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‡‡‡

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 $O(2^{\varepsilon n})$, 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 $\varepsilon < 1$, there is a (large) integer $k$ such that that $k$-CNF-SAT cannot be computed in time $2^{\varepsilon n}$.

In this paper, we show that, for every $\varepsilon < 1$, the problems HITTING SET, SET SPLITTING, and NAE-SAT cannot be computed in time $O(2^{\varepsilon n})$ 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 set covers cannot be computed in time $O(2^{\varepsilon n})$ for any $\varepsilon < 1$ unless SETH fails.

Keywords—Strong Exponential Time Hypothesis, Exponential Time Algorithms, Sparsification Lemma

I. 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 [1, 15] 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 [13, 18, 23, 29], 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 [3, 21]. 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.

A. 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” [16]. More
precisely, the hypothesis asserts that 3-CNF-SAT cannot be computed in time $2^{n^{o(n)}}$, 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 [17], parameterized problems [8, 19], approximation problems [20], and counting problems [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 $2^{n^{o(n)}}\text{poly}(m)$ [25, 31], 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 $2^{n^{o(n)}}\text{poly}(m)$ 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 [17]. In fact, very recent results indicate that such a sparsification may be impossible for general CNF formulas [24]. For this reason, we consider the Strong Exponential Time Hypothesis (SETH) of Impagliazzo and Paturi [7, 16, 17]. 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 $O(2^n)$. 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.

B. 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) $\forall \epsilon < 1.3k$. $k$-CNF-SAT, the satisfiability problem for $n$-variable $k$-CNF formulas, cannot be computed in time $O(2^n)$.  

(ii) $\forall \epsilon < 1.3k$. $k$-HITTING SET, the hitting set problem for $n$-variable $k$-CNF formulas, cannot be computed in time $O(2^n)$.  

(iii) $\forall \epsilon < 1.3k$. $k$-SET SPLITTING, the set splitting problem for $n$-variable $k$-CNF formulas, cannot be computed in time $O(2^n)$.  

(iv) $\forall \epsilon < 1.3k$. $k$-NAE-SAT, the not-all-equal assignment problem for $n$-variable $k$-CNF formulas, cannot be computed in time $O(2^n)$.

(v) $\forall \epsilon < 1.3k$. $\epsilon$-VSP-CIRCUIT-SAT, the satisfiability problem for $n$-variable series-parallel circuits of size at most $cn$, cannot be computed in time $O(2^n)$.  

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 $2^{n^{o(n)}}\text{poly}(m)$ 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 $2^{m^{o(n)}}\text{poly}(m)$. 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 [14] can solve SET COVER in time $2^{m^{o(n)}}\text{poly}(m)$. 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 $2^{n^{o(n)}}\text{poly}(m)$. Then, for all $\epsilon < 1$, we have:

(i) **Steiner Tree** cannot be computed in time $2^{n^{o(n)}}\text{poly}(m)$.  

(ii) **Connected Vertex Cover** cannot be computed in time $2^{n^{o(n)}}\text{poly}(m)$.  

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

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

All problems mentioned in this theorem have non-trivial algorithms whose running times are as above with $\epsilon = 1$ [2, 9, 11, 14, 21]. 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 I.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 [11] for CONNECTED VERTEX COVER works by reducing the problem to computing the number of solutions modulo two.

While Theorem I.1 is an equivalence, Theorem I.2 is not. One might ask whether it is possible to find reductions back to SET COVER and to strengthen Theorem I.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 I.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 [2, 21] 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 [11].

The hardness assumption for SET COVER in Theorem I.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 I.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 \( 2^n \text{poly}(m) \)-time algorithm to compute the number of set covers modulo two would violate SETH, which is a hypothesis that implies SETH. Formally, SETH asserts that, for all \( \epsilon < 1 \), there exists a (large) integer \( k \) such that \( k \text{-CNF-SAT} \) cannot be computed in time \( O(2^{\epsilon n}) \). Here, \( k \text{-CNF-SAT} \) is the problem of computing the number of satisfying assignments of a given \( k \)-CNF formula modulo two. It follows from known results [5, 26] (see also Section III-A) that, if SETH holds, then so does SETH. As a partial justification for the hardness assumption for SET COVER in Theorem I.2, we provide the following theorem.

**Theorem I.3.** Each of the following statements is equivalent to SETH:

(i) \( \forall \epsilon < 1.3k. \ k \text{-CNF-SAT} \), the parity satisfiability problem for \( n \)-variable \( k \)-CNF formulas, cannot be computed in time \( O(2^{\epsilon n}) \).

(ii) \( \forall \epsilon < 1.3k. \ k \text{-HITTING SET} \), 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) \( \forall \epsilon < 1.3k. \ k \text{-SET COVER} \), 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 I.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 I.3 that we find interesting is that SETH rules out the existence of \( 2^n \text{poly}(m) \)-time algorithms to compute the number of set covers of a set system, for any \( \epsilon < 1 \).

Theorem I.3 together with the fact that the algorithms for all problems mentioned in Theorem I.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 is indeed possible. In particular we show that, assuming SETH, there is no \( 2^n \text{poly}(n) \)-time algorithm that computes the parity of the number of Steiner trees that have size at most \( t \), and no \( 2^n \text{poly}(n) \)-time algorithm that computes the parity of the number of connected vertex covers that have size at most \( t \). Thus, unless 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 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 I.2. This conjecture provides an interesting open problem.

Our results are obtained by a collection of reductions. Section III contains the reductions that constitute the proof of Theorem I.1, and some of the reductions needed for Theorem I.3. Section IV contains the proof of Theorem I.2, the remaining reductions for Theorem I.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.

**II. Preliminaries and Notation**

In this paper, \( \Delta \) denotes the symmetric difference and \( \cup \) 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 \( \alpha: \{v_1, \ldots, v_n\} \rightarrow \{0, 1\} \) 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 $k$-$\Pi$ 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-$\Pi$ 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 $\Pi$ that we consider, we fix the canonical NP-verifier that is implicit in the way we define the problem. Then every yes-instance of $\Pi$ 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 number of solutions is odd. If solutions of $\Pi$ are sets (e.g., of vertices), we write $\oplus \Pi$ 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 $\Pi/n$. For every such parameterized problem, we now define the number $\sigma = \sigma(\Pi/n)$.

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

The optimal growth rate of $\Pi$ with respect to $n$ is $C := \sigma^2(\Pi/n)$. If the infimum in the definition above is a minimum, then $\Pi$ has an algorithm that runs in time $C^n \text{poly}(m)$ and no algorithm for $\Pi$ can have a running time $\epsilon^n \text{poly}(m)$ for any $\epsilon < C$. On the other hand, if the minimum does not exist, then no algorithm for $\Pi$ can run in time $C^n \text{poly}(m)$, but $\Pi$ has a $\epsilon^n \text{poly}(m)$-time algorithm for every $\epsilon > C$. 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, $k^{k^11.99^n}$-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 $\epsilon < 1$.

**III. 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 $F \subseteq 2^U$ with $|F| = m$ 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 $S \in F$. With the same input, the SET SPLITTING problem asks whether there is a subset $X \subseteq U$ such that, for every $S \in F$, we have $S \not\subseteq X$ and $S \not\subseteq (U \setminus X)$.

**A. 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 IV 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. [17] is that every $k$-CNF formula $\varphi$ can be written as the disjunction of $2^\alpha$ 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 $\varphi$ and $\epsilon$ in time $2^{c(m)} \cdot \text{poly}(m)$. Hence, the growth rate of $k$-CNF-SAT for formulas of density at most $c(k, \epsilon)$ is $\epsilon$-close to the growth rate of general $k$-CNF-SAT. More precisely, for every $k$ and every $\epsilon > 0$, we have $\sigma((c,k)\text{-SPARSE-CNF-SAT}/n) \leq \sigma(k\text{-CNF-SAT}/n) \leq \sigma((c,k)\text{-SPARSE-CNF-SAT}/n) + \epsilon$, 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 $c(k, \epsilon) = (k/e)^3k$ [6]. By setting $\epsilon = c(k, \omega(1))$, this immediately yields the following theorem.

Theorem III.1 ([6, 17]). For every function $c = c(k) \geq (\omega(k))^3k$, we have

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

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

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

We define $\oplus$-SETH as the assertion that these limits are equal to 1. The isolation lemmas for $k$-CNF formulas [5, 26] immediately yield that SETH implies $\oplus$-SETH. More precisely, we have the following theorem.

Theorem III.2 ([5, 26]).

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

B. From CNF-SAT to Hitting Set

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

1. Let $p' = \text{odd integer } p' = p + 2\lceil \log_2 p \rceil$, and let $U = \{u_1, \ldots, u_n\}$ with $n' = p' \cdot n/p$.
2. Partition the variables of $\varphi$ into blocks $V_i$ of size $p$, i.e., $V_i := \{v_{pi+1}, \ldots, v_{pi+1}\}$.
3. Partition $U$ into blocks $U_i$ of size $p'$, i.e., $U_i := \{u_{pi+1}, \ldots, u_{pi+1}\}$.
4. Choose an arbitrary injective function $\psi_i : 2^{V_i} \to \{\{U_i\}/2\}$. This exists since $\{\{U_i\}/2\} = (\{U_i\}/2)^2 \geq 2^{p'n} \geq 2^{2p'n} \geq 2^{p'p'} \geq 2^{p'p'/2} \geq 2^{p'p'/2} \geq 2^p = |2^U|$. We think of $\psi_i$ as a mapping that, given an assignment to the variables of $V_i$, associates it with a subset of $U_i$ of size $\lceil p'/2 \rceil$.
5. If $X \in \{\{U_i\}/2\}$ for some $i$, then add the set $X$ to $F_{p,p}$. If $X \in \{\{U_i\}/2\}$ for some $i$ such that $\psi_i^{-1}(U_i \setminus X) = \emptyset$, then add the set $X$ to $F_{p,p}$.
6. For every clause $\varphi_i$ of $\varphi$, let $A_i$ be the set $\{A \in \{U_i\}/2 \mid A \cap \{\psi_i^{-1}(\{U_i \setminus A\})\}$ sets all variables in $C \cap V_i\}$. Then $F_{p,p}$ is a hitting set of $\varphi$.

Lemma III.3. For every $n$-variable CNF formula $\varphi$ and every odd integer $p \geq 3$ that divides $n$, the number of satisfying assignments of $\varphi$ is equal to the number of hitting sets of size $\lfloor \frac{p}{2^k} \rfloor$ of the set system $F_{p,p}$, where $p' = p + 2\lceil \log_2 p \rceil$.

Proof: For convenience denote $g = \frac{n}{p}$. Define $\psi : 2^V \to 2^U$ as $\psi(A) = \bigcup_{i=1}^p \psi_i(A \cap V_i)$. Note that $\psi$ is injective, since for every $i$, $\psi_i$ is injective. Hence, the number of hitting sets of $\varphi$ is equal to the number of satisfying assignments of $\varphi$. Therefore, if there is no assignment $A \subseteq V$ such that $\psi(A) = H$, then no set $H \subseteq U$ of size $\lfloor \frac{p}{2^k} \rfloor$ is a hitting set of $F_{p,p}$.

For the forward direction of (1), note that the sets added in Step 5 are hit by the pigeon-hole principle since $|\psi_i(A \cap V_i)| = \lfloor \frac{p}{2^k} \rfloor$ and $p'$ is odd. For the sets added in Step 6, consider the following. The set $X$ of size $\lfloor \frac{p}{2^k} \rfloor$ 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 $\varphi$ and the associated index set $I$ as in Step 7. Since $A$ is a satisfying assignment of $\varphi$, there exists $i \in I$ such that $A$ sets at least one variable in $C \cap V_i$ true. Hence, $U_i \setminus \psi_i(A \cap V_i) \not\supseteq A_i$. On the other hand, $U_i \setminus \psi_i(A \cap V_i)$ is the only member of $F_{p,p}$ 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 $\lfloor \frac{p}{2^k} \rfloor$ 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 $\lfloor \frac{p}{2^k} \rfloor$. We show that $A$ is a satisfying assignment of $\varphi$. 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, $|\psi(A) \cap U_i| \geq \lfloor \frac{p}{2^k} \rfloor$ for every $i$ because it hits all sets added in Step 5. More precisely, $|\psi(A) \cap U_i| = \lfloor \frac{p}{2^k} \rfloor$ because $|\psi(A)| = \lfloor \frac{p}{2^k} \rfloor$ and there are $g$ disjoint blocks $U_1, \ldots, U_g$. Therefore, $|U_i \setminus \psi(A)| = \lfloor \frac{p}{2^k} \rfloor$, and so $U_i \cap \psi(A) = U_i \setminus \psi(A)$ is a member of $A_i$ 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 \( \lceil \log_2 n \rceil \) 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 \( F_{\varphi,p} \). 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 \( |H \cap U_i| = \lceil \log_2 n \rceil \) for every \( i \). Since it hits all sets added in Step 6, we also know that \( \psi^{-1}_i(\{H \cap U_i\}) \neq \emptyset \) for every \( i \). However, this contradicts the non-existence of \( A \subseteq V \) such that \( \psi(A) = H \).

**Theorem III.4.** For every function \( c = c(k) \), there exists a function \( c' = c'(k') \) such that

\[
\lim_{k \to \infty} (k,c)\text{-SPARSE-CNF-SAT} = \lim_{k' \to \infty} (k',c')\text{-SPARSE-HITTING SET} + O\left( \frac{\log p}{p} \right).
\]

**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

\[
\sigma((k,c)\text{-SPARSE-CNF-SAT}) \leq \sigma((k',c')\text{-SPARSE-HITTING SET}) + O\left( \frac{\log p}{p} \right).
\]

Create the set system \( F_{\varphi,p} \) as described above. For a constant \( p \), this can clearly be done in polynomial time. We set \( k' = 2p^2 + 2^p c \) (remind that \( p = 2 + 2|\log_2 p| \)). It is easy to see that the maximum size of a set of \( F_{\varphi,p} \) is at most \( k' \). Let \( m' \) be the number of sets in \( F_{\varphi,p} \). Observe that there are at most \( 2^p m' \) sets added in Step 5 and Step 6. Moreover, since each clause contains variables from at most \( k \) blocks, there are at most \( 2^{kp} n \) 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 \( F_{\varphi,p} \) in \( O(2^{\sigma(k',c')\text{-SPARSE-HITTING SET}/n \cdot n^{O(1)}) \) time, where \( n' \) is the size of the universe of \( F_{\varphi,p} \). By Lemma III.3, \( \varphi \) is satisfiable if and only if the size of a minimum hitting set is \( \lceil \log_2 n \rceil \). Since \( n' = n / p \), the theorem follows.

**C. From Hitting Set via Set Splitting to CNF-SAT**

**Theorem III.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 \( \sigma(k\text{-HITTING SET}/n) \leq \sigma(k'\text{-SET SPLITTING}/n) + O\left( \frac{\log(p+1)}{p} \right) \), where \( k' = \max(k+1, p+1) \). Let \( (F, t) \) be an instance of \( k\text{-HITTING SET} \). We can assume that the universe \( U \) of \( F \) has \( n \) elements and that \( p \) divides \( n \). Let \( U = U_1 \cup \ldots \cup U_{n/p} \) 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 \( (p+1)^{n/p} = 2^{n/p \log_2(p+1)} \). For each choice of the \( t_i \)'s, we construct an instance \( F' \) of \( k'\text{-SET SPLITTING} \) as follows.

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

Clearly \( F' \) 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 \( F' \) for every choice of the \( t_i \)'s and we decide HITTING SET in time \( O(2^{(c+1)(k'\text{-SET SPLITTING})/n_2} \cdot n^{O(1)}) \). It remains to show that \( F \) has a hitting set of size at most \( t \) if and only if \( F' \) 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 observe that \( H \cup \{R\} \) and its complement \( U \setminus H \cup \{R\} \) form a set splitting of \( F' \). 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 \( Y \in F \) has a non-empty intersection with \( H \).

For the soundness of the reduction, let \( (S, \overline{S}) \) be a set splitting of \( F' \) 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 \( Y \in F \) has a non-empty intersection with \( S \). Thus, \( S \setminus \{R\} \) is a hitting set of \( F \) and has size at most \( \sum_i t_i = t \).

**Observation III.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 \( F \) be an instance of \( k\text{-SET SPLITTING} \). We construct an equivalent \( k\text{-CNF} \) formula \( \varphi \) as follows. For each element in the universe of \( F \), we add a variable, and for each set \( X \in F \) we add a clause in which each variable occurs positively. A characteristic function of a set splitting \( U = U_1 \cup U_2 \) 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 \( F \) stand in one-to-one correspondence to variable assignments that
satisfy the NAE-SAT constraints of \( \varphi \). Thus, any algorithm for \( k \)-NAE-SAT works for \( k \)-SET SPLITTING, too.

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

D. From Parity CNF-SAT to Parity Hitting Set

Given a CNF formula \( \varphi \) 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 III-B. 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 \( U_j \):
   - add a special element \( e_i \) to the universe,
   - for every \( X \in \binom{U_j}{p/2} \), add the set \( X \cup \{e_i\} \) to the set family.

Lemma III.7. The number of hitting sets of the instance \( \mathcal{F}_{\varphi,p} '' \) of size \( \lceil p/2 \rceil \) 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 \( \lceil p/2 \rceil \) is equal to the number of hitting sets \( H' \) of \( \mathcal{F}_{\varphi,p}' \) such that \( |H' \cap U_i| = \left\lceil \frac{p}{2} \right\rceil \) for every \( 1 \leq i \leq g \). Suppose that \( H \) is a hitting set of \( \mathcal{F}_{\varphi,p}' \) of size \( \lceil p/2 \rceil \) 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 \( |H' \cap U_i| = \left\lceil \frac{p}{2} \right\rceil \) 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 \( |H' \cap U_i| = \left\lceil \frac{p}{2} \right\rceil \) 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 \( H' \). Hence we have a bijection between the two families of hitting sets.

For every hitting set \( H' \) of \( \mathcal{F}_{\varphi,p}'' \) and block \( U_1 \), we know that \( |H' \cap U_1| \geq \lceil p/2 \rceil \). 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 \( |H' \cap U_i| > \left\lceil \frac{p}{2} \right\rceil \) is even. Given such a hitting set \( H' \), let \( \gamma(H') = H' \triangledown \{e_i\} \) where \( i \) is the smallest integer such that \( |H' \cap U_i| > \left\lceil \frac{p}{2} \right\rceil \). Obviously \( \gamma \) is its own inverse and \( |\gamma(H') \cap U_i| > \left\lceil \frac{p}{2} \right\rceil \) 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 \( X \in \binom{U_1 \setminus \{e_i\}}{p/2} \) are hit since \( |\gamma(H') \cap U_i| > \left\lceil \frac{p}{2} \right\rceil \) and that those are the only sets containing \( e_i \).

Theorem III.8. For every function \( c = c(k) \), there exists a function \( c' = c'(k') \) such that

\[
\lim_{k \to \infty} \sigma((k,c) \oplus \text{SPARSE-CNFSAT})/n \leq \lim_{k' \to \infty} \sigma((k',c') \oplus \text{SPARSE-HITTING-SET})/n.
\]

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

\[
\sigma((k,c) \oplus \text{SPARSE-CNFSAT}) \leq \sigma((k',c') \oplus \text{SPARSE-HITTING-SET}) + O\left( \frac{\log p}{p} \right).
\]

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 \((k',c') \oplus \text{SPARSE-HITTING-SET})_n \), where \( k' = p'k \) and \( c' = 2p' + 2 kp' c \). Therefore we can determine the number of hitting sets modulo 2 of \( \mathcal{F}_{\varphi,p}'' \) in \( O(2^n(k,c') \oplus \text{SPARSE-HITTING-SET})_n) \) time, where \( n' \) is the size of the universe of \( \mathcal{F}_{\varphi,p}'' \). Since \( n' = \left\lceil \frac{p}{2} \right\rceil (p + 2 \log p) = n(1 + O(\log p)) \), the theorem follows.

Note that conversely, an improved algorithm for \( \oplus \text{CNFSAT} \) gives an improved algorithm for \( \oplus \text{HITTING-SET} \) given a set family \( F \subseteq U \) the required reduction simply associates a variable with the elements of \( U \) and creates a CNF-formula with for every \( S \subseteq F \) 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 \( \oplus \text{CNFSAT}/m \) and \( \text{SET COVER} \) can be shown.

E. Satisfiability for Series-Parallel Circuits

In this subsection, we show that the satisfiability of \( cn \)-size series-parallel circuits can be decided in time \( 2^m \) 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 [27]. Calabro [4] 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 [4]).

In this section a multidag \( G = (V, E) \) is a directed acyclic multigraph. Let \( \text{input}(G) \) denote the set of vertices \( v \in V \) such that the indegree of \( v \) in \( G \) is zero. Similarly, let \( \text{output}(G) \) 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 \( l: V \to \mathbb{N} \) such that \( \forall (u, v) \in E \), \( l(u) < l(v) \). A labeling \( l \) is normal if for all \( v \in \text{input}(G) \), \( l(v) = 0 \) and there exists
an integer \( d \in \mathbb{N} \) such that for all \( v \in \text{output}(G) \setminus \text{input}(G) \), \( l(v) = d \). 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 [27] and following the arguments by Calabro [4] and Viola [30], we may show the following.

**Theorem III.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 \( 2^{2^{2^{2^{2^d}}}} \) where \( \mu \) 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 III.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 \( k \leq 2^{2^{2^{2^d}}} \) in time \( O(2^{n/d}n^{O(1)}) \). This implies that

\[
\sigma(c\text{-VSP-CIRCUIT-SAT}/n) \leq \sigma(2^{2^{2^d}}\text{-CNF-SAT}/n) + \frac{1}{d}.
\]

Hence, we obtain the following theorem.

**Theorem III.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 III.1, we obtain the following theorem.

**Theorem III.11.**

\[
\lim_{k \to \infty} \sigma(k\text{-CNF-SAT}/n) \leq \lim_{c \to \infty} \sigma(c\text{-VSP-CIRCUIT-SAT}/n).
\]

**IV. 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 \( @\text{HITTING SET}/n \) showed in the previous section. More specifically, we show that \( @\text{HITTING SET}/n \) and \( @\text{SET COVER}/n \) are equivalent using a simple but novel property of bipartite graphs in Subsection IV-A, and in Subsection IV-B we show that the current algorithms for \( @\text{ST MIN TREE}/t \) and \( @\text{ST CONNECTED VERTEX COVER}/k \) are at least as hard to improve as the algorithm for \( @\text{SET COVER}/n \). Motivated we make the hypothesis that the current algorithm for \( @\text{SET COVER} \) can not be improved and show similar implications to the \( @\text{ST MIN TREE} \) and \( @\text{ST CONNECTED VERTEX COVER}/k \), \( @\text{SET PARTITIONING} \) and \( @\text{SUBSET SUM} \) problems.

Given an integer \( t \) and a set system \( F \subseteq 2^U \) where \( |F| = m \) and \( |U| = n \), the \( @\text{SET COVER} \) and \( @\text{SET COVER} \) 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 \( CF \) such that \( \bigcup_{S \in C} U = U \). Given a graph \( G = (V, E) \), with \( |V| = n \) a subset \( T \subseteq V \), and an integer \( t \) the \( @\text{ST MIN TREE} \) and \( @\text{ST CONNECTED VERTEX COVER} \) and \( @\text{ST CONNECTED VERTEX COVER} \) 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 \( X \) induces a connected graph in \( G \).

A. The flip: Parity Hitting Set equals Parity Set Cover

**Lemma IV.1.** Let \( G = (A \cup B, E) \) be a bipartite graph, then the number of independent sets of \( G \) modulo 2 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

\[
\sum_{X \subseteq A} 2^{B \setminus N(X)} = \sum_{X \subseteq A, |B \setminus N(X)| = 0} 2^{|B \setminus N(X)|} = |\{X \subseteq A : N(X) = B\}|,
\]

and the lemma follows.

It is worth mentioning that this lemma was inspired by a non-modular variant from [22, Lemma 2] (see also [28, Proposition 9.1]).

**Theorem IV.2.**

\[
\sigma(@\text{HITTING SET}/n) = \sigma(@\text{SET COVER}/n).
\]

**Proof:** Given a set system \( F \subseteq 2^U \), let \( G = (F, U, E) \) be the bipartite graph where \( (S, e) \in E \) if and only if \( e \in S \).
Note that the number of hitting sets of \( F \) is equal to \(|\{X \subseteq U : N(X) = F}\)|. Then by Lemma IV.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 \( F \) 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 IV.2 the same set system is used as an instance of \( \oplus \)Hitting Set\( /h \) and \( \oplus \)Set Cover\( /h \). Hence the above directly gives the following corollary, which we will need in the next subsection.

**Corollary IV.3.** For every function \( c = c(k) \), there exists a function \( c' = c'(k') \) such that

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

**B. From Set Cover to Steiner Tree and Connected Vertex Cover**

In this subsection we will give reductions from Set Cover\( /h \) to Steiner Tree\( /t \) and Connected Vertex Cover\( /k \). We transfer the reductions to the parity versions \( \oplus \)Set Cover\( /h \), \( \oplus \)Steiner Tree\( /t \), and \( \oplus \)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\( /h \) (and similarly for \( \oplus \)Set Cover\( /h \) and \( \oplus \)Set Cover\( /(n + t) \)). Once we have this intermediate result, the reductions to the \( \oplus \)Steiner Tree\( /t \) and \( \oplus \)Connected Vertex Cover\( /k \) problems follow more easily.

**Theorem IV.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 \( (F, U, t) \) of k-Set Cover into an instance of \( k'-\text{Set Cover} \), for some positive integer \( k' \), where the size \( t' \) of the solution in the resulting \( p'-\text{Set Cover} \) 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 \( \frac{t}{q} \leq \alpha \). 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 \( F \). We form a family \( F' \) of all unions of exactly \( q \) sets from \( F \), that is for each of \( \binom{|F|}{q} \) choices of \( q \) sets \( S_1, \ldots, S_q \in F \) we add to \( F' \) the set \( \bigcup_{i=1}^{q} S_i \). Note that since \( q \) is a constant we can create \( F' \) in polynomial time. We set \( t' = t/q \leq |U|/q \leq \alpha|U| \). It is easy to see that \( (F, U, t) \) is a YES-instance of \( k'-\text{Set Cover} \) if and only if \( (F', U, t') \) is a YES-instance of \( qk'-\text{Set Cover} \).

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 IV.5.** For every function \( c = c(k) \), we have

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

Due to space constraints, we omit the proof in this extended abstract. Using the theorem, we obtain the following results.

**Theorem IV.6.**

\[
\begin{align*}
\lim_{k \to \infty} \sigma(k-\text{Set Cover}/(n + t)) &\leq \sigma(\text{Steiner Tree}/t), \\
\lim_{k \to \infty} \sigma(k-\oplus \text{Set Cover}/(n + t)) &\leq \sigma(\oplus \text{Steiner Tree}/t).
\end{align*}
\]

**Proof:** Given an instance of Set Cover consisting of a set system \( (F, U) \) and integer \( i \), let \( G' \) be the graph obtained from the incidence graph of \( (F, U) \) by adding a vertex \( s \) universal to \( F \) 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 \( (F, U) \) of size \( i \). Hence the theorem follows.

**Theorem IV.7.**

\[
\begin{align*}
\lim_{k \to \infty} \sigma(k-\text{Set Cover}/(n + t)) &\leq \sigma(\text{Connected Vertex Cover}/t), \\
\lim_{k \to \infty} \sigma(k-\oplus \text{Set Cover}/(n + t)) &\leq \sigma(\oplus \text{Connected Vertex Cover}/t).
\end{align*}
\]

**Proof:** Given an instance \( (F, U, t) \) of Set Cover, we create an instance of Connected Vertex Cover with \( G \) being obtained from the incidence graph of \( (F, U) \) 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 \( (F, U) \) 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 \( F \). Hence, by using an algorithm for Connected Vertex Cover,
we obtain an $O(2^{(\text{Connected Vertex Cover} / t)'n O(1)}) = O(2^{(\text{Connected Vertex Cover} / t')|U| + t}n O(1))$ time algorithm for $p \oplus 2$ 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 $C$, $C \cap F$ must be a set cover of $(F, U)$ by the connectivity requirement. Hence we group all connected vertex covers in $G$ depending on which set cover in $(F, U)$ their intersection with $F$ 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 $(F, U)$, then:

$$c_j = \sum_{i=1}^{\lfloor |U| - 1 \rfloor} s_i \left( j - i - |U| - 1 \right).$$

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

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

C. From Set Cover to Set Partitioning and Subset Sum

**Theorem IV.8.**

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

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

**Theorem IV.9.**

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

**Proof:** Let $F \subseteq 2^U$ be an instance of $k \cdot \text{Set Partitioning}$. 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 $t_0$ be the target integer for SUBSET SUM. We 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 $C \subseteq F$; 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 $C$; finally, a field of length $\log_2 n + n$ containing $n$ ones. The paddings of length $\log_2 t_0$ serve to isolate the fields from each other. For every $S_i \in F$, 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 $F$ contains a partitioning of $U$ using exactly $t_0$ sets.

Clearly, if $C \subseteq F$ partitions $U$ and $|C| = t_0$, then the integers $a_i$ corresponding to $S_i \in C$ sum to $t_0$. The first field sums to $t_0$ by cardinality of $C$, 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 $t_0$. 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, $C = \{S_i | a_i \in A\}$ 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.

V. 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 number of problems are in fact asymptotically equivalent. Assuming SETH, our results imply tight lower bounds on the growth rates for a number of search problems whose growth rates are achieved by naive 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 $2^{n^{1+\epsilon}O(1)}$, $\epsilon < 1$, assuming SETH?
2. Is it possible to rule out an algorithm for GRAPH COLORING with running time $2^{n^{1+\epsilon}}$, $\epsilon < 1$, 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 $O(2^{n^{1+\delta}O(1)})$ 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 $2^{cn^{1+\epsilon}}$, $\epsilon < 1$ assuming $\oplus$-SETH?
4. Assuming SETH, is it possible to rule out an algorithm with running time $2^{n^{1+\epsilon}O(1)}$, $\epsilon < 1$ for the satisfiability
of circuits with at most \(cn\) gates of unbounded fan in, for a concrete constant \(c\)?

5) Assuming \(\text{SETH}\), is it possible to rule out an algorithm with running time \(O(c^n)\) for \(3\text{-CNF-SAT}\) for a concrete constant \(c\)?

REFERENCES

2012 IEEE 27th Conference on Computational Complexity

CCC 2012

Table of Contents

Preface.........................................................................................................................................................ix
Organizing Committee.....................................................................................................................................................x
Program Committee....................................................................................................................................................xi
Reviewers............................................................................................................................................................xii
Awards..................................................................................................................................................................xiv

Session 1A

Amplifying Circuit Lower Bounds against Polynomial Time with Applications ..................................................1
Richard J. Lipton and Ryan Williams

Is Valiant-Vazirani's Isolation Probability Improvable? ....................................................................................10
Holger Dell, Valentine Kabanets, Dieter van Melkebeek, and Osamu Watanabe

Session 1B

Parallel Approximation of Min-max Problems with Applications to Classical and Quantum Zero-Sum Games .................................................................................................................................21
Gus Gutoski and Xiaodi Wu

The Complexity of the Separable Hamiltonian Problem .....................................................................................32
André Chailloux and Or Sattath

Quantum Money with Classical Verification ........................................................................................................42
Dmitry Gavinsky

Session 2A

On the Usefulness of Predicates ...............................................................................................................................53
Per Austrin and Johan Håstad

Reductions between Expansion Problems ..............................................................................................................64
Prasad Raghavendra, David Steurer, and Madhur Tulsiani
On Problems as Hard as CNF-SAT ............................................................................................................74
  Marek Cygan, Holger Dell, Daniel Lokshtanov, Dániel Marx, Jesper Nederlof,
  Yoshio Okamoto, Ramamohan Paturi, Saket Saurabh, and Magnus Wahlström

Session 2B
Testing List H-homomorphisms ..................................................................................................................85
  Yuichi Yoshida
A Dichotomy for Real Weighted Holant Problems ......................................................................................96
  Sangxia Huang and Pinyan Lu

Session 3A
A Satisfiability Algorithm and Average-Case Hardness for Formulas over
the Full Binary Basis .................................................................................................................................107
  Kazuhsa Seto and Suguru Tamaki
Approximating AC^0 by Small Height Decision Trees and a Deterministic
Algorithm for #AC^0SAT ...........................................................................................................................117
  Paul Beame, Russell Impagliazzo, and Srikanth Srinivasan
DNF Sparsification and a Faster Deterministic Counting Algorithm ..........................................................126
  Parikshit Gopalan, Raghu Meka, and Omer Reingold

Session 3B: Invited Talk I
Communication Complexity and Information Complexity: Foundations
and New Directions ...................................................................................................................................136
  Toniann Pitassi

Session 4A
Gaussian Noise Sensitivity and Fourier Tails ..............................................................................................137
  Guy Kindler and Ryan O'Donnell
Junto-Symmetric Functions, Hypergraph Isomorphism and Crunching ....................................................148
  Sourav Chakraborty, Eldar Fischer, David Gacia–Soriano, and Arie Matsliah
Complexity Lower Bounds through Balanced Graph Properties ..............................................................159
  Guy Moshkovitz

Session 4B
Limitations of Lower-Bound Methods for the Wire Complexity of Boolean
Operators .......................................................................................................................................................170
  Andrew Drucker
Limits on Alternation-Trading Proofs for Time-Space Lower Bounds ......................................................181
  Samuel R. Buss and Ryan Williams
The Hardness of Being Private .................................................................192
Anil Ada, Arkadev Chattopadhyay, Stephen Cook, Lila Fontes, Michal Koucký, and Toniann Pitassi

Session 5A
Matrix Isomorphism of Matrix Lie Algebras ............................................203
Joshua A. Grochow
On Sunflowers and Matrix Multiplication ..............................................214
Noga Alon, Amir Shpilka, and Christopher Umans
Algebras of Minimal Multiplicative Complexity ....................................224
Markus Bläser and Bekhan Chokaev

Session 5B: Invited Talk II
Prospects for Geometric Complexity Theory ..........................................235
Peter Bürgisser

Session 6A
A Strong Direct Product Theorem for Quantum Query Complexity ..........236
Troy Lee and Jérémie Roland
A Strong Parallel Repetition Theorem for Projection Games on Expanders 247
Ran Raz and Ricky Rosen
Share Conversion and Private Information Retrieval .............................258
Amos Beimel, Yuval Ishai, Eyal Kushilevitz, and Ilan Orlov

Session 6B
Nondeterministic Circuit Lower Bounds from Mildly De-randomizing
Arthur-Merlin Games ..............................................................................269
Baris Aydintioğlu and Dieter van Melkebeek
Hitting Set Generators for Sparse Polynomials over Any Finite Fields 280
Chi-Jen Lu
Pseudorandom Generators for Read-Once ACC^0 .................................287
Dmitry Gavinsky, Shachar Lovett, and Srikanth Srinivasan

Session 7A
Non-malleable Extractors with Short Seeds and Applications to Privacy
Amplification ..........................................................................................298
Gil Cohen, Ran Raz, and Gil Segev
Better Condensers and New Extractors from Parvaresh-Vardy Codes 309
Amnon Ta-Shma and Christopher Umans
List Decoding Barnes-Wall Lattices ................................................................. 316  
*Elena Grigorescu and Chris Peikert*

**Session 7B**

Space-Efficient Algorithms for Reachability in Surface-Embedded Graphs ........................................... 326  
*Derrick Stolee and N.V. Vinodchandran*

Space Complexity in Polynomial Calculus ......................................................................................... 334  
*Yuval Filmus, Massimo Lauria, Jakob Nordström, Neil Thapen, and Noga Ron-Zewi*

Combinatorial PCPs with Short Proofs ............................................................................................. 345  
*Or Meir*

**Author Index** ............................................................................................................................... 356
Preface

The papers in this volume were accepted for presentation at the Twenty-Seventh Annual IEEE Conference on Computational Complexity, held 26-29 June 2012 at Universidade do Porto, Porto, Portugal. The conference is sponsored by the IEEE Computer Society Technical Committee on Mathematical Foundations of Computing, in cooperation with the ACM Special Interest Group on Algorithms and Computation Theory (SIGACT), the European Association for Theoretical Computer Science (EATCS), and the Instituto de Telecomunicações at Universidade do Porto.

The call for papers sought original research papers in all areas of computational complexity theory. Of the 119 submissions the program committee selected 34 for presentation at the conference. In addition, the program committee invited Peter Bürgisser and Toniann Pitassi to give invited talks at the conference.

The program committee, the local organizing committee, and the conference committee would like to thank everyone involved in the conference, including all those who submitted papers for consideration; the reviewers for their scientific contributions; the invited speakers; Shai Halevi for the software to run the electronic program committee and his prompt and helpful answers to questions regarding the software; and Juan Guerrero and Andrea Thibault of the IEEE Computer Society for their assistance with the production of this volume.

On behalf of all committee members, welcome to the conference!

Venkatesan Guruswami, Carnegie Mellon University, USA

CCC 2012 Program Committee Chair
Organizing Committee

**Conference Chair**
Peter Bro Miltersen, *Aarhus University, Denmark*

**Program Committee Chair**
Venkatesan Guruswami, *Carnegie Mellon University, USA*

**Local Organizing Committee Chair**
Luís Antunes, *Faculdade de Ciências da Universidade do Porto (FCUP-U. Porto), SQIG-IT, Portugal*

**Local Organizing Committee**
André Teixeira, *Faculdade de Ciências da Universidade do Porto (FCUP-U. Porto), SQIG-IT, Portugal*
André Souto, *SQIG-IT, Portugal*
Bruno Bauwens, *Faculdade de Ciências da Universidade do Porto (FCUP-U. Porto), Portugal*

**Conference Committee**
Manindra Agrawal, *IIT Kanpur, India*
Johan Håstad, *Kungliga Tekniska högskolan (KTH), Sweden*
Valentine Kabanets, *Simon Fraser University, Canada*
Dieter van Melkebeek, *University of Wisconsin, USA*
Amir Shpilka, *Technion, Israel*
Ryan Williams, *Stanford University, USA*

**Publicity**
Richard Beigel, *Temple University, USA*
Program Committee

Eric Allender, Rutgers University, USA  
Boaz Barak, Microsoft Research New England, USA  
Irit Dinur, Weizmann Institute, Israel  
Nicola Galesi, University of Rome La Sapienza, Italy  
Venkatesan Guruswami, Carnegie Mellon University, USA  
Neeraj Kayal, Microsoft Research India, India  
Julia Kempe, University of Paris 7 and Tel Aviv University, Israel  
Hartmut Klauck, Nanyang Technological University, Singapore  
Rahul Santhanam, University of Edinburgh, United Kingdom  
Rocco Servedio, Columbia University, USA  
Amir Yehudayoff, Technion, Israel
Reviewers

Masayuki Abe
Pavan Aduri
Carme Alvarez
Andris Ambainis
Prabhakaran Ananth
Albert Atserias
Per Austrin
David Mix Barrington
Paul Beame
Niel de Beaudrap
Salman Beigi
Eli Ben-Sasson
Debjayoti Bera
Olaf Beyersdorff
Aditya Bhaskara
Arnab Bhattacharyya
Abhishek Bhowmick
Eric Blais
Andrej Bogdanov
Fernando Brandao
Mark Braverman
Anne Broadbent
Peter Buergisser
Jin-Yi Cai
Igor Carboni
Lorenzo Carlucci
David Cash
Arkadev Chattopadhyay
Hubie Chen
Xi Chen
Mahdi Cheraghchi
Eden Chlamtac
Hana Chockler
Mattias Christandl
Kai-Min Chung
Henry Cohn
Dana Dachman-Soled
Stefan Dantchev
Bhaskar DasGupta
Costis Daskalakis
Mark Davenport
Anindya De
Amit Deshpande
Ilias Diakonikolas
Itsykson Dmitry
Yevgeniy Dodis
Aris Tentes
Zeev Dvir
Martin Dyer
Uri Feige
Uriel Feige
Michael Fellows
Michael Forbes
Lance Fortnow
Luke Friedman
Ariel Gabizon
Peter Gacs
David Gamarnik
Dmitry Gavinsky
Konstantinos Georgiou
Leslie Ann Goldberg
Paul Goldberg
Elazar Goldenberg
Oded Goldreich
Elena Grigorescu
Roberto Grossi
Ankit Gupta
Dan Gutfreund
Mohammad Taghi Hajiaghayi
Kristoffer Arnsfelt Hansen
Elad Haramati
Aram Harrow
Parthasarathy Harsha
Avinatan Hassidim
Florian Hess
John Hitchcock
Thomas Holenstein
Pavel Hrubes
Sandy Irani
Gabor Ivanyos
Prateek Jain
Rahul Jain
Stanislaw Jarecki
Emil Jerabek
Mark Jerrum
Stasys Jukna
Yael Kalai
Yael Tauman Kalai
Daniel Kane
Nathan Keller
Gillat Kol
Michal Koucky
Raghav Kulkarni
Orna Kupferman
Homin Lee
Troy Lee
Huijia Lin
Satya Lokam
Daniel Lokshutanov
Vadim Lyubashevsky
Barnaby Martin
Daniel Marx
Arie Matsliah
Pierre McKenzie
Raghv Meka
Daniele Micciancio
Ashley Montanaro
Moritz Mueller
Maarten van den Nest
Harumichi Nishimura
Jakob Nordstrom
Rafael Pass
Chris Peikert
Christopher Peikert
Pavel Pudlak
Youming Qiao
Prasad Raghavendra
Anup Rao
Oded Regev
Ben Reichardt
Alon Rosen
Benjamin Rossman
Guy Rothblum
Atri Rudra
Ramprasad Saptharishi
Or Sattath
Michael Schapira
Renate Scheidler
Grant Schoenbeck
Norbert Schuch
Pranab Sen
Ronen Shaltiel
Asaf Shapira
Alexander Sherstov
Amir Shpilka
Gaurav Sinha
Andre Souto
Srikanth Srinivasan
David Steurer
Li-Yang Tan
Till Tantau
Justin Thaler
Johan Thapper
Jacobo Toran
Boaz Tsaban
Madhur Tulsiani
Sarvagya Upadhyay
Salil Vadhan
Paul Valiant
Vinodchandran Variyam
Ramarathnam Venkatesan
Michael Viderman
Thomas Vidick
Andrew Wan
Brent Waters
Omri Weinstein
Daniel Wichs
Avi Wigderson
Ryan Williams
Peter Winkler
Ronald de Wolf
Paul Wollan
Sergey Yekhanin
Jiapeng Zhang
Shengyu Zhang
Yuan Zhou
Awards

The program committee of the Twenty-Seventh IEEE Conference on Computational Complexity is delighted to present the 2012 Ronald V. Book Prize for the Best Student Paper to Andrew Drucker for his paper "Limitations of Lower-Bound Methods for the Wire Complexity of Boolean Operators"

The award is given annually to the most outstanding paper written solely by one or more students, and is named after theoretical computer scientist Ronald Vernon Book (1937-1997).