ **Proposition 7.** *Locating-Dominating Set admits a kernel with $\mathcal{O}(2^k)$ vertices and an algorithm running in time $2^{\mathcal{O}(k^2)} + \mathcal{O}(k \log n)$.*

**Proof.** Slater proved that for any graph $G$ on $n$ vertices with a locating-dominating set of size $k$, we have $n \leq 2^k + k - 1$ [64]. Hence, if $n > 2^k + k - 1$, we can return a trivial No instance (this check takes time $\mathcal{O}(k \log n)$). Otherwise, we have a kernel with $\mathcal{O}(2^k)$ vertices. In this case, we can enumerate all subsets of vertices of size $k$, and for each of them, check in quadratic time if it is a valid solution. Overall, this takes time $\binom{n}{k} n^2$; since $n \leq 2^k + k - 1$, this is $\binom{2^{\mathcal{O}(k)}}{k} \cdot 2^{\mathcal{O}(k)}$, which is $2^{\mathcal{O}(k^2)}$. ◀

To prove Theorem 2, we present a reduction that takes as input an instance $\psi$, with $n$ variables, of 3-SAT and returns an instance $(G, k)$ of Locating-Dominating Set such that $|V(G)| = 2^{\mathcal{O}(\sqrt{n})}$ and $k = \mathcal{O}(\sqrt{n})$. By adding dummy variables in each set, we can assume that $\sqrt{n}$ is an even integer. Suppose the variables are named $x_{i,j}$ for $i, j \in [\sqrt{n}]$. The reduction constructs graph $G$ as follows:

- It partitions the variables of $\psi$ into $\sqrt{n}$ many *buckets* $X_1, X_2, \ldots, X_{\sqrt{n}}$ such that each bucket contains exactly $\sqrt{n}$ many variables. Let $X_i = \{x_{i,j} \mid j \in [\sqrt{n}]\}$ for all $i \in [\sqrt{n}]$.

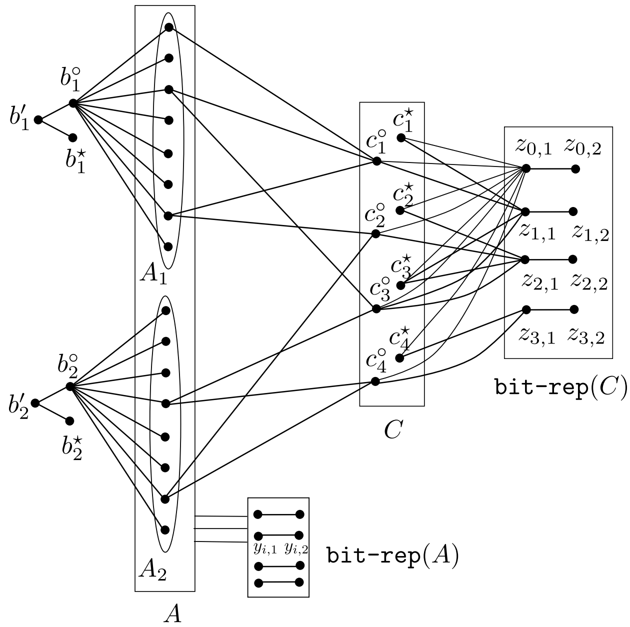


**Figure 2** Suppose an instance $\psi$ of 3-SAT has $n = 9$ variables and 4 clauses. We do not show the third variable bucket and explicit edges across $A$ and $\texttt{bit-rep}(A)$ for brevity.

- For every $X_i$, it constructs set $A_i$ of $2^{\sqrt{n}}$ new vertices, $A_i = \{a_{i,\ell} \mid \ell \in [2^{\sqrt{n}}]\}$. Each vertex in $A_i$ corresponds to a unique assignment of variables in $X_i$. Let $A$ be the collection of all the vertices added in this step.
- For every $X_i$, the reduction adds a path on three vertices $b_i^\circ$, $b_i'$, and $b_i^\star$ with edges $(b_i^\circ, b_i')$ and $(b_i', b_i^\star)$. Suppose $B$ is the collection of all the vertices added in this step.
- For every $X_i$, it makes $b_i^\circ$ adjacent with every vertex in $A_i$.

- For every clause $C_j$, the reduction adds a pair of vertices $c_j^\circ, c_j^\star$. For a vertex $a_{i,\ell} \in A_i$ for some $i \in [\sqrt{n}]$, and $\ell \in [2^{\sqrt{n}}]$, if the assignment corresponding to vertex $a_{i,\ell}$ satisfies clause $C_j$, then it adds edge $(a_{i,\ell}, c_j^\circ)$.
- The reduction adds a bit-representation gadget[4] to locate set $A$. Once again, informally speaking, it adds some supplementary vertices such that it is safe to assume these vertices are present in a locating-dominating set, and they locate every vertex in $A$. More precisely:
  - First, set $q := \lceil \log(|A|) \rceil + 1$. This value for $q$ allows to uniquely represent each integer in $[|A|]$ by its bit-representation in binary (starting from 1 and not 0).
  - For every $i \in [q]$, the reduction adds two vertices $y_{i,1}$ and $y_{i,2}$ and edge $(y_{i,1}, y_{i,2})$.
  - For every integer $q' \in [|A|]$, let $\texttt{bit}(q')$ denote the binary representation of $q'$ using $q$ bits. Connect $a_{i,\ell} \in A$ with $y_{i,1}$ if the $i^{th}$ bit in $\texttt{bit}((i + (\ell - 1) \cdot \sqrt{n}))$ is 1.
  - It adds two vertices $y_{0,1}$ and $y_{0,2}$, and edge $(y_{0,1}, y_{0,2})$. It also makes every vertex in $A$ adjacent with $y_{0,1}$.

    Let $\texttt{bit-rep}(A)$ be the collection of the vertices $y_{i,1}$ for all $i \in \{0\} \cup [q]$ added in this step.
- Finally, the reduction adds a bit-representation gadget to locate set $C$. However, it adds the vertices in such a way that for any pair $c_j^\circ, c_j^\star$, the supplementary vertices adjacent to them are identical.

---

[4] With the problem-specific adaptations, the bit-representation gadgets resembles the gadget used in [29].

- The reduction sets $p := \lceil \log(|C|/2) \rceil + 1$ and for every $i \in [p]$, it adds two vertices $z_{i,1}$ and $z_{i,2}$ and edge $(z_{i,1}, z_{i,2})$.
- For every integer $j \in [|C|/2]$, let $\texttt{bit}(j)$ denote the binary representation of $j$ using $q$ bits. Connect $c_j^{\circ}, c_j^{\star} \in C$ with $z_{i,1}$ if the $i^{th}$ bit in $\texttt{bit}(j)$ is 1.
- It adds two vertices $z_{0,1}$ and $z_{0,2}$, and edge $(z_{0,1}, z_{0,2})$. It also makes every vertex in $C$ adjacent with $y_{0,1}$.
  Let $\texttt{bit-rep}(C)$ be the collection of the vertices $z_{i,1}$ for all $i \in \{0\} \cup [p]$ added in this step.

This completes the reduction. See Figure 2 for an illustration. Please check this. The reduction sets

$$k = |B|/3 + (\lceil \log(|A|) \rceil + 1 + 1) + \lceil (\log(|C|/2) \rceil + 1 + 1) + \sqrt{n} = \mathcal{O}(\sqrt{n})$$

as $|B| = 3\sqrt{n}$, $|A| = \sqrt{n} \cdot 2^{\sqrt{n}}$, and $|C| = \mathcal{O}(n^3)$, and returns $(G, k)$ as a reduced instance.

We present a brief overview of the proof of correctness in the reverse direction. Suppose $S$ is a locating-dominating set of graph $G$ of size at most $k$. Note that $b_i^{\star}$, $y_{i,2}$ and $z_{i,2}$ are pendant vertices for appropriate $i$. We argue that it is safe to consider that vertices $b'_i$, $y_{i,1}$, and $z_{i,1}$ are in $S$. This already forces $|B|/2 + \lceil \log(|A|) \rceil + 2 + \lceil \log(|C|/2) \rceil + 2$ many vertices in $S$. The remaining $\sqrt{n}$ many vertices need to locate vertices in pairs $(b_i^{\circ}, b_i^{\star})$ and $(c_j^{\circ}, c_j^{\star})$ for every $i \in [\sqrt{n}]$ and $j \in [|C|]$. Note that the only vertices that are adjacent with $b_i^{\circ}$ but are *not* adjacent with $b_i^{\star}$ are in $A_i$. Also, the only vertices that are adjacent with $c_j^{\circ}$ but are *not* adjacent with $c_j^{\star}$ are in $A_i$ and correspond to an assignment that satisfies $C_j$. Hence, any locating-dominating set should contain at least one vertex in $A_i$ (which will locate $b_i^{\circ}$ from $b_i^{\star}$) such that the union of these vertices resolves all pairs of the form $(c_j^{\circ}, c_j^{\star})$, and hence corresponds to a satisfying assignment of $\psi$.

**Proof of Theorem 2.** Assume there exists an algorithm, say $\mathcal{A}$, that takes as input an instance $(G, k)$ of Locating-Dominating Set and correctly concludes whether it is a Yes-instance in time $2^{o(k^2)} \cdot |V(G)|^{\mathcal{O}(1)}$. Consider algorithm $\mathcal{B}$ that takes as input an instance $\psi$ of 3-SAT, uses the reduction above to get an equivalent instance $(G, k)$ of Locating-Dominating Set, and then uses $\mathcal{A}$ as a subroutine. The correctness of algorithm $\mathcal{B}$ follows from the correctness of the reduction and of algorithm $\mathcal{A}$. From the description of the reduction and the fact that $k = \sqrt{n}$, the running time of algorithm $\mathcal{B}$ is $2^{\mathcal{O}(\sqrt{n})} + 2^{o(k^2)} \cdot (2^{\mathcal{O}(\sqrt{n})})^{\mathcal{O}(1)} = 2^{o(n)}$. This, however, contradicts the ETH. Hence, Locating-Dominating Set does not admit an algorithm with running time $2^{o(k^2)} \cdot |V(G)|^{\mathcal{O}(1)}$ unless the ETH fails.

For the second part of Theorem 2, assume that such a kernelization algorithm exists. Consider the following algorithm for 3-SAT. Given a 3-SAT formula on $n$ variables, it uses the above reduction to get an equivalent instance of $(G, k)$ such that $|V(G)| = 2^{\mathcal{O}(\sqrt{n})}$ and $k = \mathcal{O}(\sqrt{n})$. Then, it uses the assumed kernelization algorithm to construct an equivalent instance $(H, k')$ such that $H$ has $2^{o(k)}$ vertices and $k' \leq k$. Finally, it uses a brute-force algorithm, running in time $|V(H)|^{\mathcal{O}(k')}$, to determine whether the reduced instance, equivalently the input instance of 3-SAT, is a Yes-instance. The correctness of the algorithm follows from the correctness of the respective algorithms and our assumption. The total running time of the algorithm is $2^{\mathcal{O}(\sqrt{n})} + (|V(G)| + k)^{\mathcal{O}(1)} + |V(H)|^{\mathcal{O}(k')} = 2^{\mathcal{O}(\sqrt{n})} + (2^{\mathcal{O}(\sqrt{n})})^{\mathcal{O}(1)} + (2^{o(\sqrt{n})})^{\mathcal{O}(\sqrt{n})} = 2^{o(n)}$. This, however, contradicts the ETH. ◀

## 5 Test Cover Parameterization by the Solution Size

In this section, we present a reduction that is very close to the reduction used in the proof of Theorem 2 to prove Theorem 3.

For notational convenience, instead of specifying an instance of Test Cover, we specify the auxiliary graph as mentioned in the definition. The reduction takes as input an instance $\psi$, with $n$ variables and $m$ clauses, of 3-SAT and returns a reduced instance $(G, \langle R, B\rangle, k)$ of Test Cover and $k = \mathcal{O}(\log(n) + \log(m)) = \mathcal{O}(\log(n))$, using the sparcification lemma [44]. The reduction constructs graph $G$ as follows.

- The reduction adds some dummy variables to ensure that $n = 2^{2q}$ for some positive integer $q$ which is power of 2. This ensures that $r = \log_2(n) = 2q$ and $s = \frac{n}{r}$ both are integers. It partitions the variables of $\psi$ into $r$ many *buckets* $X_1, X_2, \ldots, X_r$ such that each bucket contains $s$ many variables. Let $X_i = \{x_{i,j} \mid j \in [s]\}$ for all $i \in [r]$.
- For every $X_i$, the reduction constructs a set $A_i$ of $2^s$ many red vertices, that is, $A_i = \{a_{i,\ell} \mid \ell \in [2^s]\}$. Each vertex in $A_i$ corresponds to a unique assignment of the variables in $X_i$. Moreover, let $A = \cup_{i=1}^{r} A_i$.
- Corresponding to each $X_i$, let the reduction add a blue vertex $b_i$ and the edges $(b_i, a_{i,\ell})$ for all $i \in [r]$ and $\ell \in [2^s]$. Let $B = \{b_i \mid i \in [r]\}$.
- For every clause $C_j$, the reduction adds a pair of blue vertices $c_j^{\circ}, c_j^{\star}$. For a vertex $a_{i,\ell} \in A_i$ with $i \in [r]$, and $\ell \in [2^s]$, if the assignment corresponding to vertex $a_{i,\ell}$ satisfies the clause $C_j$, then the reduction adds the edge $(a_{i,\ell}, c_j^{\circ})$. Let $C = \{c_j^{\circ}, c_j^{\star} \mid j \in [m]\}$.
- The reduction adds a bit-representation gadget to locate set $C$. However, it adds the vertices in such a way that for any pair $c_j^{\circ}, c_j^{\star}$, the supplementary vertices adjacent to them are identical.
  - The reduction sets $p := \lceil \log(m) \rceil + 1$ and for every $i \in [p]$, it adds two vertices, a red vertex $z_{i,1}$ and a blue vertex $z_{i,2}$, and edge $(z_{i,1}, z_{i,2})$.
  - For every integer $j \in [m]$, let $\texttt{bit}(j)$ denote the binary representation of $j$ using $p$ bits. Connect $c_j^{\circ}, c_j^{\star} \in C$ with $z_{i,1}$ if the $i^{th}$ bit in $\texttt{bit}(j)$ is 1.
  - It add two vertices $z_{0,1}$ and $z_{0,2}$, and edge $(z_{0,1}, z_{0,2})$. It also makes every vertex in $C$ adjacent with $z_{0,1}$. Let $\texttt{bit-rep}(C)$ be the collection of all the vertices added in this step.
- The reduction adds an isolated blue vertex $b_0$.

This completes the construction. The reduction sets $k = r + \frac{1}{2}|\texttt{bit-rep}(C)| = \mathcal{O}(\log(n)) + \mathcal{O}(\log(m)) = \mathcal{O}(\log(n))$, and returns $(G, \langle R, B\rangle, k)$ as an instance of Test Cover. We refer to Figure 3 for an illustration.

We present a brief overview of the proof of correctness for the backward direction ($\Leftarrow$). Suppose $R'$ is a set of tests of the graph $G$ of order at most $k$. Since $b_0$ is an isolated blue vertex of $G$, it implies that the set $R'$ dominates and locates every pair of vertices in $B \setminus \{b_0\}$. The blue vertices in $\texttt{bit-rep}(C)$ are pendant vertices that are adjacent with red vertices in $\texttt{bit-rep}(C)$. Hence, all the red vertices in $\texttt{bit-rep}(C)$ are in $R'$. The remaining $r$ many vertices need to locate vertices in pairs $(c_j^{\circ}, c_j^{\star})$, for every $j \in [m]$, which have the same neighbourhood in $\texttt{bit-rep}(C)$. To do so, note that the only vertices adjacent to $c_j^{\circ}$ and *not* to $c_j^{\star}$ are in $A_i$ and corresponds to an assignment satisfying clause $C_j$. Hence, for every $j \in [m]$, the set $R'$ should contain at least one vertex $\{a_{i,\ell}\}$ in order to locate $c_j^{\circ}, c_j^{\star}$, where $(a_{i,\ell}, c_j^{\circ})$ is an edge for some $i \in [r]$ and $\ell \in [2^s]$. Moreover, in order to dominate the vertices of $B$, for each $i \in [r]$, the set $R'$ must have a vertex from each $A_i$. Hence, the set $R'$ is forced to contain *exactly* one vertex from each $A_i$. Concatenating the assignments corresponding to each $a_{i,\ell}$ in $R'$, we thus obtain a satisfying assignment of $\psi$. Proof of Theorem 3 follows from the arguments that are standard to proving such lower bounds.

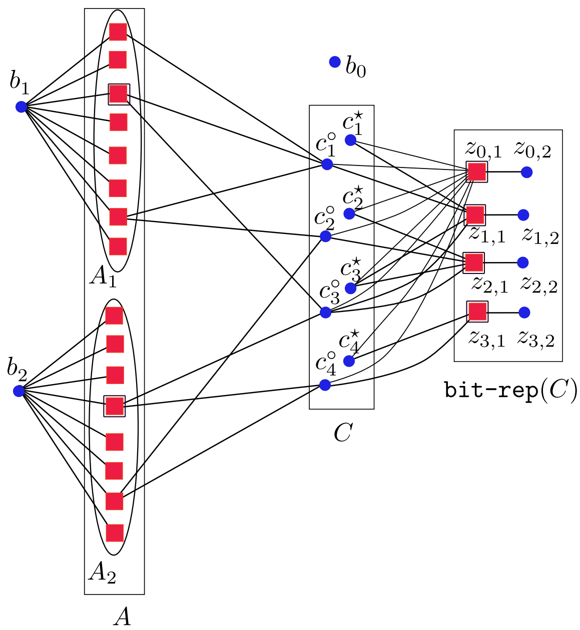


**Figure 3** An illustrative example of the graph constructed by the reduction in Section 5. Red (squared) nodes denote the tests whereas blue (filled circle) nodes the elements.

## 6 Conclusion

We presented several results that advance our understanding of the algorithmic complexity of Locating-Dominating Set and Test Cover, which we showed to have very interesting and rare parameterized complexities. Moreover, we believe the techniques used in this article can be applied to other identification problems to obtain relatively rare conditional lower bounds. The process of establishing such lower bounds boils down to designing *bit-representation gadgets* and *set-representation gadgets* for the problem in question.

Apart from the broad question of designing such lower bounds for other identification problems, we mention an interesting problem left open by our work. Can our tight double-exponential lower bound for Locating-Dominating Set parameterized by treewidth/treedepth be applied to the feedback vertex set number? The question could also be studied for other related parameters.

## References


1 Akanksha Agrawal, Daniel Lokshtanov, Saket Saurabh, and Meirav Zehavi. Split contraction: The untold story. *ACM Trans. Comput. Theory*, 11(3):18:1–18:22, 2019. doi:10.1145/3319909.

2 Ian Anderson. *Combinatorics of Finite Sets*. Oxford University Press, 1987.

3 Gabriela R. Argiroffo, Silvia M. Bianchi, Yanina Lucarini, and Annegret Katrin Wagler. Linear-time algorithms for three domination-based separation problems in block graphs. *Discret. Appl. Math.*, 281:6–41, 2020. doi:10.1016/J.DAM.2019.08.001.

4 László Babai. On the complexity of canonical labelling of strongly regular graphs. *SIAM J. Comput.*, 9(1):212–216, 1980. doi:10.1137/0209018.

5 Florian Barbero, Lucas Isenmann, and Jocelyn Thiebaut. On the distance identifying set meta-problem and applications to the complexity of identifying problems on graphs. *Algorithmica*, 82(8):2243–2266, 2020. doi:10.1007/S00453-020-00674-X.

6 Manu Basavaraju, Mathew C. Francis, M. S. Ramanujan, and Saket Saurabh. Partially polynomial kernels for set cover and test cover. *SIAM J. Discret. Math.*, 30(3):1401–1423, 2016. doi:10.1137/15M1039584.

7 Piotr Berman, Bhaskar DasGupta, and Ming-Yang Kao. Tight approximability results for test set problems in bioinformatics. *J. Comput. Syst. Sci.*, 71(2):145–162, 2005. doi:10.1016/J.JCSS.2005.02.001.

8 Ivan Bliznets and Markus Hecher. Tight double exponential lower bounds. In Xujin Chen and Bo Li, editors, *Theory and Applications of Models of Computation - 18th Annual Conference, TAMC 2024, Hong Kong, China, May 13-15, 2024, Proceedings*, volume 14637 of *Lecture Notes in Computer Science*, pages 124–136. Springer, 2024. doi:10.1007/978-981-97-2340-9_11.

9 John A Bondy. Induced subsets. *Journal of Combinatorial Theory, Series B*, 12(2):201–202, 1972.

10 Koen M. J. De Bontridder, Bjarni V. Halldórsson, Magnús M. Halldórsson, Cor A. J. Hurkens, Jan Karel Lenstra, R. Ravi, and Leen Stougie. Approximation algorithms for the test cover problem. *Math. Program.*, 98(1-3):477–491, 2003. doi:10.1007/S10107-003-0414-6.

11 Márcia R. Cappelle, Guilherme de C. M. Gomes, and Vinícius Fernandes dos Santos. Parameterized algorithms for locating-dominating sets. *CoRR*, abs/2011.14849, 2020. arXiv:2011.14849.

12 Márcia R. Cappelle, Guilherme C. M. Gomes, and Vinícius Fernandes dos Santos. Parameterized algorithms for locating-dominating sets. In Carlos E. Ferreira, Orlando Lee, and Flávio Keidi Miyazawa, editors, *Proceedings of the XI Latin and American Algorithms, Graphs and Optimization Symposium, LAGOS 2021, Online Event / São Paulo, Brazil, May 2021*, volume 195 of *Procedia Computer Science*, pages 68–76. Elsevier, 2021. doi:10.1016/J.PROCS.2021.11.012.

13 Dipayan Chakraborty, Anni Hakanen, and Tuomo Lehtilä. The $n/2$-bound for locating-dominating sets in subcubic graphs, 2024. arXiv:2406.19278.

14 Jérémie Chalopin, Victor Chepoi, Fionn Mc Inerney, and Sébastien Ratel. Non-clashing teaching maps for balls in graphs. *CoRR*, abs/2309.02876, 2023. doi:10.48550/arXiv.2309.02876.

15 L. Sunil Chandran, Davis Issac, and Anreas Karrenbauer. On the parameterized complexity of biclique cover and partition. In Jiong Guo and Danny Hermelin, editors, *11th International Symposium on Parameterized and Exact Computation, IPEC 2016*, volume 63 of *LIPIcs*, pages 11:1–11:13. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPICS.IPEC.2016.11.

16 Emmanuel Charbit, Irène Charon, Gérard D. Cohen, Olivier Hudry, and Antoine Lobstein. Discriminating codes in bipartite graphs: bounds, extremal cardinalities, complexity. *Advances in Mathematics of Communication*, 2(4):403–420, 2008. doi:10.3934/AMC.2008.2.403.

17 Irène Charon, Gérard D. Cohen, Olivier Hudry, and Antoine Lobstein. Discriminating codes in (bipartite) planar graphs. *Eur. J. Comb.*, 29(5):1353–1364, 2008. doi:10.1016/J.EJC.2007.05.006.

18 Bogdan S. Chlebus and Sinh Hoa Nguyen. On finding optimal discretizations for two attributes. In *Proceedings of the First International Conference on Rough Sets and Current Trends in Computing*, volume 1424, pages 537–544, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg. doi:10.1007/3-540-69115-4_74.

19 Vasek Chvátal. Mastermind. *Combinatorica*, 3(3):325–329, 1983. doi:10.1007/BF02579188.

20 C. Colbourn, P. J. Slater, and L. K. Stewart. Locating-dominating sets in series-parallel networks. *Congressus Numerantium*, 56:135–162, 1987.

21 Bruno Courcelle. The monadic second-order logic of graphs. I. recognizable sets of finite graphs. *Inf. Comput.*, 85(1):12–75, 1990. doi:10.1016/0890-5401(90)90043-H.

22 Bruno Courcelle. The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. *Information and Computation*, 85(1):12–75, 1990. doi:10.1016/0890-5401(90)90043-H.

23 Bruno Courcelle, Johann A. Makowsky, and Udi Rotics. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory Comput. Syst.*, 33(2):125–150, 2000. doi:10.1007/S002249910009.

24 Robert Crowston, Gregory Z. Gutin, Mark Jones, Gabriele Muciaccia, and Anders Yeo. Parameterizations of test cover with bounded test sizes. *Algorithmica*, 74(1):367–384, 2016. doi:10.1007/S00453-014-9948-7.

25 Robert Crowston, Gregory Z. Gutin, Mark Jones, Saket Saurabh, and Anders Yeo. Parameterized study of the test cover problem. In Branislav Rovan, Vladimiro Sassone, and Peter Widmayer, editors, *Mathematical Foundations of Computer Science 2012 - 37th International Symposium, MFCS 2012, Bratislava, Slovakia, August 27-31, 2012. Proceedings*, volume 7464 of *Lecture Notes in Computer Science*, pages 283–295. Springer, 2012. doi:10.1007/978-3-642-32589-2_27.

26 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.

27 Marek Cygan, Marcin Pilipczuk, and Michal Pilipczuk. Known algorithms for edge clique cover are probably optimal. *SIAM J. Comput.*, 45(1):67–83, 2016. doi:10.1137/130947076.

28 Reinhard Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012. URL: https://dblp.org/rec/books/daglib/0030488.bib.

29 Michael Dom, Daniel Lokshtanov, and Saket Saurabh. Incompressibility through colors and ids. In Susanne Albers, Alberto Marchetti-Spaccamela, Yossi Matias, Sotiris E. Nikoletseas, and Wolfgang Thomas, editors, *Automata, Languages and Programming, 36th International Colloquium, ICALP 2009, Rhodes, Greece, July 5-12, 2009, Proceedings, Part I*, volume 5555 of *Lecture Notes in Computer Science*, pages 378–389. Springer, 2009. doi:10.1007/978-3-642-02927-1_32.

30 Johannes Klaus Fichte, Markus Hecher, Michael Morak, Patrick Thier, and Stefan Woltran. Solving projected model counting by utilizing treewidth and its limits. *Artif. Intell.*, 314:103810, 2023. doi:10.1016/J.ARTINT.2022.103810.

31 Johannes Klaus Fichte, Markus Hecher, Michael Morak, and Stefan Woltran. Exploiting treewidth for projected model counting and its limits. In *Theory and Applications of Satisfiability Testing - SAT 2018 - 21st International Conference, SAT 2018, Proc.*, volume 10929 of *Lecture Notes in Computer Science*, pages 165–184. Springer, 2018. doi:10.1007/978-3-319-94144-8_11.

32 Jacob Focke, Fabian Frei, Shaohua Li, Dániel Marx, Philipp Schepper, Roohani Sharma, and Karol Wegrzycki. Hitting meets packing: How hard can it be? In *32nd Annual European Symposium on Algorithms, ESA 2024, September 2-4, 2024, Royal Holloway, London, United Kingdom*, volume 308 of *LIPIcs*, pages 55:1–55:21. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPICS.ESA.2024.55.

33 Fedor V. Fomin, Petr A. Golovach, Daniel Lokshtanov, Saket Saurabh, and Meirav Zehavi. Clique-width III: hamiltonian cycle and the odd case of graph coloring. *ACM Trans. Algorithms*, 15(1):9:1–9:27, 2019. doi:10.1145/3280824.

34 Fedor V Fomin, Daniel Lokshtanov, Saket Saurabh, and Meirav Zehavi. *Kernelization: theory of parameterized preprocessing*. Cambridge University Press, 2019.

35 Florent Foucaud. Decision and approximation complexity for identifying codes and locating-dominating sets in restricted graph classes. *J. Discrete Algorithms*, 31:48–68, 2015. doi:10.1016/J.JDA.2014.08.004.

36 Florent Foucaud, Esther Galby, Liana Khazaliya, Shaohua Li, Fionn Mc Inerney, Roohani Sharma, and Prafullkumar Tale. Problems in NP can admit double-exponential lower bounds when parameterized by treewidth or vertex cover. In Karl Bringmann, Martin Grohe, Gabriele Puppis, and Ola Svensson, editors, *51st International Colloquium on Automata, Languages, and Programming, ICALP 2024, July 8-12, 2024, Tallinn, Estonia*, volume 297 of *LIPIcs*, pages 66:1–66:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPICS.ICALP.2024.66.

37 Florent Foucaud, George B. Mertzios, Reza Naserasr, Aline Parreau, and Petru Valicov. Identification, location-domination and metric dimension on interval and permutation graphs. I. bounds. *Theoretical Computer Science*, 668:43–58, 2017. doi:10.1016/J.TCS.2017.01.006.

38 Florent Foucaud, George B Mertzios, Reza Naserasr, Aline Parreau, and Petru Valicov. Identification, location-domination and metric dimension on interval and permutation graphs. II. algorithms and complexity. *Algorithmica*, 78(3):914–944, 2017. doi:10.1007/S00453-016-0184-1.

39 M. R. Garey and David S. Johnson. *Computers and Intractability – A guide to NP-completeness*. W.H. Freeman and Company, 1979.

40 Sylvain Gravier, Ralf Klasing, and Julien Moncel. Hardness results and approximation algorithms for identifying codes and locating-dominating codes in graphs. *Algorithmic Oper. Res.*, 3(1), 2008. URL: http://journals.hil.unb.ca/index.php/AOR/article/view/2808.

41 Gregory Z. Gutin, Gabriele Muciaccia, and Anders Yeo. (non-)existence of polynomial kernels for the test cover problem. *Inf. Process. Lett.*, 113(4):123–126, 2013. doi:10.1016/J.IPL.2012.12.008.

42 Tesshu Hanaka, Noleen Köhler, and Michael Lampis. Core stability in additively separable hedonic games of low treewidth, 2024. arXiv:2402.10815, doi:10.48550/arXiv.2402.10815.

43 Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-sat. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001. doi:10.1006/JCSS.2000.1727.

44 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.

45 Klaus Jansen, Kim-Manuel Klein, and Alexandra Lassota. The double exponential runtime is tight for 2-stage stochastic ILPs. *Math. Program.*, 197:1145–1172, 2023. doi:10.1007/S10107-022-01837-0.

46 D. Jean and A Lobstein. Watching systems, identifying, locating-dominating and discriminating codes in graphs: a bibliography, 2024. Published electronically at https://dragazo.github.io/bibdom/main.pdf.

47 Dusan Knop, Michal Pilipczuk, and Marcin Wrochna. Tight complexity lower bounds for integer linear programming with few constraints. *ACM Trans. Comput. Theory*, 12(3):19:1–19:19, 2020. doi:10.1145/3397484.

48 Lukasz Kowalik, Alexandra Lassota, Konrad Majewski, Michal Pilipczuk, and Marek Sokolowski. Detecting points in integer cones of polytopes is double-exponentially hard. In *2024 Symposium on Simplicity in Algorithms (SOSA)*, pages 279–285, 2024. doi:10.1137/1.9781611977936.25.

49 Stefan Kratsch, Geevarghese Philip, and Saurabh Ray. Point line cover: The easy kernel is essentially tight. *ACM Trans. Algorithms*, 12(3):40:1–40:16, 2016. doi:10.1145/2832912.

50 M. Künnemann, F. Mazowiecki, L. Schütze, H. Sinclair-Banks, and K. Węgrzycki. Coverability in VASS Revisited: Improving Rackoff's Bound to Obtain Conditional Optimality. In *50th International Colloquium on Automata, Languages, and Programming (ICALP 2023)*, volume 261 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 131:1–131:20, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPICS.ICALP.2023.131.

51 Michael Lampis, Stefan Mengel, and Valia Mitsou. QBF as an alternative to Courcelle's theorem. In *Theory and Applications of Satisfiability Testing - SAT 2018 - 21st International Conference, SAT 2018*, volume 10929 of *Lecture Notes in Computer Science*, pages 235–252. Springer, 2018. doi:10.1007/978-3-319-94144-8_15.

52 Daniel Lokshtanov, Dániel Marx, and Saket Saurabh. Known algorithms on graphs of bounded treewidth are probably optimal. *ACM Trans. Algorithms*, 14(2):13:1–13:30, 2018. doi:10.1145/3170442.

53 Daniel Lokshtanov, Saket Saurabh, Subhash Suri, and Jie Xue. An ETH-tight algorithm for multi-team formation. In *41st IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2021*, volume 213 of *LIPIcs*, pages 28:1–28:9. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPICS.FSTTCS.2021.28.

54 Dániel Marx and Valia Mitsou. Double-exponential and triple-exponential bounds for choosability problems parameterized by treewidth. In *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016)*, volume 55 of *LIPIcs*, pages 28:1–28:15, 2016. doi:10.4230/LIPICS.ICALP.2016.28.

55 Bernard M. E. Moret and Henry D. Shapiro. On minimizing a set of tests. *SIAM Journal on Scientific and Statistical Computing*, 6(4):983–1003, 1985.

56 Tobias Müller and Jean-Sébastien Sereni. Identifying and locating-dominating codes in (random) geometric networks. *Comb. Probab. Comput.*, 18(6):925–952, 2009. doi:10.1017/S0963548309990344.

57 Marcin Pilipczuk and Manuel Sorge. A double exponential lower bound for the distinct vectors problem. *Discret. Math. Theor. Comput. Sci.*, 22(4), 2020. doi:10.23638/DMTCS-22-4-7.

58 Michal Pilipczuk. Problems parameterized by treewidth tractable in single exponential time: A logical approach. In *Mathematical Foundations of Computer Science 2011 - 36th International Symposium, MFCS 2011, Proceedings*, volume 6907 of *Lecture Notes in Computer Science*, pages 520–531. Springer, 2011. doi:10.1007/978-3-642-22993-0_47.

59 N.S.V. Rao. Computational complexity issues in operative diagnosis of graph-based systems. *IEEE Transactions on Computers*, 42(4):447–457, 1993. doi:10.1109/12.214691.

60 Alfred Rényi. On random generating elements of a finite boolean algebra. *Acta Scientiarum Mathematicarum Szeged*, 22:75–81, 1961.

61 Ignasi Sau and Uéverton dos Santos Souza. Hitting forbidden induced subgraphs on bounded treewidth graphs. *Inf. Comput.*, 281:104812, 2021. doi:10.1016/J.IC.2021.104812.

62 András Sebö and Eric Tannier. On metric generators of graphs. *Mathematics of Operations Research*, 29(2):383–393, 2004. doi:10.1287/MOOR.1030.0070.

63 Peter J. Slater. Domination and location in acyclic graphs. *Networks*, 17(1):55–64, 1987. doi:10.1002/net.3230170105.

64 Peter J. Slater. Dominating and reference sets in a graph. *Journal of Mathematical and Physical Sciences*, 22(4):445–455, 1988.

65 Jukka Suomela. Approximability of identifying codes and locating-dominating codes. *Inf. Process. Lett.*, 103(1):28–33, 2007. doi:10.1016/J.IPL.2007.02.001.

66 P. Tale. Double exponential lower bound for telephone broadcast, 2024. arXiv:2403.03501, doi:10.48550/arXiv.2403.03501.

67 Johan M. M. van Rooij, Hans L. Bodlaender, and Peter Rossmanith. Dynamic programming on tree decompositions using generalised fast subset convolution. In Amos Fiat and Peter Sanders, editors, *Algorithms - ESA 2009, 17th Annual European Symposium, Copenhagen, Denmark, September 7-9, 2009. Proceedings*, volume 5757 of *Lecture Notes in Computer Science*, pages 566–577. Springer, 2009. doi:10.1007/978-3-642-04128-0_51.