On Problems as Hard as CNF-SAT*

Marek Cygan† Holger Dell‡ Daniel Lokshtanov§ Dániel Marx¶
Jesper Nederlof‖ Yoshio Okamoto** Ramamohan Paturi††
Saket Saurabh‡‡ Magnus Wahlström§§

June 7, 2012

Abstract

The field of exact exponential time algorithms for NP-hard problems has thrived over the last decade. While exhaustive search remains asymptotically the fastest known algorithm for some basic problems, difficult and non-trivial exponential time algorithms have been found for a myriad of problems, including Graph Coloring, Hamiltonian Path, Dominating Set and 3-CNF-Sat. In some instances, improving these algorithms further seems to be out of reach. The CNF-Sat problem is the canonical example of a problem for which the trivial exhaustive search algorithm runs in time $O(2^n)$, where n is the number of variables in the input formula. While there exist non-trivial algorithms for CNF-Sat that run in time $o(2^n)$, no algorithm was able to improve the *growth rate* 2 to a smaller constant, and hence it is natural to conjecture that 2 is the optimal growth rate. The *strong exponential time hypothesis* (SETH) by Impagliazzo and Paturi [JCSS 2001] goes a little bit further and asserts that, for every $\epsilon < 1$, there is a (large) integer k such that that k-CNF-Sat cannot be computed in time $2^{\epsilon n}$.

In this paper, we show that, for every $\epsilon < 1$, the problems Hitting Set, Set Splitting, and NAE-Sat cannot be computed in time $O(2^{\epsilon n})$ unless SETH fails. Here n is the number of elements or variables in the input. For these problems, we actually get an equivalence to SETH in a certain sense. We conjecture that SETH implies a similar statement for Set Cover, and prove that, under this assumption, the fastest known algorithms for Steiner Tree, Connected Vertex Cover, Set Partitioning, and the pseudo-polynomial time algorithm for Subset Sum cannot be significantly improved. Finally, we justify our assumption about the hardness of Set Cover by showing that the parity of the number of solutions to Set Cover cannot be computed in time $O(2^{\epsilon n})$ for any $\epsilon < 1$ unless SETH fails.

*An extended abstract of this paper appears in the proceedings of CCC 2012.

†IDSIA, University of Lugano, Switzerland. marek@idsia.ch. Partially supported by National Science Centre grant no. N206 567140, Foundation for Polish Science and ONR Young Investigator award when at the University at Maryland.

‡University of Wisconsin–Madison, USA. holger@cs.wisc.edu. Research partially supported by the Alexander von Humboldt Foundation and NSF grant 1017597.

§University of California, USA. dlokshtanov@ucsd.edu.

¶Computer and Automation Research Institute, Hungarian Academy of Sciences (MTA SZTAKI), Budapest, Hungary. dmarx@cs.bme.hu. Research supported by ERC Starting Grant PARAMTIGHT (280152).

‖Utrecht University, The Netherlands. j.nederlof@uu.nl. Supported by NWO project "Space and Time Efficient Structural Improvements of Dynamic Programming Algorithms".

**Japan Advanced Institute of Science and Technology, Japan. okamotoy@uec.ac.jp. Partially supported by Grant-in-Aid for Scientific Research from Japan Society for the Promotion of Science.

††University of California, USA. paturi@cs.ucsd.edu. This research is supported by NSF grant CCF-0947262 from the Division of Computing and Communication Foundations. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

‡‡Institute of Mathematical Sciences, India. saket@imsc.res.in.

§§Max-Planck-Institut für Informatik, Germany. wahl@mpi-inf.mpg.de.

1 Introduction

Every problem in NP can be solved in time $2^{\text{poly}(m)}$ by brute force, that is, by enumerating all candidates for an NP-witness, which is guaranteed to have length polynomial in the input size m. While we do not believe that polynomial time algorithms for NP-complete problems exist, many NP-complete problems have exponential time algorithms that are dramatically faster than the naïve brute force algorithm. For some classical problems, such as Subset Sum or Hamiltonian Cycle, such algorithms were known [HK62; Bel62] even before the concept of NP-completeness was discovered. Over the last decade, a subfield of algorithms devoted to developing faster exponential time algorithms for NP-hard problems has emerged. A myriad of problems have been shown to be solvable much faster than by naïve brute force, and a variety of algorithm design techniques for exponential time algorithms has been developed.

What the field of exponential time algorithms sorely lacks is a complexity-theoretic framework for showing running time lower bounds. Some problems, such as Independent Set and Dominating Set have seen a chain of improvements [FGK09; RND09; Rob86; KLR09], each new improvement being smaller than the previous. For these problems, the running time of the discovered algorithms seems to converge towards $O(C^n)$ for some unknown constant C, where n denotes the number of vertices of the input graphs. For other problems, such as Graph Coloring or Steiner Tree, non-trivial algorithms have been found, but improving the *growth rate* C of the running time any further seems to be out of reach [BHK09; Ned09]. The purpose of this paper is to develop tools that allow us to explain why we are stuck for these problems. Ideally, for any problem whose best known algorithm runs in time $O(C^n)$, we want to prove that the existence of $O(c^n)$-time algorithms for any constant $c < C$ would have implausible complexity-theoretic consequences.

Previous Work. Impagliazzo and Paturi's *Exponential Time Hypothesis* (ETH) addresses the question whether NP-hard problems can have algorithms that run in "subexponential time" [IP01]. More precisely, the hypothesis asserts that 3-CNF-Sat cannot be computed in time $2^{o(n)}$, where n is the number of variables in the input formula. ETH is considered to be a plausible complexity-theoretic assumption, and subexponential time algorithms have been ruled out under ETH for many decision problems [IPZ01], parameterized problems [CCF+05; LMS11], approximation problems [Mar07], and counting problems [DHM+12]. However, ETH does not seem to be sufficient for pinning down what exactly the best possible growth rate is. For this reason, we base our results on a stronger hypothesis.

The fastest known algorithms for CNF-Sat have running times of the form $2^{n-o(n)}\text{poly}(m)$ [Sch05; Wil11], which does not improve upon the growth rate 2 of the naïve brute force algorithm that runs in time $2^n\text{poly}(m)$. Hence a natural candidate for a stronger hypothesis is that CNF-Sat cannot be computed in time $2^{\epsilon n}\text{poly}(m)$ for any $\epsilon < 1$. However, we do not know whether our lower bounds on the growth rate of specific problems can be based on this hypothesis. The main technical obstacle is that we have no analogue of the sparsification lemma, which applies to k-CNF formulas and makes ETH a robust hypothesis [IPZ01]. In fact, very recent results indicate that such a sparsification may be impossible for general CNF formulas [SS11]. For this reason, we consider the *Strong Exponential Time Hypothesis* (SETH) of Impagliazzo and Paturi [IP01; IPZ01; CIP09]. This hypothesis asserts that, for every $\epsilon < 1$, there is a (large) integer k such that k-CNF-Sat cannot be computed by any bounded-error randomized algorithm in time $O(2^{\epsilon n})$. In particular, SETH implies the hypothesis for CNF-Sat above, but we do not know whether they are equivalent. Since SETH is a statement about k-CNF formulas for constant $k = k(\epsilon)$, we can apply the sparsification lemma for every fixed k, which allows us to use SETH as a starting point in our reductions.

Our results. Our first theorem is that SETH is equivalent to lower bounds on the time complexity of a number of standard NP-complete problems.

Theorem 1.1. *Each of the following statements is equivalent to SETH:*

(i) $\forall \epsilon < 1 . \exists k.$ k-CNF-Sat*, the satisfiability problem for n-variable k-CNF formulas, cannot be computed in time $O(2^{\epsilon n})$.*

(ii) $\forall \epsilon < 1 . \exists k.$ k-Hitting Set*, the hitting set problem for set systems over $[n]$ with sets of size at most k, cannot be computed in time $O(2^{\epsilon n})$.*

(iii) $\forall \epsilon < 1 . \exists k.$ k-Set Splitting*, the set splitting problem for set systems over $[n]$ with sets of size at most k, cannot be computed in time $O(2^{\epsilon n})$.*

(iv) $\forall \epsilon < 1 . \exists k.$ k-NAE-Sat*, the not-all-equal assignment problem for n-variable k-CNF formulas, cannot be computed in time $O(2^{\epsilon n})$.*

(v) $\forall \epsilon < 1 . \exists c.$ c-VSP-Circuit-SAT*, the satisfiability problem for n-variable series-parallel circuits of size at most cn, cannot be computed in time $O(2^{\epsilon n})$.*

For all of the above problems, the naïve brute force algorithm runs in time $O(2^n)$. While there may not be a consensus that SETH is a "plausible" complexity-theoretic assumption, our theorem does indicate that finding an algorithm for CNF-Sat whose growth rate is smaller than 2 is as difficult as finding such an algorithm for any of the above problems. Since our results are established via suitable reductions, this can be seen as a completeness result under these reductions. Moreover, we actually prove that the optimal growth rates for all of the problems above are *equal* as k tends to infinity. This gives an additional motivation to study the Strong Exponential Time Hypothesis.

An immediate consequence of Theorem 1.1 is that, if SETH holds, then CNF-Sat, Hitting Set, Set Splitting, NAE-Sat, and the satisfiability problem of series-parallel circuits do not have bounded-error randomized algorithms that run in time $2^{\epsilon n}\text{poly}(m)$ for any $\epsilon < 1$. All of these problems are *search* problems, where the objective is to find a particular object in a search space of size 2^n. Of course, we would also like to show tight connections between SETH and the optimal growth rates of problems that *do* have non-trivial exact algorithms. Our prototypical such problem is Set Cover: Given a set system with n elements and m sets, we want to select a given number t of sets that cover all elements. Exhaustively trying all possible ways to cover the elements takes time at most $2^m\text{poly}(m)$. However, m could be much larger than n, and it is natural to ask for the best running time that one can achieve in terms of n. It turns out that a simple dynamic programming algorithm [FKW04] can solve Set Cover in time $2^n\text{poly}(m)$. The natural question is whether the growth rate of this simple algorithm can be improved. While we are not able to resolve this question, we connect the existence of an improved algorithm for Set Cover to the existence of faster algorithms for several problems. Specifically, we show the following theorem.

Theorem 1.2. *Assume that, for all $\epsilon < 1$, there is a k such that* Set Cover *with sets of size at most k cannot be computed in time $2^{\epsilon n}\text{poly}(m)$. Then, for all $\epsilon < 1$, we have:*

(i) Steiner Tree *cannot be computed in time $2^{\epsilon t}\text{poly}(n)$,*

(ii) Connected Vertex Cover *cannot be computed in time $2^{\epsilon t}\text{poly}(n)$,*

(iii) Set Partitioning *cannot be computed in time $2^{\epsilon n}\text{poly}(m)$, and*

(iv) Subset Sum *cannot be computed in time $t^{\epsilon}\text{poly}(n)$.*

All problems mentioned in this theorem have non-trivial algorithms whose running times are as above with $\epsilon = 1$ [BHK+07; Ned09; CNP+11; FKW04; CLR+09]. Under the assumption in the theorem, we therefore obtain tight lower bounds on the growth rate of exact algorithms for Steiner Tree, Connected Vertex Cover, Set Partitioning, and Subset Sum. The best currently known algorithms for these problems share two interesting common features. First, they are all *dynamic programming* algorithms. Thus, Theorem 1.2 hints at Set Cover being a "canonical" dynamic programming problem. Second, the algorithms can all be modified to compute the number of solutions modulo two in the same running time. In fact, the currently fastest algorithm [CNP+11] for Connected Vertex Cover works by reducing the problem to computing the number of solutions modulo two.

While Theorem 1.1 is an equivalence, Theorem 1.2 is not. One might ask whether it is possible to find reductions back to Set Cover and to strengthen Theorem 1.2 in this manner. We believe that this would be quite difficult: A suitable reduction from, say, Steiner Tree to Set Cover that proves the converse of Theorem 1.2 would probably also work for $\epsilon = 1$. This would give an alternative proof that Steiner Tree can be computed in time $2^t\text{poly}(m)$. Hence, finding such a reduction is likely to be a challenge since the fastest known algorithms [BHK+07; Ned09] for Steiner Tree are quite non-trivial — it took more than 30 years before the classical $3^t\text{poly}(n)$-time Dreyfus–Wagner algorithm for Steiner Tree was improved to $2^t\text{poly}(n)$. Similar comments apply to Connected Vertex Cover since its $2^t\text{poly}(n)$ time algorithm is quite complex [CNP+11].

The hardness assumption for Set Cover in Theorem 1.2 needs some justification. Ideally we would like to replace this assumption with SETH, that is, we would like to prove that SETH implies the hardness assumption for Set Cover in Theorem 1.2. We do not know a suitable reduction, but we are able to provide a different kind of evidence for hardness: We show that a $2^{\epsilon n}\text{poly}(m)$-time algorithm to compute the number of set covers modulo two would violate $\oplus$-SETH, which is a hypothesis that implies SETH. Formally, $\oplus$-SETH asserts that, for all $\epsilon < 1$, there exists a (large) integer k such that k-$\oplus$CNF-Sat cannot be computed in time $O(2^{\epsilon n})$. Here, k-$\oplus$CNF-Sat is the problem of computing the number of satisfying assignments of a given k-CNF formula modulo two. It follows from known results [CIK+03; Tra08] (see also Section 3.1) that, if SETH holds, then so does $\oplus$-SETH. As a partial justification for the hardness assumption for Set Cover in Theorem 1.2, we provide the following theorem.

Theorem 1.3. *Each of the following statements is equivalent to* $\oplus$-SETH*:*

(i) $\forall\epsilon < 1.\exists k.$ k-$\oplus$CNF-Sat*, the parity satisfiability problem for n-variable k-CNF formulas, cannot be computed in time $O(2^{\epsilon n})$.*

(ii) $\forall\epsilon < 1.\exists k.$ k-$\oplus$Hitting Set*, the parity hitting set problem for set systems over $[n]$ with sets of size at most k, cannot be computed in time $O(2^{\epsilon n})$.*

(iii) $\forall\epsilon < 1.\exists k.$ k-$\oplus$Set Cover*, the parity set cover problem for set systems over $[n]$ with sets of size at most k, cannot be computed in time $O(2^{\epsilon n})$.*

In the statement of Theorem 1.3, the $\oplus$Hitting Set and $\oplus$Set Cover problems are defined as follows: the input is a set system and the objective is to compute the parity of the number of hitting sets (resp. set covers) in the system. An immediate consequence of Theorem 1.3 that we find interesting is that $\oplus$-SETH rules out the existence of $2^{\epsilon n}\text{poly}(m)$-time algorithms to compute the number of set covers of a set system, for any $\epsilon < 1$.

Theorem 1.3 together with the fact that the algorithms for all problems mentioned in Theorem 1.2 can be modified to count solutions modulo two leads to the following questions: Can we show running time lower bounds for the counting versions of these problems? We show that this

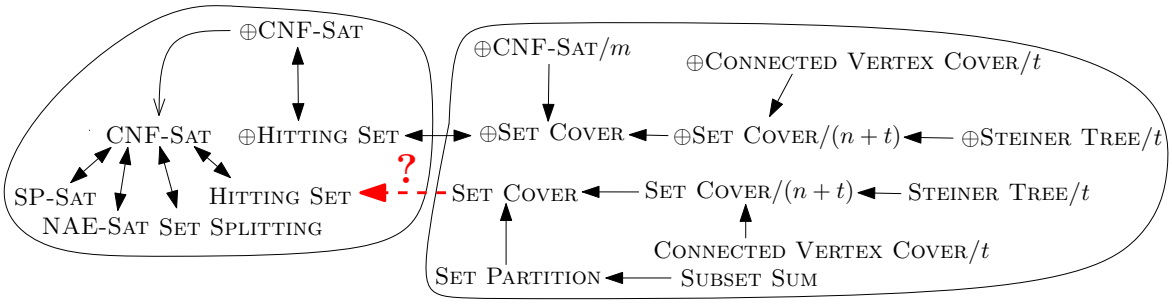

Figure 1: Overview of all reductions we give. An arrow $\Pi \to \Pi'$ depicts a reduction from the problem Π' to the problem Π. In other words, improving the best-known algorithm for Π implies that the best-known algorithm for Π' can be improved as well. The thin arrowhead indicates the isolation lemma result known from previous work [CIK+03; Tra08]. The left group contains problems, for which the best-known algorithm is naïve brute force, and is discussed in Section 3. The right group contains problems, for which the best-known algorithms are based on dynamic programming flavoured techniques, and is discussed in Section 4. The red and dashed arrow indicates the open problem whether SETH implies the assumption of Theorem 1.2.

is indeed possible. In particular we show that, assuming $\oplus$-SETH, there is no $2^{\epsilon t}\text{poly}(n)$-time algorithm that computes the parity of the number of Steiner trees that have size at most t, and no $2^{\epsilon t}\text{poly}(n)$-time algorithm that computes the parity of the number of connected vertex covers that have size at most t. Thus, unless $\oplus$-SETH fails, any improved algorithm for Set Cover, Steiner Tree, or Connected Vertex Cover cannot be used to compute the parity of the number of solutions.

We find it intriguing that SETH and $\oplus$-SETH can be used to show tight running time lower bounds, sometimes for problems for which the best algorithm has been improved several times, such as for Steiner Tree or Connected Vertex Cover. We feel that such sharp bounds are unlikely to just be a coincidence, leading us to conjecture that the relationship between the considered problems is even closer than what we show. Specifically, we conjecture that SETH implies the hardness assumption for Set Cover in Theorem 1.2. This conjecture provides an interesting open problem.

Our results are obtained by a collection of reductions. Section 3 contains the reductions that constitute the proof of Theorem 1.1, and some of the reductions needed for Theorem 1.3. Section 4 contains the proof of Theorem 1.2, the remaining reductions for Theorem 1.3, and the hardness results for counting Steiner trees and connected vertex covers. A schematic representation of our reductions can be found in Figure 1.

2 Preliminaries and Notation

In this paper, Δ denotes the symmetric difference and $\dot\cup$ denotes the disjoint union. For a set U and a positive integer $i \leq |U|$, we denote the family of all subsets of U of size i by $\binom{U}{i}$. In this paper, $\equiv$ will always denote congruence modulo 2, that is, $i \equiv j$ holds for integers i, j if and only if i and j have the same parity. Every assignment $\alpha\colon \{v_1, \ldots, v_n\} \to \{0, 1\}$ to n Boolean variables $v_1, \ldots, v_n$ is identified with the set $A := \{v_i \mid \alpha(v_i) = 1\} \subseteq \{v_1, \ldots, v_n\}$.

Since we consider a significant number of problems in the paper, each of which has a few variants, we use the following notation for clarity. We write k-Π for problems whose input consists of set systems of sets of size at most k, or CNF formulas with clauses of width at

most k. We write (k, c)-Sparse-Π if, in addition, the set systems or formulas are required to have density at most c. That is, the number of sets or clauses is at most cn, where n is the number of elements or variables.

For each problem Π that we consider, we fix the canonical NP-verifier that is implicit in the way we define the problem. Then every yes-instance of Π has associated with it a set of NP-witnesses or "solutions". We write $\oplus\Pi$ for the problem of deciding whether, for a given instance, the parity of the number of solutions odd. If solutions of Π are sets (e.g., of vertices), we write $\oplus_t\Pi$ for the problem of deciding whether, for a given instance, the parity of the number of solution sets that have size exactly t is odd.

Running times in this paper have the form $c^n \cdot \text{poly}(m)$, where c is a nonnegative constant, m is the total size of the input, and n is a somewhat smaller parameter of the input, typically the number of variables, vertices, or elements. The constant c is the *growth rate* of the running time, and it may be different for different choices for the parameter. To make this parameterization explicit, we use the notation Π/n. For every such parameterized problem, we now define the number $\sigma = \sigma(\Pi/n)$.

Definition 2.1. *For a parameterized problem Π/n, let $\sigma(\Pi/n)$ be the infimum over all $\epsilon > 0$ such that there exists a randomized $2^{\epsilon n}\text{poly}(m)$-time algorithm for Π whose error probability is at most $1/3$.*

The *optimal growth rate* of Π with respect to n is $C := 2^{\sigma(\Pi/n)}$. If the infimum in the definition above is a minimum, then Π has an algorithm that runs in time $C^n\text{poly}(m)$ and no algorithm for Π can have a running time $c^n\text{poly}(m)$ for any $c < C$. On the other hand, if the minimum does not exist, then no algorithm for Π can run in time $C^n\text{poly}(m)$, but Π has a $c^n\text{poly}(m)$-time algorithm for every $c > C$. We formally define SETH as the assertion that $\lim_{k\to\infty} \sigma(k\text{-CNF-Sat}/n) = 1$.

We remark that it is consistent with current knowledge that SETH fails and yet CNF-Sat does not have $2^{\epsilon n}\text{poly}(m)$-algorithms for any $\epsilon < 1$: If SETH fails, then k-CNF-Sat has, say, $k^k 1.99^n$-time algorithms for every k, which does not seem to translate to a $2^{\epsilon n}\text{poly}(m)$-time algorithm for CNF-Sat for any $\epsilon < 1$.

3 On Improving Branching Algorithms

In this section we show that significantly faster algorithms for search problems such as Hitting Set and Set Splitting imply significantly faster algorithms for CNF-Sat. More precisely, we prove that the growth rates of these problems are equal, or equivalently, $\sigma(\text{CNF-Sat}/n) = \sigma(\text{Hitting Set}/n) = \sigma(\text{Set Splitting}/n)$. We also give a reduction from $\oplus$CNF-Sat to $\oplus$Hitting Set, thus establishing a connection between the parity versions of these two problems.

Given an n-variable CNF formula with m clauses, the problems CNF-Sat and $\oplus$CNF-Sat problems are to determine whether there exists a satisfying assignment and whether the number of satisfying assignments is odd, respectively. With the same input, the NAE-Sat problem is to determine whether there exists an assignment such that every clause contains both a literal set to true and a literal set to false.

Given an integer t and a set system $\mathcal{F} \subseteq 2^U$ with $|\mathcal{F}| = m$ and $|U| = n$, the problems Hitting Set and $\oplus$Hitting Set are to determine whether there exists a hitting set of size at most t and whether the number of hitting sets is odd, respectively. A *hitting set* is a subset $H \subseteq U$ such that $H \cap S \neq \emptyset$ for every $S \in \mathcal{F}$. With the same input, the Set Splitting problem asks whether there is a subset $X \subseteq U$ such that, for every $S \in \mathcal{F}$, we have $S \not\subseteq X$ and $S \not\subseteq (U \setminus X)$.

3.1 Previous results for CNF-SAT

In the following few subsections, we show reductions from CNF-SAT/n to HITTING SET/n and SET SPLITTING/n. These reductions work even when the given instance of CNF-SAT/n is dense, that is, when there is no bound on the number of clauses that is linear in the number of variables. However, our starting point in Section 4 is the SPARSE-HITTING SET/n problem, where the number of sets in the set system is linear in n. For this reason we formulate our results for the sparse versions of HITTING SET/n and SET SPLITTING/n, and we develop a sparse version of SETH first.

The sparsification lemma by Impagliazzo et al. [IPZ01] is that every k-CNF formula φ can be written as the disjunction of $2^{\epsilon n}$ formulas in k-CNF, each of which has at most $c(k,\epsilon)\cdot n$ clauses. Moreover, this disjunction of sparse formulas can be computed from φ and ϵ in time $2^{\epsilon n}\cdot \mathrm{poly}(m)$. Hence, the growth rate of k-CNF-SAT for formulas of density at most $c(k,\epsilon)$ is ϵ-close to the growth rate of general k-CNF-SAT. More precisely, for every k and every $\epsilon > 0$, we have $\sigma\big((k,c)\text{-SPARSE-CNF-SAT}/n\big) \le \sigma\big(k\text{-CNF-SAT}/n\big) \le \sigma\big((k,c)\text{-SPARSE-CNF-SAT}/n\big) + \epsilon$, where the first inequality is trivial and the second inequality follows from the sparsification lemma. The density $c = c(k,\epsilon)$ is the *sparsification constant*, and the best known bound is $c(k,\epsilon) = (k/\epsilon)^{3k}$ [CIP06]. By setting $\epsilon = \epsilon(k) = \omega(1)$, this immediately yields the following theorem.

Theorem 3.1 ([IPZ01; CIP06])**.** *For every function $c = c(k) \ge (\omega(k))^{3k}$, we have*

$$\lim_{k\to\infty} \sigma\Big(k\text{-CNF-SAT}/n\Big) = \lim_{k\to\infty} \sigma\Big((k,c)\text{-SPARSE-CNF-SAT}/n\Big)\,.$$

Hence, SETH is equivalent to the right-hand side being equal to 1. In [DHM+12] it was observed that the sparsification lemma can be made parsimonious, which gives the following equality for the same functions $c = c(k)$:

$$\lim_{k\to\infty} \sigma\Big(k\text{-}\oplus\text{CNF-SAT}/n\Big) = \lim_{k\to\infty} \sigma\Big((k,c)\text{-}\oplus\text{SPARSE-CNF-SAT}/n\Big)\,.$$

We define $\oplus$-SETH as the assertion that these limits are equal to 1. The isolation lemmas for k-CNF formulas [CIK+03; Tra08] immediately yield that SETH implies $\oplus$-SETH. More precisely, we have the following theorem.

Theorem 3.2 ([CIK+03; Tra08])**.** $\lim_{k\to\infty} \sigma(k\text{-CNF-SAT}/n) \le \lim_{k\to\infty} \sigma(k\text{-}\oplus\text{CNF-SAT}/n)\,.$

3.2 From CNF-SAT to Hitting Set

The following construction will be useful in this subsection and in Subsection 3.4. Given a CNF formula $\varphi = C_1 \wedge \ldots \wedge C_m$ over n variables $v_1,\ldots,v_n$ and an odd integer $p \ge 3$ that divides n, we construct the set system $\mathcal{F}_{\varphi,p} \subseteq 2^U$ as follows.

1. Let p' be the odd integer $p' = p + 2\lceil \log_2 p\rceil$, and let $U = \{u_1,\ldots,u_{n'}\}$ with $n' = p'\cdot n/p$.
2. Partition the variables of φ into blocks V_i of size p, i.e., $V_i := \{v_{pi+1},\ldots,v_{p(i+1)}\}$.
3. Partition U into blocks U_i of size p', i.e., $U_i = \{u_{p'i+1},\ldots,u_{p'(i+1)}\}$.
4. Choose an arbitrary injective function $\psi_i\colon 2^{V_i} \to \binom{U_i}{\lceil p'/2\rceil}$. This exists since

$$\left|\binom{U_i}{\lceil p'/2\rceil}\right| = \binom{p'}{\lceil p'/2\rceil} \ge \frac{2^{p'}}{p'} \ge \frac{2^p p^2}{p + 2\lceil\log_2 p\rceil} \ge 2^p = \left|2^{V_i}\right|.$$

We think of ψ_i as a mapping that, given an assignment to the variables of V_i, associates with it a subset of U_i of size $\lceil p'/2 \rceil$.

5. If $X \in \binom{U_i}{\lceil p'/2 \rceil}$ for some i, then add the set X to $\mathcal{F}_{\varphi,p}$.
6. If $X \in \binom{U_i}{\lfloor p'/2 \rfloor}$ for some i such that $\psi_i^{-1}(\{U_i \setminus X\}) = \emptyset$, then add the set X to $\mathcal{F}_{\varphi,p}$.
7. For every clause C of φ, do the following:
   - Let $I = \{1 \leq j \leq \frac{n}{p} \mid C \text{ contains a variable of block } V_j\}$;
   - For every $i \in I$, we let $\mathcal{A}_i$ be the set

$$\left\{ A \in \binom{U_i}{\lfloor p'/2 \rfloor} \;\middle|\; \text{some assignment in } \psi_i^{-1}(\{U_i \setminus A\}) \text{ sets all variables in } C \cap V_i \text{ to false} \right\};$$

   - For every tuple $(A_i)_{i \in I}$ with $A_i \in \mathcal{A}_i$, add the set $\bigcup_{i \in I} A_i$ to $\mathcal{F}_{\varphi,p}$.

Lemma 3.3. *For every n-variable CNF formula φ and every odd integer $p \geq 3$ that divides n, the number of satisfying assignments of φ is equal to the number of hitting sets of size $\lceil \frac{p'}{2} \rceil \frac{n}{p}$ of the set system $\mathcal{F}_{\varphi,p}$, where $p' = p + 2\lceil \log_2 p \rceil$.*

Proof. For convenience denote $g = \frac{n}{p}$. Define $\psi \colon 2^V \to 2^U$ as $\psi(A) = \bigcup_{i=1}^{g} \psi_i(A \cap V_i)$. Note that ψ is injective, since for every i, ψ_i is injective. Hence to prove the lemma, it is sufficient to prove that (1) A is a satisfying assignment if and only if $\psi(A)$ is a hitting set of size $\lceil \frac{p'}{2} \rceil g$, and (2) if there is no assignment $A \subseteq V$ such that $\psi(A) = H$, than no set $H \subseteq U$ of size $\lceil \frac{p'}{2} \rceil g$ is a hitting set of $\mathcal{F}_{\varphi,p}$.

For the forward direction of (1), note that the sets added in Step 5 are hit by the pigeon-hole principle since $|\psi_i(A \cap V_i)| = \lceil \frac{p'}{2} \rceil$ and p' is odd. For the sets added in Step 6, consider the following. The set X of size $\lfloor p'/2 \rfloor$ is added because for some i, $\psi_i^{-1}(\{U_i \setminus X\}) = \emptyset$. Thus $\psi_i(A \cap V_i)$ automatically hits X. For the sets added in Step 7, consider a clause C of φ and the associated index set I as in Step 7. Since A is a satisfying assignment of φ, there exists $i \in I$ such that A sets at least one variable in $C \cap V_i$ to true. Hence, $U_i \setminus \psi_i(A \cap V_i) \notin \mathcal{A}_i$. On the other hand, $U_i \setminus \psi_i(A \cap V_i)$ is the only member of $\mathcal{F}_{\varphi,p}$ that cannot be hit by $\psi(A \cap V_i)$. Therefore, all sets added in Step 7 are hit by $\psi(A)$. It is easy to check that $\psi(A)$ has size $\lceil \frac{p'}{2} \rceil g$ since there are g blocks.

For the reverse direction of (1), let A be an assignment such that $\psi(A)$ is a hitting set of size $\lceil \frac{p'}{2} \rceil g$. We show that A is a satisfying assignment of φ. Suppose for the sake of contradiction that a clause C is not satisfied by A, and let I be as defined in Step 7 for this C. Since $\psi(A)$ is a hitting set, $|\psi(A) \cap U_i| \geq \frac{p'}{2}$ for every i because it hits all sets added in Step 5. More precisely, $|\psi(A) \cap U_i| = \lceil \frac{p'}{2} \rceil$ because $|\psi(A)| = \lceil \frac{p'}{2} \rceil g$ and there are g disjoint blocks $U_1, \ldots, U_g$. Therefore, $|U_i \setminus \psi(A)| = \lfloor \frac{p'}{2} \rfloor$, and so $U_i \cap \psi(A) = U_i \setminus (U_i \setminus \psi(A))$ is a member of $\mathcal{A}_i$ for every i. This means that in Step 7 the set $\bigcup_{i \in I} A_i$ with $A_i = U_i \setminus \psi(A)$ was added, but this set is not hit by $\psi(A)$. So it contradicts that $\psi(A)$ is a hitting set.

For (2), let $H \subseteq U$ be a set of size $\lceil \frac{p'}{2} \rceil g$ and assume that there is no assignment $A \subseteq V$ such that $\psi(A) = H$. We show that H is not a hitting set of $\mathcal{F}_{\varphi,p}$. For the sake of contradiction, suppose that H is a hitting set. Then, as in the proof of the reverse direction of (1), we obtain $|H \cap U_i| = \lceil \frac{p'}{2} \rceil$ for every i. Since it hits all sets added in Step 6, we also know that $\psi_i^{-1}(\{H \cap U_i\}) \neq \emptyset$ for every i. However, this contradicts the non-existence of $A \subseteq V$ such that $\psi(A) = H$. □

Theorem 3.4. *For every function $c = c(k)$, there exists a function $c' = c'(k')$ such that*

$$\lim_{k\to\infty} \sigma((k,c)\text{-Sparse-CNF-Sat}/n) \leq \lim_{k'\to\infty} \sigma((k',c')\text{-Sparse-Hitting Set}/n)\,.$$

Proof. To prove the theorem we show that for any positive integers k, c and for any positive odd integer $p \geq 3$, there exist positive integers k' and c' such that

$$\sigma((k,c)\text{-Sparse-CNF-Sat}/n) \leq \sigma((k',c')\text{-Sparse-Hitting Set}/n) + O\left(\frac{\log p}{p}\right)\,.$$

Create the set system $\mathcal{F}_{\varphi,p}$ as described above. For a constant p, this can clearly be done in polynomial time. We set $k' = p'k$ and $c' = 2p' + 2^{kp'}c$ (remind that $p' = p + 2\lceil \log_2 p \rceil$). It is easy to see that the maximum size of a set of $\mathcal{F}_{\varphi,p}$ is at most k'. Let m' be the number of sets in $\mathcal{F}_{\varphi,p}$. Observe that there are at most $2^{p'}n/p$ sets added in Step 5 and Step 6. Moreover, since each clause contains variables from at most k blocks, there are at most $2^{kp'}m$ sets added in Step 7. Therefore $m'/n' \leq m'/n \leq 2^{p'} + 2^{kp'}c = c'$ and we can determine the minimum hitting set of $\mathcal{F}_{\varphi,p}$ in $O(2^{\sigma((k',c')\text{-Sparse-Hitting Set}/n)n'}n^{O(1)})$ time, where n' is the size of the universe of $\mathcal{F}_{\varphi,p}$. By Lemma 3.3, φ is satisfiable if and only if the size of a minimum hitting set is $\lceil \frac{p'}{2} \rceil \frac{n}{p}$. Since $n' = \frac{n}{p}(p + 2\lceil \log p \rceil) = n(1 + O(\frac{\log p}{p}))$, the theorem follows. □

3.3 From Hitting Set via Set Splitting to CNF-SAT

Theorem 3.5.

$$\lim_{k\to\infty} \sigma(k\text{-Hitting Set}/n) \leq \lim_{k\to\infty} \sigma(k\text{-Set Splitting}/n)\,.$$

Proof. Observe that to prove the theorem it is enough to show that for every positive integers k, p we have

$$\sigma(k\text{-Hitting Set}/n) \leq \sigma(k'\text{-Set Splitting}/n) + \frac{\log_2(p+1)}{p}\,,$$

where $k' = \max(k+1, p+1)$. Let $(\mathcal{F}, t)$ be an instance of k-Hitting Set. We can assume that the universe U of $\mathcal{F}$ has n elements and that p divides n. Let $U = U_1 \dot\cup \ldots \dot\cup U_{n/p}$ be a partition in which each part has exactly $|U_i| = p$ elements of the universe U. Let $t_1, \ldots, t_{n/p}$ be nonnegative integers such that $\sum_{i=1}^{n/p} t_i = t$. The t_i's are our current guess for how many elements of a t-element hitting set will intersect with the U_i's. The number of ways to write t as the ordered sum of n/p nonnegative integers $t_1, \ldots, t_{n/p}$ with $0 \leq t_i \leq p$ can be bounded by $(p+1)^{n/p} = 2^{n/p \cdot \log_2(p+1)}$. For each choice of the t_i's, we construct an instance $\mathcal{F}'$ of k'-Set Splitting as follows.

1. Let R (red) and B (blue) be two special elements and add the set $\{R, B\}$ to $\mathcal{F}'$.
2. For all i with $t_i < p$ and for all $X \in \binom{U_i}{t_i+1}$, add $X \cup \{R\}$ to $\mathcal{F}'$.
3. For every $Y \in \mathcal{F}$, add $Y \cup \{B\}$ to $\mathcal{F}'$.

Clearly $\mathcal{F}'$ can be computed in polynomial time and its universe has $n+2$ elements. The sets added in step 2 have size at most $p+1$ and the sets added in step 3 have size at most $k+1$. Given an algorithm for Set Splitting, we compute $\mathcal{F}'$ for every choice of the t_i's and we decide Hitting Set in time $O(2^{(\epsilon+\sigma(k'\text{-Set Splitting}))\cdot n}m^{O(1)})$. It remains to show that $\mathcal{F}$ has a hitting set of size at most t if and only if $\mathcal{F}'$ has a set splitting for some choice of $t_1, \ldots, t_{n/p}$.

For the completeness of the reduction, let H be a hitting set of size t and set $t_i = |U_i \cap H|$ for all i. We now observe that $H \cup \{R\}$ and its complement $(U - H) \cup \{B\}$ form a set splitting of $\mathcal{F}'$. The set $\{R, B\}$ added in step 1 is split. The sets $X \cup \{R\}$ added in step 2 are split since at least one of the $t_i + 1$ elements of $X \subseteq U_i$ is not contained in H. Finally, the sets $Y \cup \{B\}$ added in step 3 are split since each $Y \in \mathcal{F}$ has a non-empty intersection with H.

For the soundness of the reduction, let $(S, \overline{S})$ be a set splitting of $\mathcal{F}'$ for some choice of $t_1, \ldots, t_{n/p}$. Without loss of generality, assume that $R \in S$. By the set added in step 1, this means that $B \in \overline{S}$. The sets added in step 2 guarantee that $U_i \cap S$ contains at most t_i elements for all i. Finally, the sets added in step 3 make sure that each set $Y \in \mathcal{F}$ has a non-empty intersection with S. Thus, $S \setminus \{R\}$ is a hitting set of $\mathcal{F}$ and has size at most $\sum_i t_i = t$. □

Observation 3.6. *For any positive integer k we have*

$$\sigma(k\text{-Set Splitting}/n) \leq \sigma(k\text{-NAE-Sat}/n) \leq \sigma(k\text{-CNF-Sat}/n)\,.$$

Proof. For the first reduction, let $\mathcal{F}$ be an instance of k-Set Splitting. We construct an equivalent k-CNF formula φ as follows. For each element in the universe of $\mathcal{F}$, we add a variable, and for each set $X \in \mathcal{F}$ we add a clause in which each variable occurs positively. A characteristic function of a set splitting $U = U_1 \dot\cup U_2$ is one that assigns 1 to the elements in U_1 and 0 to the elements of U_2. Observe that the characteristic functions of set splittings of $\mathcal{F}$ stand in one-to-one correspondence to variable assignments that satisfy the NAE-Sat constraints of φ. Thus, any algorithm for k-NAE-Sat works for k-Set Splitting, too.

For the second reduction, let φ be a k-NAE-Sat-formula. The standard reduction to k-CNF-Sat creates two copies of every clause of φ and flips the sign of all literals in the second copies. Then any NAE-Sat-assignment of φ satisfies both copies of the clauses of φ'. On the other hand, any satisfying assignment of φ' sets a literal to true and a literal to false in each clause of φ. Thus any algorithm for k-CNF-Sat works for k-NAE-Sat, too. □

3.4 From Parity CNF-SAT to Parity Hitting Set

Given a CNF formula φ over n variables and clauses of size at most k and an odd integer $p > 2$ that divides n, we first create the set system $\mathcal{F}_{\varphi,p} \subseteq 2^U$ as described in Section 3.2. Given the set system $\mathcal{F}_{\varphi,p} \subseteq 2^U$, create the set system $\mathcal{F}'_{\varphi,p}$ as follows:

8. For every block U_i:
   - add a special element e_i to the universe,
   - for every $X \in \binom{U_i}{\lfloor p'/2 \rfloor}$, add the set $X \cup \{e_i\}$ to the set family.

Lemma 3.7. *The number of hitting sets of the instance $\mathcal{F}_{\varphi,p}$ of size $\lceil p'/2 \rceil \frac{n}{p}$ is odd if and only if the number of hitting sets of the instance $\mathcal{F}'_{\varphi,p}$ is odd.*

Proof. Let $g = \frac{n}{p}$. We first prove that the number of hitting sets of $\mathcal{F}_{\varphi,p}$ of size $\lceil p'/2 \rceil g$ is equal to the number of hitting sets H' of $\mathcal{F}'_{\varphi,p}$ such that $|H' \cap U_i| = \lceil \frac{p'}{2} \rceil$ for every $1 \leq i \leq g$. Suppose that H is a hitting set of $\mathcal{F}_{\varphi,p}$ of size $\lceil p'/2 \rceil g$, then it is easy to see that $H \cup \{e_1, \ldots, e_g\}$ is a hitting set of $\mathcal{F}'_{\varphi,p}$ since all the sets added in Step 8 are hit by some e_i, and indeed $|H' \cap U_i| = \lceil \frac{p'}{2} \rceil$ for every $1 \leq i \leq g$ since otherwise the set $U_i \setminus H'$ added in Step 5 is not hit by H'. For the reverse direction, suppose H' is a hitting set of $\mathcal{F}'_{\varphi,p}$ such that $|H' \cap U_i| = \lceil \frac{p'}{2} \rceil$ for every $1 \leq i \leq g$. Then $\{e_1, \ldots, e_g\} \subseteq H'$ since all the sets added in Step 8 are hit by H'. And hence we have a bijection between the two families of hitting sets.

For every hitting set H' of $\mathcal{F}'_{\varphi,p}$ and block U_i, we know that $|H' \cap U_i| \geq \lceil p'/2 \rceil$. So it remains to show that the number of hitting sets H' of $\mathcal{F}'_{\varphi,p}$ such that there is an $1 \leq i \leq g$ with $|H' \cap U_i| > \lceil \frac{p'}{2} \rceil$ is even. Given such a hitting set H', let $\gamma(H') = H' \Delta \{e_i\}$ where i is the smallest integer such that $|H' \cap U_i| > \lceil \frac{p'}{2} \rceil$. Obviously γ is its own inverse and $|\gamma(H') \cap U_i| > \lceil \frac{p'}{2} \rceil$ so now it remains to show that $\gamma(H')$ is also a hitting set of $\mathcal{F}'_{\varphi,p}$. To see this, notice that all sets $X \cup \{e_i\}$ added in Step 8 where $X \in \binom{U_i}{\lfloor p'/2 \rfloor}$ are hit since $|\gamma(H') \cap U_i| > \lceil \frac{p'}{2} \rceil$ and that those are the only sets containing e_i. □

Theorem 3.8. *For every function $c = c(k)$, there exists a function $c' = c'(k')$ such that*

$$\lim_{k\to\infty} \sigma((k,c)\text{-}\oplus\textsc{Sparse-CNF-Sat}/n) \leq \lim_{k'\to\infty} \sigma((k',c')\text{-}\oplus\textsc{Sparse-Hitting Set}/n)\,.$$

Proof. To prove the theorem we show that for any positive integers k, c, p, there exist positive integers k', c', such that we have

$$\sigma((k,c)\text{-}\oplus\textsc{Sparse-CNF-Sat}/n) \leq \sigma((k',c')\text{-}\oplus\textsc{Sparse-Hitting Set}/n) + O\left(\frac{\log p}{p}\right).$$

Create the set system $\mathcal{F}'_{\varphi,p}$ as described above. For a constant p, this can clearly be done in polynomial time. Recall that there are at most $(2^{p'} + 2kp'c)n$ sets in $\mathcal{F}_{\varphi,p}$, each of size at most $p'k$. Since in Step 8 we add at most $2^{p'}n/p$ sets, each of size at most p', we infer that $\mathcal{F}'_{\varphi,p}$ is an instance of (k',c')-$\oplus$Sparse-Hitting Set$/n$, where $k' = p'k$ and $c' = 2^{p'+1} + 2kp'c$. Therefore we can determine the number of hitting sets modulo 2 of $\mathcal{F}'_{\varphi,p}$ in $O(2^{\sigma((k',c')\text{-}\oplus\textsc{Sparse-Hitting Set}/n)n'} m^{O(1)})$ time, where n' is the size of the universe of $\mathcal{F}'_{\varphi,p}$. Since $n' = \lceil \frac{n}{p} \rceil (p + 2\lceil \log p \rceil) = n(1 + O(\frac{\log p}{p}))$, the theorem follows. □

Note that conversely, an improved algorithm for $\oplus$CNF-Sat gives an improved algorithm for $\oplus$Hitting Set: given a set family $\mathcal{F} \subseteq U$ the required reduction simply associates a variable with the elements of U and creates a CNF-formula with for every $S \in \mathcal{F}$ a clause which is a disjunction of the variables associated with the elements of S. The correspondence between hitting sets and satisfying assignments is then immediate. Also, using a construction dual to this, a similar relation between $\oplus$CNF-Sat$/m$ and Set Cover can be shown.

3.5 Satisfiability for Series-Parallel Circuits

In this subsection, we show that the satisfiability of cn-size *series-parallel* circuits can be decided in time time $2^{\delta n}$ for $\delta < 1$ independent of c if and only if SETH is not true. Here the size of a circuit is the number of wires. Our proof is based on a result of Valiant regarding paths in sparse graphs [Val77]. Calabro [Cal08] discusses various notions of series-parallel graphs and provides a more complete proof of Valiant's lower bound on the size of series-parallel graphs (which he calls Valiant series-parallel graphs) that have "many" long paths. We remark that the class of Valiant series-parallel graphs is not the same as the notion of series-parallel graphs used most commonly in graph theory (see [Cal08]).

In this section a *multidag* $G = (V, E)$ is a directed acyclic multigraph. Let input(G) denote the set of vertices $v \in V$ such that the indegree of v in G is zero. Similarly, let output(G) denote the set of vertices $v \in V$ such that the outdegree of v in G is zero. A *labeling* of G is a function $l\colon V \to \mathbb{N}$ such that $\forall (u,v) \in E$, $l(u) < l(v)$. A labeling l is *normal* if for all $v \in$ input(G), $l(v) = 0$ and there exists an integer $d \in \mathbb{N}$ such that for all $v \in$ output$(G) \setminus$ input(G), $l(v) = d$. A multidag G is *Valiant series-parallel* (VSP) if it has a normal labeling l such that there exist no $(u,v), (u',v') \in E$ such that $l(u) < l(u') < l(v) < l(v')$.

We say that a boolean circuit C is a VSP *circuit* if the underlying multidag of C is a VSP graph and the indegree of every node is at most two (namely, the fan-in of each gate is at most two). Using the depth-reduction result by Valiant [Val77] and following the arguments by Calabro [Cal08] and Viola [Vio09], we may show the following.

Theorem 3.9. *Let C be a* VSP *circuit of size cn with n input variables. There is an algorithm A which on input C and a parameter $d \geq 1$ outputs an equivalent depth-3 unbounded fan-in OR-AND-OR circuit C' with the following properties.*

1. *Fan-in of the top OR gate in C' is bounded by $2^{n/d}$.*
2. *Fan-in of the bottom OR gates is bounded by $2^{2^{\mu cd}}$ where μ is an absolute constant.*
3. *A runs in time $O(2^{n/d}n^{O(1)})$ if c and d are constant.*

In other words, for all $d \geq 1$, Theorem 3.9 reduces the satisfiability of a cn-size VSP circuit to that of the satisfiability of a disjunction of $2^{n/d}$ k-CNFs where $k \leq 2^{2^{\mu cd}}$ in time $O(2^{n/d}n^{O(1)})$. This implies that

$$\sigma(c\text{-VSP-Circuit-SAT}/n) \leq \sigma(2^{2^{\mu cd}}\text{-CNF-Sat}/n) + \frac{1}{d}.$$

Hence, we obtain the following theorem.

Theorem 3.10.

$$\lim_{c\to\infty} \sigma(c\text{-VSP-Circuit-SAT}/n) \leq \lim_{k\to\infty} \sigma(k\text{-CNF-Sat}/n).$$

For the reverse direction, observe that a CNF formula with cn clauses, all of size at most k, can be written as a $4ck$-size VSP circuit. This observation implies that

$$\sigma((k,c)\text{-Sparse-CNF-Sat}/n) \leq \sigma(4ck\text{-VSP-Circuit-SAT}/n).$$

Together with the sparsification lemma, Theorem 3.1, we obtain the following theorem.

Theorem 3.11. $\lim_{k\to\infty} \sigma(k\text{-CNF-Sat}/n) \leq \lim_{c\to\infty} \sigma(c\text{-VSP-Circuit-SAT}/n)$.

4 On Improving Dynamic Programming Based Algorithms

In this section we give some reductions that show that several dynamic programming based algorithms cannot be improved unless (the parity version of) CNF-Sat can be, using the hardness of $\oplus$Hitting Set$/n$ showed in the previous section. More specifically, we show that $\oplus$Hitting Set$/n$ and $\oplus$Set Cover$/n$ are equivalent using a simple but novel property of bipartite graphs in Subsection 4.1, and in Subsection 4.2 we show that the current algorithms for $\oplus_t$Steiner Tree$/t$ and $\oplus_t$Connected Vertex Cover$/k$ are at least as hard to improve as the algorithm for $\oplus$Set Cover$/n$. Motivated we make the hypothesis that the current algorithm for Set Cover can not be improved and show similar implications to the Steiner Tree$/t$ and Connected Vertex Cover$/k$, Set Partitioning and Subset Sum problems.

Given an integer t and a set system $\mathcal{F} \subseteq 2^U$ where $|\mathcal{F}| = m$ and $|U| = n$, the Set Cover and $\oplus$Set Cover problems ask to determine whether there is a hitting set of size at most t and whether the number of hitting sets if odd, respectively. Here a set cover refer to a subset $\mathcal{CF}$ such that $\cup_{S\in C}\mathcal{C} = U$. Given a graph $G = (V, E)$, with $|V| = n$ a subset $T \subseteq V$, and an integer t the Steiner Tree and $\oplus_t$Steiner Tree problems ask to determine whether is a hitting set

of size at most t and whether the number of hitting sets is odd, respectively. Here, a Steiner tree is a subset $T \subseteq X \subseteq V$ such that X induces a connected graph in G. Given a graph $G = (V, E)$ with $|V| = n$ and an integer t, the Connected Vertex Cover and $\oplus_t$Connected Vertex Cover problems ask to determine whether there is a connected vertex cover of size at most t and whether the number of connected vertex covers is odd, respectively. Here, a *connected vertex cover* is a subset $X \subseteq V$ such that $X \cap e \neq \emptyset$ for every $e \in E$ and X induces a connected graph. We will also use the extended notation as explained in Section 2 denoting several variants of these problems (see also the appendix).

4.1 The flip: Parity Hitting Set equals Parity Set Cover

Lemma 4.1. *Let $G = (A \cup B, E)$ be a bipartite graph, then the number of independent sets of G modulo 2 is equal to*

$$|\{X \subseteq A : N(X) = B\}|.$$

Proof. Grouping on their intersection with A, the number of independent sets of G is equal to

$$\sum_{X \subseteq A} 2^{|B \setminus N(X)|} \equiv \sum_{\substack{X \subseteq A \\ |B \setminus N(X)|=0}} 2^0 = |\{X \subseteq A : N(X) = B\}|$$

and the lemma follows. □

It is worth mentioning that this lemma was inspired by a non-modular variant from [NR10, Lemma 2] (see also [vRo11, Proposition 9.1]).

Theorem 4.2. $\sigma(\oplus\text{Hitting Set}/n) = \sigma(\oplus\text{Set Cover}/n)$.

Proof. Given a set system $\mathcal{F} \subseteq 2^U$, let $G = (\mathcal{F}, U, E)$ be the bipartite graph where $(S, e) \in E$ if and only if $e \in S$. Note that the number of hitting sets of $\mathcal{F}$ is equal to $|\{X \subseteq U : N(X) = \mathcal{F}\}|$. Then by Lemma 4.1, the number of hitting sets is equal to the number of independent sets of G modulo 2. And similarly, since the lemma is symmetric with respect to the two color classes of the bipartite graph, the number of set covers of $\mathcal{F}$ is also equal to the number of independent sets of G modulo 2. Thus the problems are equivalent. □

Observe that in the proof of Theorem 4.2 the same set system is used as an instance of $\oplus$Hitting Set$/n$ and $\oplus$Set Cover$/n$. Hence the above directly gives the following corollary, which we will need in the next subsection.

Corollary 4.3. *For every function $c = c(k)$, there exists a function $c' = c'(k')$ such that*

$$\lim_{k\to\infty} \sigma((k,c)\text{-}\oplus\text{Sparse-Hitting Set}/n) \leq \lim_{k\to\infty} \sigma((k,c)\text{-}\oplus\text{Sparse-Set Cover}/n).$$

4.2 From Set Cover to Steiner Tree and Connected Vertex Cover

In this subsection we will give reductions from Set Cover$/n$ to Steiner Tree$/t$ and Connected Vertex Cover$/k$. We transfer the reductions to the parity versions $\oplus$Set Cover$/n$, $\oplus_t$Steiner Tree$/t$, and $\oplus_t$Connected Vertex Cover$/k$. For the reduction, we first need an intermediate result, showing that Set Cover$/(n+t)$, that is, Set Cover parameterized by the sum of the size of the universe and solution size, is as hard as Set Cover$/n$ (and similarly for $\oplus$Set Cover$/n$ and $\oplus$Set Cover$/(n+t)$). Once we have this intermediate result, the reductions to the $\oplus_t$Steiner Tree$/t$ and $\oplus_t$Connected Vertex Cover$/k$ problems follow more easily.

Theorem 4.4. $\lim_{k\to\infty} \sigma(k\text{-Set Cover}/n) \leq \lim_{k\to\infty} \sigma(k\text{-Set Cover}/(n+t))$.

Proof. As a proof we present a reduction which for fixed $\alpha > 0$ transforms an instance $(\mathcal{F}, U, t)$ of k-Set Cover into an instance of k'-Set Cover, for some positive integer k', where the size t' of the solution in the resulting p'-Set Cover instances is at most $\alpha|U|$, without changing the universe size.

Without loss of generality, we assume that $t \leq |U|$. Consider any $\alpha > 0$. Let q be the smallest positive integer such that $\frac{1}{q} \leq \alpha$. We may assume that t is divisible by q, since otherwise we may add at most q additional elements to the universe U and singleton sets to the family $\mathcal{F}$. We form a family $\mathcal{F}'$ of all unions of exactly q sets from $\mathcal{F}$, that is for each of $\binom{|\mathcal{F}|}{q}$ choices of q sets $S_1, \ldots, S_q \in \mathcal{F}$ we add to $\mathcal{F}'$ the set $\bigcup_{i=1}^{q} S_i$. Note that since q is a constant we can create $\mathcal{F}'$ in polynomial time. We set $t' = t/q \leq |U|/q \leq \alpha|U|$. It is easy to see that $(\mathcal{F}, U, t)$ is a YES-instance of k-Set Cover if and only if $(\mathcal{F}', U, t')$ is a YES-instance of qk-Set Cover. □

Observe that in the proof above, because of the grouping of q sets, one solution for the initial instance may correspond to several solutions in the resulting instance. For this reason the counting variant of the above reduction is much more technically involved.

Theorem 4.5. *For every function $c = c(k)$, we have*

$$\lim_{k\to\infty} \sigma((k,c)\text{-}\oplus\text{Sparse-Set Cover}/n) \leq \lim_{k'\to\infty} \sigma(k'\text{-}\oplus_t\text{Set Cover}/(n+t))$$

The involved proof is postponed to the end of this section, but first let us look at its consequences.

Theorem 4.6.

$$\lim_{k\to\infty} \sigma(k\text{-Set Cover}/(n+t)) \leq \sigma(\text{Steiner Tree}/t)\,, \textit{ and}$$
$$\lim_{k\to\infty} \sigma(k\text{-}\oplus_t\text{Set Cover}/(n+t)) \leq \sigma(\oplus_t\text{Steiner Tree}/t)\,.$$

Proof. Given an instance of Set Cover consisting of a set system $(\mathcal{F}, U)$ and integer i, let G' be the graph obtained from the incidence graph of $(\mathcal{F}, U)$ by adding a vertex s universal to $\mathcal{F}$ with a pendant vertex u, and define the terminal set to be $U \cup \{u\}$. It is easy to see that the number of Steiner trees with $|U| + i + 1$ edges is equal to the number of set covers of $(\mathcal{F}, U)$ of size i. Hence the theorem follows. □

Theorem 4.7.

$$\lim_{k\to\infty} \sigma(k\text{-Set Cover}/(n+t)) \leq \sigma(\text{Connected Vertex Cover}/t),\ \textit{and}$$
$$\lim_{k\to\infty} \sigma(k\text{-}\oplus_t\text{Set Cover}/(n+t)) \leq \sigma(\oplus_t\text{Connected Vertex Cover}/t).$$

Proof. Given an instance $(\mathcal{F}, U, t)$ of Set Cover, we create an instance of Connected Vertex Cover with G being obtained from the incidence graph of $(\mathcal{F}, U)$ by adding a vertex s adjacent to all vertices corresponding to sets and adding pendant vertices for every element of $U \cup \{s\}$. Moreover let $t' = t + |U| + 1$ in the Connected Vertex Cover instance.

It is easy to see that for every i, there exists a set cover of $(\mathcal{F}, U)$ of size $i \leq t$ if and only if there exists a connected vertex cover of G of size at most $i + |U| + 1 \leq t'$ since we can take without loss of optimality all vertices having a pendant vertex, and then connecting these

vertices is equivalent to covering all elements of U with sets in $\mathcal{F}$. Hence, by using an algorithm for Connected Vertex Cover, we obtain an $O(2^{\sigma(\text{Connected Vertex Cover}/t)t'}n^{O(1)}) = O(2^{\sigma(\text{Connected Vertex Cover}/t)(|U|+t)}n^{O(1)})$ time algorithm for p-$\oplus_t$Set Cover.

For the parity case, let us study the number of connected vertex covers of size j of G for every j. Similarly to the previous case, note that for any connected vertex cover C, $C \cap \mathcal{F}$ must be a set cover of $(\mathcal{F}, U)$ by the connectivity requirement. Hence we group all connected vertex covers in G depending on which set cover in $(\mathcal{F}, U)$ their intersection with $\mathcal{F}$ is. Let c_j be the number of connected vertex covers of G of size j and s_i be the number of set covers of size i in $(\mathcal{F}, U)$, then:

$$c_j = \sum_{i=1}^{j-|U|-1} s_i \binom{|U|+1}{j-i-|U|-1}.$$

Now the number s_i modulo 2 can be determined in polynomial time once $(c_1, \ldots, c_{i+|U|+1})$ modulo 2 are computed by recovering s_1 up to s_i in increasing order, since for $i = j - |U| - 1$ we have $\binom{|U|+1}{j-i-|U|-1} = 1$.

Thus, if in time $O(2^{\sigma(\text{Connected Vertex Cover}/t)t'}n^{O(1)})$ we can compute the number of connected vertex covers of size n modulo 2, we can compute the parity of all $(c_1, \ldots, c_{i+|U|+1})$ and hence the parity of s_i in $O(2^{\sigma(\text{Connected Vertex Cover}/t)(|U|+t)}n^{O(1)})$. □

4.3 From Set Cover to Set Partitioning and Subset Sum

Theorem 4.8.

$$\lim_{p\to\infty} \sigma(p\text{-Set Cover}/n) \leq \lim_{p\to\infty} \sigma(p\text{-Set Partitioning}/n).$$

Proof. Let $(\mathcal{F}, t)$ be an instance of p-Set Cover. Create an instance $(\mathcal{F}', t)$ of p-Set Partitioning by for every $S \in \mathcal{F}$ adding all subsets of S to $\mathcal{F}'$. Clearly $(\mathcal{F}', t)$ has a set partitioning of size at most t if and only if $(\mathcal{F}, t)$ has a set cover of size at most t. Since the size of the sets in $\mathcal{F}$ is bounded by p, the reduction runs in polynomial time. □

Theorem 4.9.

$$\lim_{k\to\infty} \sigma(k\text{-Set Partitioning}/n) \leq \sigma(\text{Subset Sum}/m).$$

Proof. Let $\mathcal{F} \subseteq 2^U$ be an instance of k-Set Partitioning. We iterate over all potential sizes $1 \leq t_0 \leq n$ of the solution for the Set Partitioning problem.

We create an instance of Subset Sum as follows. Let the target integer t for Subset Sum have a bit expansion consisting of three fields. First, as the most significant bits, a field coding the value of t_0, to check the cardinality of the solution $\mathcal{C} \subseteq \mathcal{F}$; second, a field of length $\log_2 t_0 + \log_2 n$ containing the value n, to check the total size of all sets in $\mathcal{C}$; finally, a field of length $\log_2 t_0 + n$ containing n ones. The paddings of length $\log_2 t_0$ serve to isolate the fields from each other. For every $S_i \in \mathcal{F}$, we create an integer a_i with the same field division as t, where the first field encodes 1, the second field encodes $|S_i|$, and the third field contains a one in position j if and only if $u_j \in S_i$. We argue that the resulting Subset Sum instance is a YES-instance if and only if $\mathcal{F}$ contains a partitioning of U using exactly t_0 sets.

Clearly, if $\mathcal{C} \subseteq \mathcal{F}$ partitions U and $|\mathcal{C}| = t_0$, then the integers a_i corresponding to $S_i \in \mathcal{C}$ sum to t. The first field sums to t_0 by cardinality of $\mathcal{C}$, the second sums to n, and in the third field the non-zero digits are simply partitioned between the a_i.

So let A be a collection of integers a_i that sum to t. By the first field, we have $|A| \leq t_0$; thus the padding of length $\log t_0$ is enough to isolate the fields, and we have $|A| = t_0$. By the same argument on the second field, the sum over all $a_i \in A$ of the number of non-zero bits in the third field is exactly n. Now, the only way that the third field can actually contain n true bits is if the true bits in the third field are partitioned among the a_i. Thus, $\mathcal{C} = \{S_i \mid a_i \in A\}$ is a set partitioning of U of cardinality exactly t_0.

By looping over all $1 \leq t_0 \leq t$ for the Set Partitioning instance, this solves the problem. Note that the length of the bit string t is $n+O(\log n)$, which disappears into the asymptotics. □

4.4 Proof of Theorem 4.5

As a proof we present a reduction which for fixed $\alpha > 0$ transforms an instance $(\mathcal{F}', U')$ of (k,c)-$\oplus$Sparse-Set Cover into polynomially many instances of the k'-$\oplus_t$Set Cover problem, for some positive integer k', where the size t of the solution in the resulting k'-$\oplus_t$Set Cover instances is at most $\alpha|U'|$.

In order to find the parity of the number of all set covers of the instance $(\mathcal{F}', U')$ we find the parity of the number of set covers of a particular size. That is we iterate over all possible sizes $j = 1, \ldots, |\mathcal{F}'|$ of a set cover. Let us assume that we want to find the parity of the number of set covers of size j and for each positive integer $j' < j$ we know the parity of the number of set covers of $(\mathcal{F}', U')$ of size j'. Let q be the smallest power of two satisfying $\frac{|\mathcal{F}'|}{q} + 2 \leq \alpha|U'|$. We assume that $\alpha|U'| \geq 3$ since otherwise the instance is small and we can solve it by brute force (recall that α is a given constant). Observe that q is upper bounded by a constant independent of $|U'|$ since $|\mathcal{F}'| \leq c|U'|$.

We create a temporary set system $(\mathcal{F}_0, U_0)$ to ensure that the size of the set covers we are looking for is divisible by q. Let $r = j \bmod q$. We make $(\mathcal{F}_0, U_0)$ by taking the set system $(\mathcal{F}', U')$ and adding $q - r$ new elements to the universe U_0 and also $q - r$ singleton sets of the new elements to the family $\mathcal{F}_0$. Now we are looking for the parity of the number of set covers of size $j_0 = j + (q - r)$ in $(\mathcal{F}_0, U_0)$. Observe that for each $j' < j_0$ we know the parity of the number of set covers of size j' in $(\mathcal{F}_0, U_0)$ since it is equal to the parity of set covers of $(\mathcal{F}', U')$ of size $j' - (q - r) < j$ which we already know.

To obtain a k'-$\oplus_t$Set Cover instance we set $U^* = U_0$ and we form a family $\mathcal{F}^*$ of all unions of exactly q sets from $\mathcal{F}_0$, that is for each of $\binom{|\mathcal{F}_0|}{q}$ choices of q sets $S_1, \ldots, S_q \in \mathcal{F}_0$ we add to $\mathcal{F}^*$ the set $\bigcup_{i=1}^{q} S_i$ (note that $\mathcal{F}^*$ might be a multiset). Finally we set $t^* = j_0/q$ which is an integer since $j + (q - r)$ is divisible by q. Observe that $t^* \leq \frac{j}{q} + 1 \leq \alpha|U'| - 1$, by the definition of q, but $(\mathcal{F}^*, U^*, t^*)$ might not be a proper instance of kq-$\oplus_t$Set Cover, since $\mathcal{F}^*$ could be a multiset. Note that each subset of U^* appears in $\mathcal{F}^*$ at most $(2^{kq})^q = 2^{kq^2}$ times, since $\mathcal{F}_0$ has no duplicates and each set in $\mathcal{F}^*$ is a union of exactly q sets from $\mathcal{F}_0$. To overcome this technical obstacle we make a new instance $(\mathcal{F}, U, t)$, where as U we take U^* with $z = 1 + kq^2$ elements added, $U = U^* \cup \{e_1, \ldots, e_z\}$. We use elements $\{e_1, \ldots, e_{z-1}\}$ to make sets from $\mathcal{F}^*$ different in $\mathcal{F}$ by taking a different subset of $\{e_1, \ldots, e_{z-1}\}$ for duplicates. Additionally we add one set $\{e_1, \ldots, e_z\}$ to the family $\mathcal{F}$ and set $t = t^* + 1$. In this way we obtain $(\mathcal{F}, U, t)$, that is a proper $(kq + z)$-$\oplus_t$Set Cover instance and $t = t^* + 1 \leq \alpha|U'|$. Observe that in the final instance we have $|U| \leq n + q + z$ and $|\mathcal{F}| \leq (cn + q)^q + 1$, which is a polynomial since k, c, q and z are constants.

To summarize the reduction, we have taken an instance of (k,c)-$\oplus$Sparse-Set Cover and iterated over the size of solution. Next we made the size divisible by q by adding additional elements to the universe and created a multiset family $\mathcal{F}^*$ from which we made a set family by differentiating identical sets with additional elements of the universe. Our goal was to decide whether the k-$\oplus_t$Set Cover instance $(\mathcal{F}', U')$ (for $k' = kq + z$) has odd number of set covers,

which means that we want to control the correspondence between the parity of the number of solutions in each part of the construction. Observe that the only step of the construction which has nontrivial correspondence between the number of solutions of the former and the latter instance is the grouping step where we transform an instance $(\mathcal{F}_0, U_0, j_0)$ into a multiset instance $(\mathcal{F}^*, U^*, t^*)$.

Hence we assume that we know the parity of the number of set covers of size $t^* = j_0/q$ in $(\mathcal{F}^*, U^*)$ as well as the parity of the number of set covers of size j' for each $j' < j_0$ in $(\mathcal{F}_0, U_0)$. Our objective is to compute the parity of the number of set covers of size j_0 in $(\mathcal{F}_0, U_0)$ in polynomial time and for this reason we introduce a few definitions and lemmas. Recall that each set in $\mathcal{F}^*$ corresponds to a union of exactly q sets in $\mathcal{F}_0$ and let $\Gamma\colon \mathcal{F}^* \to 2^{\mathcal{F}_0}$ be a function that for each set in $\mathcal{F}^*$ assigns a family of exactly q sets from $\mathcal{F}_0$ that it was made of. Moreover let $\mathcal{S}^* \subseteq 2^{\mathcal{F}^*}$ be the family of set covers of size t^* in $(\mathcal{F}^*, U^*)$ and let $\mathcal{S}_0 \subseteq 2^{\mathcal{F}_0}$ be the set of set covers of size *at most* j_0 in $(\mathcal{F}_0, U_0)$. We construct a mapping $\Phi\colon \mathcal{S}^* \to \mathcal{S}_0$ which maps each set cover $A \in \mathcal{S}^*$ to a set cover $A_0 \in \mathcal{S}_0$ such that A_0 is exactly the set of sets from $\mathcal{F}_0$ used in the t^* unions of q sets from $\mathcal{F}_0$, that is $\Phi(A) = \bigcup_{X \in A} \Gamma(X)$. In the following lemma we prove that for a set cover $A_0 \in \mathcal{S}_0$ the size of $\Phi^{-1}(A_0)$ depends solely on the size of A_0.

Lemma 4.10. *Let $A_0, B_0 \in \mathcal{S}_0$ such that $|A_0| = |B_0|$. Then $|\Phi^{-1}(A_0)| = |\Phi^{-1}(B_0)|$.*

Proof. Let $A_0 = \{X_1, \ldots, X_a\}$ be a set from $\mathcal{S}_0$, where each $X_i \in \mathcal{F}_0$. Observe that for any $A \in \mathcal{S}^*$ we have $\Phi(A) = A_0$ if and only if $\bigcup_{i=1}^{a} \Gamma(X_i) = A$. Consequently $|\Phi^{-1}(A_0)|$ is equal to the number of set covers of size t^* in the set system $(\binom{A_0}{q}, A_0)$ and hence $|\Phi^{-1}(A_0)|$ depends only on the size of A_0. □

Now we prove that for each set cover $A_0 \in \mathcal{S}_0$ of size j_0 an odd number of set covers from $\mathcal{S}^*$ is mapped by Φ to A_0.

Lemma 4.11. *For any nonnegative integers a, b such that $b \leq a$ the binomial coefficient $\binom{a}{b}$ is odd if and only if* $\mathrm{ones}(b) \subseteq \mathrm{ones}(a)$*, where* $\mathrm{ones}(x)$ *is the set of indices containing ones in the binary representation of x.*

Proof. For a nonnegative integer x by $f(x)$ let us denote the greatest integer i such that $x!$ is divisible by 2^i, that is

$$\begin{aligned} f(x) &= \sum_{i \geq 1} \left\lfloor \frac{x}{2^i} \right\rfloor \\ &= \Big(\sum_{i \geq 1} \frac{x}{2^i}\Big) - \frac{1}{2} \cdot \left|\left\{i \geq 1 : \left\lfloor \frac{x}{2^{i-1}} \right\rfloor \text{ is odd}\right\}\right| \\ &= \Big(\sum_{i \geq 1} \frac{x}{2^i}\Big) - \frac{|\mathrm{ones}(x)|}{2} \end{aligned}$$

Since $\binom{a}{b} = \frac{a!}{b!(a-b)!}$ we infer that $\binom{a}{b}$ is odd if and only if $f(a) = f(b) + f(a-b)$, which by the above formula is equivalent to $|\mathrm{ones}(a)| = |\mathrm{ones}(b)| + |\mathrm{ones}(a-b)|$. However for any nonnegative integers x, y we have $\mathrm{ones}(x+y) \leq \mathrm{ones}(x) + \mathrm{ones}(y)$ and moreover $\mathrm{ones}(x+y) = \mathrm{ones}(x) + \mathrm{ones}(y)$ if and only if there are no carry-operations when adding x to y, which is equivalent to $\mathrm{ones}(x) \cap \mathrm{ones}(y) = \emptyset$.

Therefore by setting $x = b$ and $y = a - b$ we infer that $\binom{a}{b}$ is odd if and only if $\mathrm{ones}(b) \cap \mathrm{ones}(a-b) = \emptyset$ which is equivalent to $\mathrm{ones}(b) \subseteq \mathrm{ones}(a)$ and the lemma follows. □

Lemma 4.12. *Let $A_0 \in \mathcal{S}_0$ such that $|A_0| = j_0$ then $|\Phi^{-1}(A_0)|$ is odd.*

Proof. Since $|\Phi^{-1}(A_0)|$ is equal to the number of set covers of size t^* in the set system $(\binom{A_0}{q}, A_0)$ and $|A_0| = j_0 = t^*q$ we infer that $|\Phi^{-1}(A_0)|$ is equal to the number of unordered partitions of A_0 into sets of size q. Hence $|\Phi^{-1}(A_0)| = \prod_{i=0}^{t^*-1} \binom{j_0-1-iq}{q-1}$. Since j_0 is divisible by q and q is a power of two using Lemma 4.11 we have $|\Phi^{-1}(A_0)| \equiv 1 \pmod 2$. □

For $j = 1, \ldots, j_0$ by s_j let us denote the parity of the number of set covers of $(\mathcal{F}_0, U_0)$ of size j modulo 2. Recall that we know the value of s_j for each $j < j_0$ and we want to compute s_{j_0} knowing also $|\mathcal{S}^*| \bmod 2$. By Lemma 4.10 we can define d_j for $j = 1, \ldots, j_0$, that is the value of $|\Phi^{-1}(A_0)| \bmod 2$ for a set $A_0 \in \mathcal{S}_0$ of size j. By Lemma 4.12 we know that d_{j_0} equals one. Thus we have the following congruence modulo 2.

$$|\mathcal{S}^*| \quad = \quad \sum_{A_0 \in \mathcal{S}_0} |\Phi^{-1}(A_0)| \equiv \sum_{j=1}^{j_0} s_j d_j = s_{j_0} + \sum_{j=1}^{j_0-1} s_j d_j \,.$$

Hence knowing $|\mathcal{S}^*| \bmod 2$ and all values s_j for $j < j_0$ in order to compute s_{j_0} it is enough to compute all the values d_j, what we can do in polynomial time thanks to the following lemma.

Lemma 4.13. *For each $j = 1, \ldots, j_0$ we can calculate the value of d_j in polynomial time.*

Proof. Again we use that fact that for a set $A_0 \in \mathcal{S}_0$ we have that $|\Phi^{-1}(A_0)|$ is equal to the number set covers of size t^* in the set system $(\binom{A_0}{q}, A_0)$. Using the inclusion-exclusion principle modulo two we obtain the following formula when $|A_0| = j$.

$$|\Phi^{-1}(A_0)| \equiv \sum_{X \subseteq A_0} \left| \left\{ \mathcal{H} \subseteq \binom{X}{q} \middle| |\mathcal{H}| = t^* \right\} \right| = \sum_{i=0}^{j} \binom{j}{i} \binom{\binom{i}{q}}{t^*},$$

Where the second equality follows by grouping all summands $X \subseteq A_0$ with $|X| = i$ for every $0 \leq i \leq |A_0|$. □

Consequently by solving a polynomial of n number of instances of the k'-$\oplus_t$Set Cover problem with universe size bounded by $n+q+z$ and set family size bounded by $(cn+q)^q+1$ we verify whether the initial set system $\mathcal{F}' \subseteq 2^{U'}$ has an odd number of set covers, which finishes the proof of Theorem 4.5. □

5 Summary and Open Problems

We have shown that the exponential time complexity of a number of basic problems is strongly interconnected. Specifically, our results imply that the optimal growth rates of a a number of problems are in fact asymptotically equal. Assuming SETH, our results imply tight lower bounds on the growth rates for a number of search problems whose growth rates are achieved by naïve brute force algorithms. For problems solvable by dynamic programming, we gave tight lower bounds assuming that the optimal growth rate of Set Cover is achieved by its known dynamic programming algorithm. Finally, we connected the two types of results by showing that SETH implies tight lower bounds on the optimal growth rates of corresponding parity variants. We conclude our work with some open problems.

1. Is it possible to rule out an algorithm for Set Cover with running time $2^{\epsilon n} m^{O(1)}$, $\epsilon < 1$, assuming SETH?
2. Is it possible to rule out an algorithm for Graph Coloring with running time $2^{\epsilon n}$, $\epsilon < 1$, assuming SETH? What about a lower bound for Graph Coloring under the assumption that there does not exist a $\delta < 1$ such that Set Cover with sets of size at most k has a $O(2^{\delta n} m^{O(1)})$ time algorithm for every k?

3. Is it possible to rule out an algorithm that *counts* the number of proper c-colorings of an input graph in time $2^{\epsilon n}$, $\epsilon < 1$ assuming $\oplus$-SETH?
4. Assuming SETH, is it possible to rule out an algorithm with running time $2^{\epsilon n} n^{O(1)}$, $\epsilon < 1$ for the satisfiability of circuits with at most cn gates of *unbounded fan in*, for a concrete constant c?
5. Assuming SETH, is it possible to rule out an algorithm with running time $O(c^n)$ for 3-CNF-Sat for a concrete constant c?

References

[Bel62] Richard Bellman, "Dynamic programming treatment of the travelling salesman problem," *Journal of the ACM*, vol. 9, no. 1, pp. 61–63, 1962. DOI: 10.1145/321105.321111.

[BHK+07] Andreas Björklund, Thore Husfeldt, Petteri Kaski, and Mikko Koivisto, "Fourier meets Möbius: Fast subset convolution," in *Proceedings of the 39th ACM Symposium on Theory of Computing, STOC 2007*, 2007, pp. 67–74. DOI: 10.1145/1250790.1250801.

[BHK09] Andreas Björklund, Thore Husfeldt, and Mikko Koivisto, "Set partitioning via inclusion-exclusion," *SIAM Journal on Computing*, vol. 39, no. 2, pp. 546–563, 2009. DOI: 10.1137/070683933.

[Cal08] Chris Calabro, "A lower bound on the size of series-parallel graphs dense in long paths," Electronic Colloquium on Computational Complexity (ECCC), Tech report TR08-110, 2008. [Online]. Available: http://eccc.hpi-web.de/eccc-reports/2008/TR08-110/.

[CCF+05] Jianer Chen, Benny Chor, Mike Fellows, Xiuzhen Huang, David W. Juedes, Iyad A. Kanj, and Ge Xia, "Tight lower bounds for certain parameterized NP-hard problems," *Information and Computing*, vol. 201, no. 2, pp. 216–231, 2005. DOI: 10.1016/j.ic.2005.05.001.

[CIK+03] Chris Calabro, Russell Impagliazzo, Valentine Kabanets, and Ramamohan Paturi, "The complexity of unique k-SAT: An isolation lemma for k-CNFs," in *Proceedings of the 18th Annual IEEE Conference on Computational Complexity, CCC 2003*, 2003, p. 135. DOI: 10.1109/CCC.2003.1214416.

[CIP06] Chris Calabro, Russell Impagliazzo, and Ramamohan Paturi, "A duality between clause width and clause density for SAT," in *Proceedings of the 21th Annual IEEE Conference on Computational Complexity, CCC 2006*, 2006, pp. 252–260. DOI: 10.1109/CCC.2006.6.

[CIP09] ——, "The complexity of satisfiability of small depth circuits," in *Proceedings of the 4th International Workshop on Parameterized and Exact Computation, IWPEC 2009*, 2009, pp. 75–85. DOI: 10.1007/978-3-642-11269-0_6.

[CLR+09] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, *Introduction to algorithms*, Third. MIT Press, 2009, ISBN: 978-0-262-03384-8.

[CNP+11] Marek Cygan, Jesper Nederlof, Marcin Pilipczuk, Michal Pilipczuk, Johan M. M. van Rooij, and Jakub Onufry Wojtaszczyk, "Solving connectivity problems parameterized by treewidth in single exponential time," in *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science, FOCS 2011*, 2011, pp. 150–159. DOI: 10.1109/FOCS.2011.23.

[DHM+12] Holger Dell, Thore Husfeldt, Dániel Marx, Nina Taslaman, and Martin Wahlén, "Exponential time complexity of the permanent and the Tutte polynomial," *ACM Transactions on Algorithms*, 2012+, To appear.

[FGK09] Fedor V. Fomin, Fabrizio Grandoni, and Dieter Kratsch, "A measure & conquer approach for the analysis of exact algorithms," *Journal of the ACM*, vol. 56, no. 5, 2009. DOI: 10.1145/1552285.1552286.

[FKW04] Fedor V. Fomin, Dieter Kratsch, and Gerhard J. Woeginger, "Exact (exponential) algorithms for the dominating set problem," in *Proceedings of the 30th International Workshop on Graph-Theoretic Concepts in Computer Science, WG 2004*, 2004, pp. 245–256. DOI: 10.1007/978-3-540-30559-0_21.

[HK62] Michael Held and Richard M. Karp, "A dynamic programming approach to sequencing problems," *Journal of the Society for Industrial and Applied Mathematics*, vol. 10, no. 1, pp. 196–210, 1962. DOI: 10.1145/800029.808532.

[IP01] Russell Impagliazzo and Ramamohan Paturi, "On the complexity of k-SAT," *Journal of Computer and System Sciences*, vol. 62, no. 2, pp. 367–375, 2001. DOI: 10.1006/jcss.2000.1727.

[IPZ01] Russell Impagliazzo, Ramamohan Paturi, and Francis Zane, "Which problems have strongly exponential complexity?," *Journal of Computer and System Sciences*, vol. 63, no. 4, pp. 512–530, 2001. DOI: 10.1006/jcss.2001.1774.

[KLR09] Joachim Kneis, Alexander Langer, and Peter Rossmanith, "A fine-grained analysis of a simple independent set algorithm," in *Proceedings of the IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS2009*, 2009, pp. 287–298. DOI: 10.4230/LIPIcs.FSTTCS.2009.2326.

[LMS11] Daniel Lokshtanov, Dániel Marx, and Saket Saurabh, "Slightly superexponential parameterized problems," in *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011*, 2011, pp. 760–776. [Online]. Available: http://www.siam.org/proceedings/soda/2011/SODA11_059_lokshtanovd.pdf.

[Mar07] Dániel Marx, "On the optimality of planar and geometric approximation schemes," in *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2007*, 2007, pp. 338–348. DOI: 10.1109/FOCS.2007.50.

[Ned09] Jesper Nederlof, "Fast polynomial-space algorithms using Möbius inversion: Improving on Steiner tree and related problems," in *Proceedings of the 36th Internationcal Colloquium on Automata, Languages and Programming, ICALP 2009*, 2009, pp. 713–725. DOI: 10.1007/978-3-642-02927-1_59.

[NR10] Jesper Nederlof and Johan M. M. van Rooij, "Inclusion/exclusion branching for partial dominating set and set splitting," in *Proceedings of the 5th International Symposium on Parameterized and Exact Computation, IPEC 2010*, 2010, pp. 204–215. DOI: 10.1007/978-3-642-17493-3_20.

[RND09] Johan M. M. van Rooij, Jesper Nederlof, and Thomas C. van Dijk, "Inclusion/exclusion meets measure and conquer," in *Proceedings of the 17th Annual European Symposium on Algorithms, ESA 2009*, 2009, pp. 554–565. DOI: 10.1007/978-3-642-04128-0_50.

[Rob86] J. M. Robson, "Algorithms for maximum independent sets," *Journal of Algorithms*, vol. 7, no. 3, pp. 425–440, 1986. DOI: 10.1016/0196-6774(86)90032-5.

[Sch05] Rainer Schuler, "An algorithm for the satisfiability problem of formulas in conjunctive normal form," *Journal of Algorithms*, vol. 54, no. 1, pp. 40–44, 2005. DOI: 10.1016/j.jalgor.2004.04.012.

[SS11] Rahul Santhanam and Srikanth Srinivasan, "On the limits of sparsification," Electronic Colloquium on Computational Complexity (ECCC), Tech report TR11-131, 2011. [Online]. Available: http://eccc.hpi-web.de/eccc-reports/2011/TR11-131/.

[Tra08] Patrick Traxler, "The time complexity of constraint satisfaction," in *Proceedings of the 3rd International Workshop on Parameterized and Exact Computation, IWPEC 2008*, 2008, pp. 190–201. DOI: 10.1007/978-3-540-79723-4_18.

[Val77] Leslie G. Valiant, "Graph-theoretic arguments in low-level complexity," in *Proceedings of the 6th Symposium on Mathematical Foundations of Computer Science, MFCS 1977*, 1977, pp. 162–176. DOI: 10.1007/3-540-08353-7_135.

[Vio09] Emanuele Viola, "On the power of small-depth computation," *Foundations and Trends in Theoretical Computer Science*, vol. 5, no. 1, pp. 1–72, 2009. DOI: 10.1561/0400000033.

[vRo11] Johan M. M. van Rooij, "Exact exponential-time algorithms for domination problems in graphs," PhD thesis, Utrecht University, Jun. 2011.

[Wil11] Ryan Williams, "Non-uniform ACC circuit lower bounds," in *Proceedings of the 26th Annual IEEE Conference on Computational Complexity, CCC 2011*, 2011, pp. 115–125. DOI: 10.1109/CCC.2011.36.

A Problem definitions

⊕Bipartite Independent Set

Input A bipartite graph $(A \cup B, E)$ where $|A| = n$, $|B| = m$.

Question Is the number of independent sets odd?

k-CNF-Sat

Input A CNF formula consisting of m clauses of size at most k and on n variables.

Question Is there a satisfying assignment?

⊕CNF-Sat

Input A CNF formula consisting of m clauses of size at most k and on n variables.

Question Is the number of satisfying assignments odd?

Connected Vertex Cover

Input An integer t and a graph $G = (V, E)$.

Question Is there a subset $X \subseteq V$ such that $|X| \leq t$, $X \cap e \neq \emptyset$ for every $e \in E$ and $G[X]$ is connected?

$\oplus_t$Connected Vertex Cover

Input An integer t and a graph $G = (V, E)$.

Question Is the number of subsets $X \subseteq V$ such that $|X| = t$, $X \cap e \neq \emptyset$ for every $e \in E$ and $G[X]$ is connected, odd?

Hitting Set

Input An integer t and a set system $\mathcal{F} \subseteq 2^U$ where $|\mathcal{F}| = m$, $|U| = n$.

Question Is there a subset $H \subseteq U$ with $|H| \leq t$ such that $H \cap S \neq \emptyset$ for every $S \in \mathcal{F}$?

⊕Hitting Set

Input A set system $\mathcal{F} \subseteq 2^U$ where $|\mathcal{F}| = m$, $|U| = n$.

Question Is the number of $H \subseteq U$ such that $H \cap S \neq \emptyset$ for every $S \in \mathcal{F}$ odd?

k-NAE-Sat

Input A CNF formula consisting of m clauses of size at most k and on n variables.

Question Is there an assignment so that each clause contains literal set to true and a literal set to false?

k-Set Cover

Input An integer t and a set system $\mathcal{F} \subseteq 2^U$ where $|\mathcal{F}| = m$, $|U| = n$ and for every $S \in \mathcal{F}$, $|S| \leq k$.

Question Is there a subset $\mathcal{C} \subseteq \mathcal{F}$ with $|\mathcal{C}| \leq t$ such that $\bigcup_{S \in \mathcal{C}} S = U$?

k-$\oplus_t$Set Cover

Input An integer t and a set system $\mathcal{F} \subseteq 2^U$ where $|\mathcal{F}| = m$, $|U| = n$, for every $S \in \mathcal{F}$, $|S| \leq k$.

Question Is the number of $\mathcal{C} \subseteq \mathcal{F}$ with $|\mathcal{C}| = t$ such that $\bigcup_{S \in \mathcal{C}} S = U$ odd?

k-Set Partitioning

Input An integer t and a set system $\mathcal{F} \subseteq 2^U$ where $|\mathcal{F}| = m$, $|U| = n$, for every $S \in \mathcal{F}$, $|S| \leq k$.

Question Is there a subset $\mathcal{C} \subseteq \mathcal{F}$ with $|\mathcal{C}| = t$ such that $\bigcup_{S \in \mathcal{C}} S = U$ and for every $S, S' \in \mathcal{F}$ with $S \neq S'$, $S \cap S' = \emptyset$?

(k,c)-Sparse-CNF-Sat

Input A CNF formula consisting of m clauses of size at most k and on n variables, where $m \leq cn$.

Question Is there a satisfying assignment?

(k,c)-$\oplus$Sparse-CNF-Sat

Input A CNF formula consisting of m clauses of size at most k and on n variables, where $m \leq cn$.

Question Is the number of satisfying assignments odd?

(k,c)-Sparse-Hitting Set

Input An integer t and a set system $\mathcal{F} \subseteq 2^U$ where $|\mathcal{F}| = m \leq cn$, $|U| = n$ and for every $S \in \mathcal{F}$, $|S| \leq k$.

Question Is there a subset $H \subseteq U$ with $|H| \leq t$ such that $H \cap S \neq \emptyset$ for every $S \in \mathcal{F}$?

(k,c)-$\oplus$Sparse-Hitting Set

Input A set system $\mathcal{F} \subseteq 2^U$ where $|\mathcal{F}| = m \leq cn$, $|U| = n$ and for every $S \in \mathcal{F}$, $|S| \leq k$.

Question Is the number of subsets $H \subseteq U$ such that $H \cap S \neq \emptyset$ for every $S \in \mathcal{F}$, odd?

(k,c)-Sparse-Set Cover

Input An integer t and a set system $\mathcal{F} \subseteq 2^U$ where $|\mathcal{F}| = m$, $|U| = n$, for every $S \in \mathcal{F}$, $|S| \leq k$ and $m \leq cn$.

Question Is there a subset $\mathcal{C} \subseteq \mathcal{F}$, such that $|\mathcal{C}| \leq t$ and $\bigcup_{S \in \mathcal{C}} S = U$?

(k,c)-$\oplus$Sparse-Set Cover

Input A set system $\mathcal{F} \subseteq 2^U$ where $|\mathcal{F}| = m$, $|U| = n$, for every $S \in \mathcal{F}$, $|S| \leq k$ and $m \leq cn$.

Question Is the number of $\mathcal{C} \subseteq \mathcal{F}$ with $\bigcup_{S \in \mathcal{C}} S = U$ odd?

k-Set Splitting

Input A set system $\mathcal{F} \subseteq 2^U$ where $|\mathcal{F}| = m$, $|U| = n$, for every $S \in \mathcal{F}$, $|S| \leq k$.

Question Is there a subset $X \subseteq U$ such that, for every $S \in \mathcal{F}$, neither $S \subseteq X$ nor $S \subseteq (U - X)$?

Steiner Tree

Input An integer t and a graph $G = (V, E)$ with terminals $T \subseteq V$.
Question Is there a subset $T \subseteq X \subseteq V$ with $|X| \leq t$ and $G[X]$ connected?

$\oplus_t$Steiner Tree
Input An integer t and a graph $G = (V, E)$ with terminals $T \subseteq V$.
Question Is the number of subsets $T \subseteq X \subseteq V$ with $|X| = t$ and $G[X]$ connected, odd?

Subset Sum
Input Integers $a_1, \ldots, a_n \in \mathbb{Z}_+$ and a target integer t on m bits.
Question Is there a subset $X \subseteq \{1, \ldots, n\}$ with $\sum_{i \in X} a_i = t$?

c-VSP-Circuit-SAT
Input A cn-size Valiant series-parallel circuit over n variables.
Question Is there a satisfying assignment?