On the Complexity of Circuit Satisfiability (Extended Abstract)

Ramamohan Paturi [*]
University of California, San Diego
paturi@cs.ucsd.edu

Pavel Pudlák[†]
Institute of Mathematics
Academy of Sciences of the Czech Republic
pudlak@math.cas.cz

ABSTRACT

In this paper, we are concerned with the exponential complexity of the *Circuit Satisfiability* (**CktSat**) problem and more generally with the exponential complexity of **NP**-complete problems. Over the past 15 years or so, researchers have obtained a number of exponential-time algorithms with improved running times for exactly solving a variety of **NP**-complete problems. The improvements are typically in the form of better exponents compared to exhaustive search. Our goal is to develop techniques to prove specific lower bounds on the exponents under plausible complexity assumptions. We consider natural, though restricted, algorithmic paradigms and prove upper bounds on the success probability. Our approach has the advantage of clarifying the relative power of various algorithmic paradigms.

Our main technique is a a success probability amplification technique, called *the Exponential Amplification Lemma*, which shows that for any $f(n,m)$-size bounded probabilistic circuit family $\mathcal{A}$ that decides **CktSat** with success probability at least $2^{-\alpha n}$ for $\alpha < 1$ on inputs which are circuits of size m with n variables, there is another probabilistic circuit family $\mathcal{B}$ that decides **CktSat** with size roughly $f(\alpha n, f(n,m))$ and success probability about $2^{-\alpha^2 n} > 2^{-\alpha n}$. In contrast, the standard method for boosting success probability by repeated trials will improve it to $(1-(1-2^{-\alpha n})^t)$ ($\approx t2^{-\alpha n}$ for $t = O(2^{\alpha n})$) using circuits of size about $tf(n,m)$.

Using this lemma, we derive tight bounds on the exponent of the success probability for deciding the **CktSat** problem in a variety of probabilistic computational models under complexity assumptions. For example, we show that the success probability cannot be better than $2^{-n+o(n)}$ for deciding **CktSat** by probabilistic polynomial size circuits unless **CktSat** (thereby all of **NP**) for polynomial size instances can be decided by 2^{n^μ} size *deterministic* circuits for some $\mu < 1$, which is considered an unlikely event. As another example, we show that probabilistic quasilinear size circuits cannot achieve success probability better than $2^{-n+o(n)}$ unless **CktSat** (as well as **NP**) has $O(m^{O(\lg\lg m)})$ size deterministic circuits, which is very close to the statement $\mathbf{NP} \subseteq \mathbf{P/poly}$, an unlikely scenario.

Categories and Subject Descriptors

F.2.3 [**Theory of Computation**]: ANALYSIS OF ALGORITHMS AND PROBLEM COMPLEXITY—*Tradeoffs among Complexity Measures*

General Terms

Algorithms,Theory

Keywords

Circuit Satisfiability, **NP**-completeness

1. INTRODUCTION

It is well-known that all **NP**-complete problems are equivalent as far as polynomial-time solvability is concerned. However, much less is known about the exact complexity of these problems. If we assume $\mathbf{NP} \neq \mathbf{P}$ or other appropriate complexity statement, what can we say about the exact worst-case complexity of **NP**-complete problems? An important context for this question is the development of a series of exact exponential-time algorithms with improved run times for a number of problems including IndependentSet, k-SAT, and k-colorability. The series of improvements are typically in the form of better exponents compared to exhaustive search. However, exact worst-case complexity of these problems seem to differ considerably. These improvements prompt several complexity questions, chief among them is whether we can expect continued improvements in the exponent. Is there a limit beyond which one

[*]This research is supported by NSF grant CCF-0947262 from the Division of Computing and Communication Foundations.

[†]Partially supported by the Institute for Theoretical Computer Science, Prague (project 1M0545 of MŠMT ČR), grant IAA100190902 of GA AV ČR and Institutional Research Plan No. AV0Z10190503.

STOC'10, June 5–8, 2010, Cambridge, Massachusetts, USA.
Copyright 2010 ACM 978-1-4503-0050-6/10/06 ...$10.00.

should not expect improvement? How do these limits differ for different problems? Can we explain the differing limits in terms of the structural properties of the problems? What are the likely exact complexities of various **NP**-complete problems? Are the likely complexities of various problems related?

The current state of the art in complexity theory is far from being able to resolve these questions, especially the question of best exponents, even under reasonable complexity assumptions. We believe that it would be productive to approach these questions from the viewpoint of known algorithmic paradigms. Such an approach might be able to clarify the relative power of various algorithmic paradigms and might even be able to shed light on the best exponents in natural, though restricted, computational models under complexity assumptions. Furthermore, the study of the limitations of algorithmic paradigms might result in sharper versions of existing problems and suggest new directions of research. This paper presents an attempt in this direction and obtains nontrivial, interesting new results.

We propose to approach the randomized exact algorithms for **NP**-complete problems by studying the important subclass **OPP** of algorithms and its generalizations. **OPP** is the class of one-sided probabilistic polynomial-time algorithms. In contrast to the conventional complexity classes, **OPP** is a class of algorithms rather than a class of problems. This class captures a common design paradigm for randomized exact exponential-time algorithms: to repeat sufficiently many times a one-sided error probabilistic polynomial-time algorithm that is correct with an exponentially small probability so that the overall algorithm finds a witness with constant probability. **OPP** includes Davis-Putnam-style backtracking algorithms developed in recent times to provide improved exponential-time upper bounds [4, 3, 33, 15, 14, 16, 12, 19, 18] for a variety of **NP**-hard problems. While the original versions of some of these algorithms are couched as exponential-time algorithms, one can observe from a formalization by Eppstein [17] that these algorithms can be converted into probabilistic polynomial-time algorithms whose success probability is the reciprocal of the best exponential-time bound. **OPP** also includes local search algorithms such as Schöning's [39]. **OPP** is interesting not just because of ubiquity, but because such algorithms are ideal from the point of view of space efficiency, parallelization, and speed-up by quantum computation. What are the limitations of such algorithms for deciding **NP**-complete problems? Could the best algorithms for canonical **NP**-complete problems such as **CktSat** (the problem of deciding whether a circuit is satisfiable) or k-SAT be in **OPP**?

In addition to **OPP**, there are several other important algorithmic paradigms for designing exact algorithms for **NP**-hard problems, for example, exponential-time divide-and-conquer [27, 37], inclusion-exclusion/Möbius inversion [25, 8, 6, 7], dynamic programming [20, 36], group algebra [26, 44], sieve algorithms [2, 31], and Voronoi cell computations [30]. However, we argue that **OPP** and its generalizations could serve as an excellent starting point for the study of exponential-time algorithms for **NP**-complete problems in general.

Consider the problem of k-colorability for $k \geq 3$. The best-known algorithm [8] for this problem applies the inclusion-exclusion principle to achieve an $\tilde{O}(2^n)$ algorithm where n in the number of vertices of the graph. This algorithm and the prior best-known algorithms [37, 16, 11, 9] do not belong to the class **OPP**. This raises a natural question whether we can expect an **OPP** algorithm for k-colorability whose success probability is at least 2^{-n}. Beyond this, can we expect **OPP**-style optimal algorithms for k-colorability? Does there exist any **OPP** algorithm for k-colorability whose success probability is at least c^{-n} where c is independent of k? Negative answers (or evidence to that effect) for these questions would provide convincing proof (or evidence) that exponential-time inclusion-exclusion and dynamic programming paradigms are strictly more powerful than that of **OPP**. On the other hand, results that would show k-colorability can be solved by an **OPP** algorithm with c^{-n} success probability would be exciting. Similar situation exists with respect to the Hamiltonian path problem where we know of no **OPP** algorithms that succeed with significantly better than $1/n!$ success probability whereas it is well-known that there are $O(n^2 2^n)$ algorithms [5, 25] based on exponential-time dynamic programming and inclusion-exclusion techniques. It seems that resolving the question, whether **OPP**-style optimal algorithms exist for k-colorability and Hamiltonian path, is related to a fundamental issue regarding the tradeoff between time and success probability which in turn might hold one of the keys (the others being **P** versus **NP** and derandomization) for the complexity theory of exact algorithms for **NP**-complete problems.

To study the exponents in the exact complexity of **NP**-complete problems, it is useful to parameterize **NP** problem instances with two parameters. Usually, **NP** problem instances are parameterized by the size of the input. However, for the purpose of capturing the exponential complexity, it is more natural to parameterize the instances in terms of an additional *complexity* parameter. For example, the **CktSat** problem instances are parameterized by n, the number of variables, and m, the length of the input, which is a description of a circuit. A graph problem such as k-colorability is parameterized by n, the number of vertices, and m, the length of the representation of the graph. These parameters are natural and robust with respect to the representation of the input. We assume that problems are presented together with a complexity parameter. It is also useful to endow the class **NP** with a canonical complexity parameter so we can state properties of **NP** in terms of the complexity parameter and the size of the input. Consider an **NP** predicate in the form $\exists x \Phi(y, x)$ where y is the input instance, x the witness, and Φ is a polynomial-time (in the length of y) computable predicate. We canonically parameterize **NP** problem instances by n, the length of the witness, as well as m, the length of the input. Focus on the length of the witness is natural when we consider exponential-time exact algorithms.

We use the notation $\mathbf{NP}(n,m)$ and $\mathbf{CktSat}(n,m)$ to be explicit about the parameterization. Complexity bounds are usually expressed as functions of both n and m. We would like to observe that any problem instance in $\mathbf{NP}(n,m)$ can be reduced to a $\mathbf{CktSat}(n,m')$ instance in time t preserving the witness length, where t and m' are polynomially bounded in m.

The results in the paper concern the more general class $\mathbf{OP}(T(n,m))$ of one-sided probabilistic algorithms that run in time $T(n,m)$ with success probability bounded by a small exponential function in n, independent of m. In addition to the case where $T(n,m)$ is polynomially bounded in m, we consider other natural cases. In fact, several of the branch-and-bound algorithms mentioned earlier run in quasilinear time where the success probability is inverse of the exponential run time. We will also consider more powerful models where the algorithm can check exponentially many candidates to find a witness. In particular, we consider the class $\mathbf{OP}(T(n,m))$ of algorithms for the following cases of $T(n,m)$:

- $T(n,m)$ is polynomially bounded in m,
- $T(n,m)$ is quasilinearly bounded in m (of the form $O(m\lg^k m)$ for some $k \geq 0$),
- $T(n,m)$ is subexponential in n and quasilinear in m, and
- $T(n,m)$ is a small exponential in n and quasilinear in m.

Our results include upper bounds on the success probability for deciding the **CktSat** problem in all of the models mentioned above subject to various complexity assumptions. These bounds are tight in the sense they achieve the best possible constant in the exponent. In particular, we show that the **CktSat** problem cannot be decided with success probability better than $2^{-n+o(n)}$ by **OPP** algorithms unless there are $m2^{O(n^{\mu}\lg^{1-\mu} m)}$ size *deterministic* circuits with $\mu < 1$ for deciding $\mathbf{CktSat}(n,m)$ (Theorem 4). The latter condition implies that $\mathbf{CktSat}(n,m)$ can be decided by deterministic circuits of size $2^{n^{\mu'}}$ for some $\mu' < 1$ if m is polynomially bounded in n, which is considered an unlikely event.

We will also prove similar upper bounds on the success probability for quasilinear probabilistic circuits. However, in this case, a much weaker assumption suffices. In particular, we show that the success probability cannot be better than $2^{-n+o(n)}$ in quasilinear probabilistic models unless $\mathbf{CktSat}(n,m)$ has $O(\mathrm{poly}(m)n^{O(\lg\lg m)})$ deterministic size circuits. The latter statement is very close to the statement $\mathbf{NP} \in \mathbf{P/poly}$, which is an unlikely event.

We will further show that the success probability in the subexponential model cannot be better than $2^{-n+o(n)}$ unless $\mathbf{CktSat}(n,m)$ has $2^{o(n)}\mathrm{poly}(m)$ deterministic circuits (Theorem 6). In particular, the latter condition violates **ETH**.

We will also show an optimal upper bound on the success probability for small exponential time models of the form $2^{\alpha n} m \lg^k m$ for $\alpha < 1$. We show that the success probability in this model cannot be better than $2^{-(1-\alpha-\varepsilon)n+o(n)}$ for any $\varepsilon > 0$ unless $\mathbf{CktSat}(n,m)$ has $2^{\beta n}\mathrm{poly}(m)$ size deterministic circuits where $\beta = 1/(1+\varepsilon/\alpha) < 1$. Although the latter statement is weaker, it still would be surprising if there is a constant factor reduction in the number of existential quantifiers for all of **NP**.

Our results essentially rule out any but exhaustive search algorithms for **CktSat** in the **OPP** and other models. We believe that our results are the first of their kind. We hope that these results would put a new emphasis on obtaining similar results for combinatorial problems such as k-colorability and Hamiltonian path as well as on the general problem of time-success probability tradeoffs.

In Section 1.1, we present some related prior work. In Section 1.2, we describe the key ideas underlying our results. We define basic concepts and complexity measures in Section 2. In Section 3, we present informal descriptions of the lemmas (Exponential Amplification Lemma and the hash-down lemma) followed by their proofs. Finally, in Section 3.3 we present our conditional upperbounds on the success probability for deciding **CktSat**.

1.1 Related Prior Work

While there is a large amount of literature on the topic of the satisfiability problem, we will focus in this section on previous research dealing with lower bounds on the exponential complexity of **CktSat** and related **NP**-complete problems.

Stearns and Hunt in their 1990 paper [40] considered the concept of *power index* to characterize the exponential complexities of basic **NP**-complete problems. Power index is defined to be the infimum of all θ for which the problem is in $\mathbf{DTIME}(2^{m^{\theta}})$ where m is the length of the input. They have hypothesized that the power index of the CNF Satisfiability problem is 1 (*Satisfiability Hypothesis*) and using this assumption they have shown that the power index of the Clique problem is $\frac{1}{2}$ (in terms of the number of edges in the graph). They get their results mainly by analyzing how reductions change input lengths. The less a reduction blows up the input size, the tighter the connection between the power indices of the problems.

While these results are interesting, they are more sensitive to how the input is represented. We feel that it is more natural to parameterize **NP** in terms of witness size as well as input size, since the obvious exhaustive search algorithm is strongly exponential in the witness size, but may only be weakly exponential (as in the case of the Clique problem) in the input size. More importantly, while [40] assumed the Satisfiability Hypothesis to obtain lower bounds on the power indices for other **NP**-complete problems, in this paper we provide a justification for their assumption by showing that an analogous power index (for the success probability) for a closely related problem, **CktSat**, is 1 in **OPP**-style models under complexity assumptions that are intimately related to the **P** versus **NP** question. Moreover, success probability is parameterized using the more natural and robust parameter, the witness length.

In an earlier paper, Schnorr [38] considered the problems in the classes **NQL**, the nondeterministic quasilinear time and **QL**, the deterministic quasilinear time, under quasilinear time reductions. He showed that the CNF Satisfiability problem (and thereby the **CktSat** problem) are complete for **NQL** under quasilinear time reductions. In fact, Stearns and Hunt cite this result to provide an indirect justification for the Satisfiability Hypothesis since the power index of the CNF Satisfiability problems is at least as large as the power index of any language in **NQL**. Schnorr in this paper comments that $\mathbf{P} \neq \mathbf{NP}$ implies $\mathbf{QL} \neq \mathbf{NQL}$ but the converse is not clear. Our Theorem 5 provides certain strengthening of the forward implication: the probabilistic version of **QL** cannot achieve better than $2^{-n+o(n)}$ success probability unless **NP** has almost polynomial size circuits.

More recently, Impagliazzo, Paturi and Zane [22] explored the question whether we can expect continued improvements in terms of better exponents for the exact complexity of problems such as k-SAT, k-colorability and IndependentSet and showed that the possibility of arbitrarily small exponents for various **NP**-complete problems is one and the same. In particular, they defined the notion of subexponential time reduction families (SERF) and showed that several search problems including k-SAT, IndependentSet, k-Set Cover, Clique, Vertex Cover and k-colorability are SERF-equivalent. They also showed that some of these problems such as k-SAT and k-colorability are SERF-complete for the class of **SNP** of search problems expressible by second order existential formulas whose first order part is universal. If any of these problems can be solved subexponentially (in terms of witness length), then every problem in **SNP** can be solved in subexponential time (in terms of witness length). The key to the equivalence is a lemma called the *Sparsification Lemma* which shows one can achieve witness size-preserving reductions among these problems in subexponential time.

In a subsequent paper [21], Impagliazzo and Paturi considered the exact complexity s_k of k-SAT where $s_k = \inf\{\varepsilon | \exists \text{ a } 2^{\varepsilon n} \text{ randomized algorithm for deciding } k\text{-SAT}\}$. Under the assumption $s_3 > 0$, called the *Exponential Time Hypothesis* (**ETH**), they showed that the sequence is increasing infinitely often as k increases. Even under **ETH**, it is an open question to prove a specific lower bound for s_3, that is, to prove $s_3 > c$ for some specific $c > 0$. It is also open whether $s_\infty = 1$ where $s_\infty = \lim_{k\to\infty} s_k$. More generally, it is an open question to prove optimal exponential lower bounds for any **NP**-complete problem under plausible complexity assumptions.

Adopting **ETH** as an axiom casts light on the complexity of many other problems. Marx [28, 29] used Sparsification Lemma to show that **ETH** implies that the complexity of database queries is determined by their treewidth. Very recently, Traxler [41] has shown that **ETH** implies $(k, 2)$-CSP has k^{cn} complexity where c is an absolute constant, thus ruling out the possibility (under **ETH**) of a c^n time algorithm where c is independent of k. In contrast, k-coloring, a very important case of $(k, 2)$-CSP, has long been known to have such a c^n algorithm for c independent of k, and very recently, the c has been improved to 2 [8].

There are a number of results that relate the complexity of **CktSat** in terms of the tractability of parameterized problems. Abrahamson, Downey and Fellows have shown that the existence of $2^{o(n)}\text{poly}(m)$ algorithms for **CktSat** problem for circuits of size m and n variables is equivalent to the problem of the tractability of the class of fixed parameter problems [1]. Other interesting results regarding the connection between the possibilities of somewhat improved algorithms for parameterized problems and subexponential time algorithms of the form $2^{o(n)}\text{poly}(m)$ for **CktSat** can be found in [13].

While the work of [22] and the subsequent results based on **ETH** as well as the results connecting the complexity of **CktSat** with fixed parameter tractability are interesting and represent progress, we still do not have specific lower bounds on the exponents even under **ETH** as the constants in results based on **ETH** (such as those in [41]) depend on the assumed constant s_3 in **ETH**. In contrast, our results do obtain a specific lower bound on the exponent of the success probability for **OPP** and other models under reasonable complexity assumptions. Interestingly, our work suggests the possibility that the question of specific lower bounds on the exponents is related to the questions of **P** versus **NP** and time-success probability tradeoff for **NP**-complete problems.

We would also like to mention the recent time-space lower bounds (see [43] for a survey): the Formula Satisfiability problem cannot be decided by a deterministic random-access machine that runs in time $m^{1.801}$ and space $m^{o(1)}$ where m is the input length. Unlike our results, these results do not depend on any complexity assumptions. On the other hand, our results deal with exponential time/probability and related algorithmic paradigms.

1.2 Key Ideas

Our key idea is a reduction which effects a simultaneous tradeoff between computational resources and problem instance parameters. Several basic conditions are needed to obtain such a tradeoff. One of them is to be able to parameterize the problem instances by two parameters, witness size and input size. Another is to be able to parameterize the computational models with two resources or complexity parameters (as functions of the instance parameters). For example, size and success probability are the computational resources for probabilistic circuit models. We then need a *non black-box* reduction technique where the computation itself (after hashing down) is the reduced instance. Identity of the space of instances and the space of computations seems to be crucially necessary for the reduction technique. The last condition distinguishes the **CktSat** problem from other satisfiability problems such as **CNFSAT**. Consider a probabilistic computation that checks the satisfiability of a CNF F. To represent the computation as a CNF F', either F' would have to be of exponentially large size or it would require an increase in the number of variables. In either case, it is not clear how to achieve the desired tradeoff between the complexity parameters and instance parameters when CNFs are the instances.

These basic ideas are brought together to obtain a simultaneous tradeoff between computational resources and between instance parameters to prove our key lemma, *the Exponential Amplification Lemma* (Lemma 2). The Exponential Amplification Lemma is a success probability amplification technique which shows that for any $f(n,m)$-size bounded probabilistic circuit family $\mathcal{A}$ that decides **CktSat** with success probability at least $2^{-\alpha n}$ on inputs which are circuits of size m with n variables, there is another probabilistic circuit family $\mathcal{B}$ that decides **CktSat** with size roughly $f(\alpha n, f(n,m))$ and success probability about $2^{-\alpha^2 n}$ which is greater than $2^{-\alpha n}$ for $\alpha < 1$. In contrast, the standard method for boosting success probability by repeated trials will improve it to $(1-(1-2^{-\alpha n})^t)$ ($\approx t2^{-\alpha n}$ for $t = O(2^{\alpha n})$) using circuits of size about $tf(n,m)$. Consider the following example that will make clear the advantage of the Exponential Amplification Lemma over the standard probability boosting technique. Let $f(n,m) = m^k$ for some $k \geq 0$. If **CktSat** for polynomial size circuits can be solved with $2^{-\alpha n}$ success probability by a probabilistic circuit family of size m^k, then the lemma guarantees a probabilistic circuit family of size about m^{2k} which solves **CktSat** for polynomial size circuits with success probability of about $2^{-\alpha^2 n}$. On the other hand, the standard boosting technique improves the success probability to $m^k 2^{-\alpha n}$ using the same resources. From this example, it is clear that the lemma provides an exponential advantage for $\alpha < 1$.

Elements of our technique are present in [21] where instances of CNF satisfiability are parameterized by the number of variables and the maximum width of the clauses and a subexponential time reduction was used to trade up the width for reducing the number of variables thereby obtaining relationships among the exponents. Traxler [41] also obtains a similar tradeoff between the size of the domain and the number of variables for constraint satisfaction problems. Whereas the techniques in [21, 41] only involve reductions among instances trading one parameter for another, our current technique for **CktSat** obtains simultaneous tradeoffs between algorithmic resources and instance parameters by effecting them in terms of each other.

2. CIRCUITS, CIRCUIT SATISFIABILITY, AND CIRCUIT FAMILIES

Let $\mathcal{C}$ denote the class of circuits with a single output over the standard, bounded fan-in basis AND, OR, and NOT. Each source node in the directed acyclic graph of a circuit C is either labeled by a variable or by a constant. Let $n = n(C)$ denote the number of variables of C, Let $\mathbf{size}(C)$ denote the count of gates in the circuit where each source node is counted as a gate. Encoding of a circuit contains two parts: the first part encodes the number of variables in unary notation and the second part is a standard encoding of the circuit as a binary string. For a circuit C, let $\mathbf{desc}(C)$ denote the encoding of the circuit and let $m(C) = |\mathbf{desc}(C)|$. $m(C)$ is at least $n(C)$ and is $O(\mathbf{size}(C)\lg(\mathbf{size}(C)))$.

In this paper, we are primarily concerned with **CktSat**, the *Circuit Satisfiability* problem: given an encoding of $C \in \mathcal{C}$, does there exist a $x \in \{0,1\}^{n(C)}$, that is, a setting of the variables of C, such that C with setting x outputs 1. In such a case, we say that the circuit C is *satisfiable.*

A *probabilistic circuit* is a circuit where each variable is further distinguished as an input variable or a random variable. For input variables y and random variables z, $C(y,z)$ denotes the output of the probabilistic circuit C. $C^y(z)$ denotes the *specialization* of the circuit when its first argument is fixed at the value y. Let $\mathbf{Pr}[C^y(z) = 1]$ denote the probability that the circuit C outputs 1 for the input y, where the random variables z of C take the values 0 or 1 with equal probability.

Input instances for the **CktSat** problem are encodings of circuits and are parameterized by the number of variables and the length of the encoding. In our constructions, we also deal with circuits with multiple outputs. Earlier notions and notation extend naturally to such circuits.

We consider probabilistic circuit families indexed by instance parameters as computational models. A family $\mathcal{F}$ of circuits is a collection $\{F_{n,m} | n, m \geq 1\}$ where $F_{n,m}$ is a probabilistic circuit whose inputs are encodings of circuits with n variables and of encoding length m. For $f : \mathbb{N} \times \mathbb{N} \to \mathbb{R}$, we say that a circuit family $\{F_{n,m}\}$ is f-bounded if $\mathbf{size}(F_{n,m}) \leq O(f(n,m))$. We say that a circuit family $\{F_{n,m}\}$ *decides* **CktSat** with *success probability* $p(n)$ if for all inputs which are encodings of circuits with n variables and of encoding length m, $F_{n,m}$ outputs 1 with probability at least $p(n)$ for all satisfiable circuits and otherwise outputs 0 with probability 1. In other words, $p(n) = \inf_{m,y} \mathbf{Pr}[F^y_{n,m}(z) = 1]$, where y is a length m string which is an encoding of a satisfiable circuit with n variables and z denotes the string of random variables of $F_{n,m}$.

Let $\mathcal{F}$ be a circuit family $\{F_{n,m}\}$ deciding **CktSat** with success probability $p(n)$. We define the (exponential) *complexity* of $\mathcal{F}$ for deciding **CktSat** for inputs which are circuits with n variables as

$$E_{\mathbf{CktSat}}(\mathcal{F}, n) = \lg(1/p(n))/n.$$

The idea is that $E_{\mathbf{CktSat}}(\mathcal{F}, n)$ captures the exponent c of the success probability $p(n)$ when expressed as $(2^{-n})^c$. The larger the exponent, the higher the complexity of $\mathcal{F}$. The complexity of $\mathcal{F}$ is defined as $E_{\mathbf{CktSat}}(\mathcal{F}) = \limsup_{n\to\infty} E_{\mathbf{CktSat}}(\mathcal{F}, n)$.

We define the complexity $E_{\mathbf{CktSat}}(f)$ of deciding **CktSat** as the *best* exponent achievable by an f-bounded probabilistic circuit family. More precisely,

$$E_{\mathbf{CktSat}}(f) = \inf\{\varepsilon | \exists \text{ an } f\text{-bounded family } \mathcal{F} \text{ for deciding } \\ \mathbf{CktSat} \text{ such that} E_{\mathbf{CktSat}}(\mathcal{F}) \leq \varepsilon\}.$$

We are interested in f-bounded circuit families where $f(n,m) = O(2^{\alpha(n)n} m^k \lg^l m)$ where $\alpha(n)$ is $0, o_n(1)$, or constant, $k \geq 1$, and $l \geq 0$. Such circuit families support computing paradigms where one can evaluate $2^{\alpha(n)n}$ witnesses to find a satisfying solution since a circuit of encoding length m can be evaluated at a given input by a circuit of size $m \lg^l m$ for some $l \geq 0$ [34, 35]. We use the notation $\tilde{O}(f)$ to suppress polylogarithmic factors in f.

3. COMPLEXITY OF CIRCUIT SATISFIABILITY

We know that $E_{\mathbf{CktSat}}(\mathbf{2^{o(n)}\tilde{O}(m)}) \leq E_{\mathbf{CktSat}}(\mathbf{m^k})$ $\leq E_{\mathbf{CktSat}}(\mathbf{\tilde{O}(m)}) \leq 1$. We also know that $E_{\mathbf{CktSat}}(\mathbf{2^{\alpha n}\tilde{O}(m)}) \leq 1 - \alpha$. It is open whether any of these complexities can be lower bounded. In this paper, we prove $E_{\mathbf{CktSat}}(\mathbf{2^{o(n)}\tilde{O}(m)}) = E_{\mathbf{CktSat}}(\mathbf{m^k}) = E_{\mathbf{CktSat}}(\mathbf{\tilde{O}(m)}) = 1$ under complexity assumptions. We also prove that for any $\alpha < 1$ and $\varepsilon > 0$, $E_{\mathbf{CktSat}}(\mathbf{2^{\alpha n}\tilde{O}(m)}) \geq 1 - \alpha - \varepsilon$ under a certain complexity assumption.

Our results follow from a repeated application of the Exponential Amplification Lemma, which shows how to construct a probabilistic circuit family $\mathcal{G}$ for deciding **CktSat** with improved success probability $2^{-\alpha^2 n}$ from a given f-bounded circuit family $\mathcal{F}$ that decides **CktSat** with success probability $2^{-\alpha n}$. In the following, we provide an informal description of the proof of the lemma.

Let $F_{n,m}$ be a probabilistic circuit in the family $\mathcal{F}$ and let $F_{n,m}(y, z)$ denote the probabilistic computation of $F_{n,m}$ on an input y where z represents the random variables. Let y be the encoding of a circuit C with n variables and encoding length m. Since $F_{n,m}(y, z)$ decides the satisfiability of C with one-sided error, the specialization $F^y_{n,m}(z)$ and the circuit C are equivalent with respect to satisfiability. We will treat the computation $C' = F^y_{n,m}(z)$ as a **CktSat** problem instance. The size of C' is $m' = f(n, m)$ and it may have as many as $r = f(n, m)$ variables. The idea is to use the encoding of C' as input to the probabilistic circuit $F_{r,m'}$ from the family $\mathcal{F}$ to decide the satisfiability of C.

So far the transformation from C to C' does not represent any useful tradeoff between the instance parameters. To make progress, we need the following crucial observation. If C is satisfiable and if $\alpha < 1$, then the density of satisfying assignments in C' will be high, that is, at least $2^{-\alpha n}$. Using a technique of Valiant and Vazirani [42], we can hash down the circuit C' to obtain another circuit C'' with reduced number of variables, about αn, in such a way that C'' and C' are equivalent with respect to satisfiability and that the size of C'' is about the same as that of C' (up to polylogarithmic factors). Overall, we can constructively transform a circuit C with n variables and encoding length m to C'' with about αn variables and encoding length $m'' \approx f(n, m)$ (up to polylogarithmic factors). We will then input C'' to the probabilistic circuit $F_{\alpha n, m''}$ from the family $\mathcal{F}$ to achieve an improved success probability of about $2^{-\alpha^2 n}$. The probabilistic circuit family $\mathcal{G}$ is designed to implement this amplification in such a way that $\mathcal{G}$ is g-bounded where $g(n, m) \approx f(\alpha n, \tilde{O}(f(n, m))$ (up to polylogarithmic factors).

In the next two subsections we state and prove the hash-down lemma and the Exponential Amplification Lemma. In the subsequent section, we state our results on the exponential complexity of **CktSat**.

3.1 Hash-Down Lemma

As is clear from the previous description, an important ingredient in the proof of the Exponential Amplification Lemma is the construction of a circuit $\mathbf{Sparse}(C)$, which equivalent to the circuit C with respect to satisfiability, but with fewer variables. This construction is closely related to that of the unique satisfiability construction of Valiant and Vazirani [42] and is captured in the following hash-down lemma, Lemma 1.

Let $C \in \mathcal{C}$ be a satisfiable circuit with n variables and let m denote its description length. Let $S^C \subseteq \{0, 1\}^n$ be the nonempty set of satisfying assignments to the variables of C. Let $s = \lfloor \lg |S^C| \rfloor - 2$; we will assume that $s > 0$.

The intuition for the construction of $\mathbf{Sparse}(C)$ is as follows. If the set S^C is intersected with a random subcube of dimension $(n-s)$, we expect to get a nonempty intersection. Therefore, $(n - s)$ bits are sufficient to locate a satisfying assignment in the intersection. This intuition is operationalized by restricting C to inputs from a random coset of a random linear transformation from $\{0, 1\}^n$ to $\{0, 1\}^s$. Such a restriction turns out to be equivalent with respect to satisfiability since the random coset contains a satisfying assignment with sufficiently high probability. Moreover, members of the coset can be generated by a small-size circuit given their $(n-s)$-bit address in the coset. The details are provided below.

We follow the standard idea of using pairwise independent functions for hashing. However, rather than using random linear transformations we will use random Toeplitz matrices to achieve the desired pairwise independence. It requires only a linear number of bits to specify a random Toeplitz matrix over GF(2). Linear randomness together with fast GCD and convolution algorithms [10, 32] will only require a quasilinear computation to select a random coset and to address its members. While we can live with polynomial overhead for some of our theorems, quasilinear hashing is necessary when we work with quasilinear size circuit models.

Let n be fixed. Let $t = (t_{-n+1}, \cdots, t_{-1}, t_0, t_1, \cdots, t_{n-1}) \in \{0, 1\}^{2n-1}$. We will denote by T_t the Toeplitz matrix determined by t, which is the matrix defined by $T_t(i, j) = t_{(j-i)}$ for $0 \leq i, j \leq n - 1$. For a column vector $z \in \{0, 1\}^n$, let $(T_t z)_s$ denote the column vector of the first s bits of $T_t z$, the product of the matrix T_t with the column vector z. For $t \in \{0, 1\}^{2n-1}$ and $w \in \{0, 1\}^s$, define the affine linear transformation $h_{t,w}(z) = (T_t z)_s + w$. It is well-known that $\{h_{t,w}\}$ is a pairwise independent family of functions from $\{0, 1\}^n$ to $\{0, 1\}^s$.

We would like to parameterize the cosets $H_{t,w} = \{z | h_{t,w}(z) = 0\}$. For $w \in \{0, 1\}^s$ and $x \in \{0, 1\}^{n-s}$ we will define the column vector $(w; x)$ to be the concatenation of w and x. For $w \in \{0, 1\}^s$ and t such that T_t is invertible, we define $J_{t,w}(x) := T_t^{-1}(w; x)$ to obtain the needed parameterization of the coset $H_{t,w}$. When T_t is nonsingular, it is easy to see that $\mathbf{Image}(J_{t,w}) = H_{t,w}$. Indeed, if $z \in H_{t,w}$, then by definition $h_{t,w}(z) =$

$(T_t z)_s + w = 0$. This implies that there exists an $x \in \{0,1\}^{n-s}$ such that $T_t z + (w;x) = 0$. Since we are working over GF(2) and since T_t is nonsingular, it follows that $z = T_t^{-1}(w;x) = J_{t,w}(x)$. Similarly, if $z \in \mathbf{Image}(J_{t,w})$, then $z \in H_{t,w}$.

The circuit $\mathbf{Sparse}^{t,w}(C)$ with $(n-s)$ variables is the composition of $J_{t,w}$ with C, *i.e.*, $\mathbf{Sparse}^{t,w}(C)(x) = C(J_{t,w}(x))$.

Lemma 1. *C and $\mathbf{Sparse}^{t,w}(C)$ are equivalent with respect to satisfiability with probability at least $1/4$. Furthermore, $\mathbf{Sparse}^{t,w}(C)$ can be constructed in size $\tilde{O}(m)$ and its description can be computed by a circuit of size $\tilde{O}(m)$ given $\mathbf{desc}(C), t$ and w.*

Proof. Our goal is to prove

$$\mathbf{Pr}_{t,w}[S^C \cap \mathbf{Image}(J_{t,w}) \neq \varnothing] \geq 1/4,$$

where we are assuming the uniform distribution on $t \in \{0,1\}^{2n-1}$ and $w \in \{0,1\}^s$. However, $\mathbf{Image}(J_{t,w}) = H_{t,w}$ if T_t is nonsingular. We will first argue that $\mathbf{Pr}_{t,w}[S^C \cap H_{t,w} \neq \varnothing] \geq 3/4$ assuming T_t is nonsingular. We then use the result of Kaltofen and Lobo [24] which states $\mathbf{Pr}[T_t \text{ is nonsingular}] \geq 1/2$ to conclude

$$\begin{aligned}&\mathbf{Pr}_{t,w}[S^C \cap \mathbf{Image}(J_{t,w}) \neq \varnothing]\\ &\geq \mathbf{Pr}_{t,w}[S^C \cap H_{t,w} \neq \varnothing] - \mathbf{Pr}[T_t \text{ is singular}] \geq 1/4\end{aligned}$$

For $z \in S^C$, let $q_{t,w}(z)$ denote the indicator function for the event $z \in H_{t,w}$, *i.e.*, $h_{t,w}(z) = 0$. Let $Q_{t,w} := \sum_{z \in S^C} q_{t,w}(z)$. We have $Q_{t,w} \neq 0$ iff $S^C \cap H_{t,w} \neq \varnothing$. We will upper bound the probability of the event $Q_{t,w} = 0$ using Chebyshev's inequality: $\mathbf{Pr}[Q_{t,w} = 0] \leq \frac{\mathbf{Var}[Q_{t,w}]}{(\mathbf{E}[Q_{t,w}])^2}$. Using the property of pairwise independence of the functions $h_{t,w}$, we get

$$\begin{aligned}\mathbf{Var}[Q_{t,w}] &= \mathbf{E}[Q_{t,w}^2] - \mathbf{E}[Q_{t,w}]^2\\ &= (|S^C|^2 - |S^C|)2^{-2s} + |S^C|2^{-s} - |S^C|^2 2^{-2s}\\ &= |S^C|(2^{-s} - 2^{-2s}).\end{aligned}$$

Hence

$$\begin{aligned}\frac{\mathbf{Var}[Q_{t,w}]}{\mathbf{E}[Q_{t,w}]^2} &= \frac{|S^C|(2^{-s} - 2^{-2s})}{|S^C|^2 2^{-2s}}\\ &= \frac{1 - 2^{-s}}{|S^C|2^{-s}} \leq \frac{1 - 2^{-s}}{4} < 1/4.\end{aligned}$$

This completes the proof that C and $\mathbf{Sparse}^{t,w}(C)$ are equivalent with respect to satisfiability with probability at least 1/4..

Using fast GCD and convolution algorithms as well as the Gohberg-Semencul formula for the inverse of a Toeplitz matrix [10, 32], one can compute $J_{t,w}(x) = T_t^{-1}(w;x)$ with bit complexity $\tilde{O}(n)$. Since $m \geq n$, the circuit $\mathbf{Sparse}^{t,w}(C)$ can be constructed in $\tilde{O}(m)$ size. It is easy to check that the description of $\mathbf{Sparse}^{t,w}(C)$ can be computed by a circuit of size $\tilde{O}(m)$. □

It should be noted that there are linear size circuit constructions [23] for pairwise independent hash functions which would improve upon the construction in Lemma 1 in terms of circuit size by eliminating the polylogarithmic factor. However, we still need a logarithmic factor for describing the circuit and as a result the improved construction would not make a significant difference as far as our results are concerned.

3.2 Exponential Amplification Lemma

Lemma 2. **Exponential Amplification Lemma:** *Let $\mathcal{F}$ be an f-bounded family for some $f : \mathbb{N} \times \mathbb{N} \to \mathbb{R}$ such that $E_{\mathbf{CktSat}}(\mathcal{F}) < \delta$ for $0 < \delta < 1$. Assume $f(n,m) \geq m$ for all n. Then there exists a g-bounded circuit family $\mathcal{G}$ such that, for all sufficiently large n, $E_{\mathbf{CktSat}}(\mathcal{G}) < \delta^2$ where*

$$g(n,m) = O(f(\lceil \delta n \rceil + 5, \tilde{O}(f(n,m)))).$$

Proof. Let $F_{n,m}$ be the probabilistic circuit from the family $\mathcal{F}$ that decides **CktSat** for circuits of description size m and n variables. $F_{n,m}$ itself has m input variables which encode the description of a circuit D of n variables. Also $F_{n,m}$ has r random variables y. By assumption $\mathbf{size}(F_{n,m}) = O(f(n,m))$, $r = O(f(n,m))$, and the success probability $p(n)$ of $F_{n,m}$ is greater than $2^{-\delta n}$ for all sufficiently large n.

Let $n' = \lceil \delta n \rceil + 5$ and $s = r - n'$. It follows that $p(n) > 2^{-n'}$. We will construct a g-bounded family $\mathcal{G}$ of probabilistic circuits that decides **CktSat** with success probability $p(r-s) = p(n')$ for circuits with n variables where $g(n,m) = O(f(n', \tilde{O}(f(n,m))))$. A key idea is, for a given circuit D, to view $F_{n,m}(\mathbf{desc}(D), y)$ as a (deterministic) boolean circuit $C(y) := F_{n,m}^{\mathbf{desc}(D)}(y)$ of r variables. When D is satisfiable, it follows that $C(y)$ has at least $2^{(r - \lceil \delta n \rceil)}$ solutions for all sufficiently large n. We apply the **Sparse**() function to C to obtain a circuit with n' variables which is equivalent to D with respect to satisfiability. We then apply the description of $\mathbf{Sparse}(C)$ as input to an appropriate circuit from the family $\mathcal{F}$ to improve success probability. The details of this construction are presented in the following algorithm and figures.

Circuit $G_{n,m}$ (see Figure 1):

```
1  Input: desc(D) of $D \in \mathcal{C}$ with $n$ variables and description length $m$.
2  $C(y) := F_{n,m}^{\mathbf{desc}(D)}(y)$, the specialization of $F_{n,m}$ to the input desc(D).
3  If $r > n'$,
4        Select random $t \in \{0,1\}^{2r-1}$ and $w \in \{0,1\}^s$.
5        If $T_t$ is invertible, compute the description of a circuit $J_{t,w}(x)$ that computes $T_t^{-1}(w;x)$ for $x \in \{0,1\}^{n'}$.
6        $\mathbf{Sparse}^{t,w}(C)(x) := C(J_{t,w}(x))$.
7        $H(x) := \mathbf{Sparse}^{t,w}(C)(x)$ where $x \in \{0,1\}^{n'}$. (see Figure 2)
8  else
9        $H(x) = C(x)$ where $x \in \{0,1\}^r$.
10 PrepCkt: Compute the description of the circuit $H$. Let $m' = |\mathbf{desc}(H)|$.
11 Apply $F_{n'',m'}$ to desc(H) where $n'' = n'$ if $r > n'$, otherwise $n'' = r$.
12 Output: $F_{n'',m'}(\mathbf{desc}(H))$.
```

We will first argue that D and H are equivalent with respect to satisfiability. It is clear that if D is unsatisfiable, H is unsatisfiable. If D is satisfiable, then it follows from Lemma 1 that $\mathbf{Sparse}^{t,w}(C)$ is satisfiable with probability at least 1/4. Since $n'' \leq \lceil \delta n \rceil + 5$, it follows $F_{n'',m'}$ outputs 1 on input $\mathbf{desc}(H)$ with probability at least $2^{-\delta n''}$, which implies that the success probability of $G_{n,m}$ is at least $2^{-\delta' \delta n - 6\delta' - 5}$ for some $\delta' < \delta$. It then follows that $E_{\mathbf{CktSat}}(\mathcal{G}, n) < \delta^2$ for all sufficiently large n.

We will now upper bound the size of $G_{n,m}$. By Lemma 1, $\mathbf{Sparse}^{t,w}(C)(x)$ can be described using at most $m' = \tilde{O}(f(n,m))$ bits and the description itself can be computed by a circuit of size $\tilde{O}(f(n,m)))$. It follows that the size of H is also bounded by $\tilde{O}(f(n,m))$ whether $r > n'$ or not.

Thus, the size of $G_{n,m}$ is upper bounded by

$$f(n'',m')+\tilde{O}(f(n,m)) \leq f(\lceil \delta n \rceil+5,\tilde{O}(f(n,m)))+\tilde{O}(f(n,m))$$
$$\leq O(f(\lceil \delta n \rceil+5,\tilde{O}(f(n,m)))) \text{ since } f(n,m) \geq m.$$

□

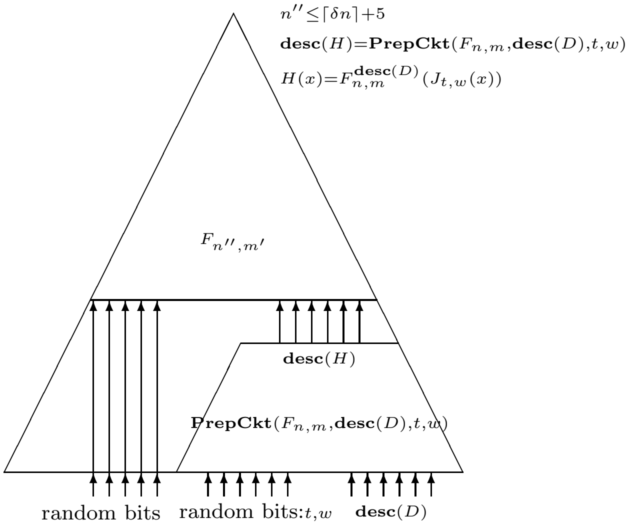
$n'' \leq \lceil \delta n \rceil + 5$
$\mathbf{desc}(H) = \mathbf{PrepCkt}(F_{n,m}, \mathbf{desc}(D), t, w)$
$H(x) = F_{n,m}^{\mathbf{desc}(D)}(J_{t,w}(x))$
$F_{n'',m'}$
$\mathbf{desc}(H)$
$\mathbf{PrepCkt}(F_{n,m}, \mathbf{desc}(D), t, w)$
random bits
random bits:t,w
$\mathbf{desc}(D)$

Figure 1: Circuit $G_{n,m}$

We present the following lemma which states how to boost the success probability by repeated independent trails. The proof follows from standard arguments.

Lemma 3. *If C is a probabilistic circuit of size M deciding* **CktSat** *with success probability $q > 0$ for a set of inputs, then there exists a deterministic circuit D of size $\tilde{O}(M^2)/q$ that decides* **CktSat** *on the same set of inputs. The same holds for any set in* **NP** *in place of* **CktSat**.

3.3 Results

In this section, we will show that for certain $f(n,m)$, any $f(n,m)$-bounded probabilistic circuit family deciding **CktSat** cannot succeed with probability better than $2^{-n+o(n)}$, based on the assumption that **CktSat** does not have small size deterministic circuits. The outline of our argument is as follows: we start with an $f(n,m)$-bounded probabilistic circuit family that decides **CktSat** with success probability $2^{-\delta n}$ for $\delta < 1$ and construct a deterministic circuit family for deciding **CktSat** by repeated application of the Exponential Amplification Lemma followed by an application of Lemma 3. The size of the resulting deterministic circuit family depends on $f(n,m)$. The smaller the f, the smaller the size of the deterministic circuit family.

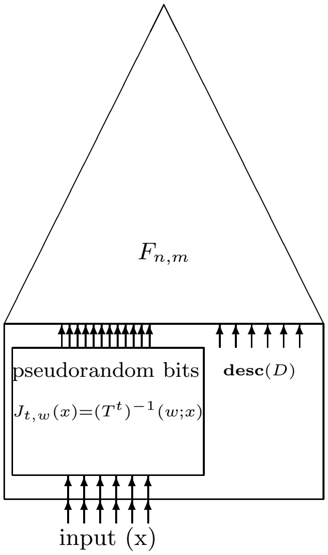

Figure 2: $H(x) = F_{n,m}^{\mathbf{desc}(D)}(J_{t,w}(x))$

In the following theorems we state the size of the resulting deterministic circuits for deciding $\mathbf{CktSat}(n,m)$. While our size bounds hold for any n and m, the implausibility of small size deterministic circuits for deciding $\mathbf{CktSat}(n,m)$ also depends on the relative size of m with respect to n. For example, the **CktSat** problem for circuits where the description length m is exponentially large in n (say, $m = 2^n$) is decidable by a *polynomially bounded* (say, $f(n,m) = \Omega(m^2 \lg m)$) deterministic circuit family since $\mathbf{CktSat}(n,m)$ can be decided by circuits of size $2^n\tilde{O}(m)$. In remarks following the theorems, we point out explicitly for what values of m (as a function of n) the existence of the resulting deterministic circuit family is considered implausible.

Theorem 4. *For $k > 1$, either $E_{\mathbf{CktSat}}(\mathbf{m^k}) = 1$ or there exists a $\mu < 1$ such that $\mathbf{CktSat}(n,m)$ can be decided by deterministic circuits of size $m2^{O(n^\mu \lg^{1-\mu} m)}$.*

Remarks: Consider the second clause in the disjunction of the statement of the theorem: there exists a $\mu < 1$ such that $\mathbf{CktSat}(n,m)$ can be decided by deterministic circuits of size $m2^{O(n^\mu \lg^{1-\mu} m)}$. This statement implies that $\mathbf{CktSat}(n,m)$ can be decided by $O(2^{n^{\mu'}})$ size deterministic circuits for some $\mu' < 1$ for the case where m is bounded by a polynomial in n. It is currently believed that such circuits are unlikely to exist. Even when $m = 2^{o(n)}$, the statement implies that **CktSat** can be decided by deterministic circuits of size $2^{o(n)}$, which contradicts **ETH**.

Outline of the Proof: We assume that there exists a family $\mathcal{F}$ of probabilistic circuits of size $O(m^k)$ for some $k > 1$ achieving success probability $2^{-\delta n}$ for some $\delta < 1$. We apply the Exponential Amplification Lemma $d = \max(1, \lfloor (\lg \frac{n}{2\lg m + O(\lg\lg m)})/(\lg(\frac{1}{\delta} + \lg k) \rfloor)$ times followed by an application of Lemma 3 to obtain a deterministic circuit family of size $m2^{O(n^{\mu}\lg^{1-\mu} m)}$ where $\mu = \frac{\lg k}{\lg(k/\delta)} < 1$ and the constant in the big O notation depends on k and δ.

Theorem 5. *Either $E_{\mathbf{CktSat}}(\mathbf{\tilde{O}(m)}) = 1$ or* **CktSat**(n,m) *can be decided by deterministic circuits of size $O(\text{poly}(m)n^{O(\lg\lg m)})$.*

Remarks: The statement **CktSat**(n,m) can be decided by quasi-polynomial size deterministic circuits is very close to the statement $\mathbf{NP} \subseteq \mathbf{P/poly}$, which is considered unlikely. Theorem 5 is proved by applying the Exponential Amplification Lemma about $O(\lg n)$ times.

We will also get tight upper bounds on the success probability when $f(n,m)$ is quasilinear in m and subexponential or exponential in n.

Theorem 6. *Either $E_{\mathbf{CktSat}}(\mathbf{2^{o(n)}\tilde{O}(m)}) = 1$ or* **CktSat**(n,m) *can be decided by deterministic circuits of size $2^{o(n)}\text{poly}(m)$.*

Remarks: Theorem 6 is proved by applying the Exponential Amplification Lemma $\omega_n(1)$ times. When we restrict m to be polynomial in n, the second clause in the statement of the theorem implies that **CktSat**(n,m) has subexponential size circuits, which contradicts **ETH**.

Theorem 7. *For every $\alpha, \varepsilon > 0$, either $E_{\mathbf{CktSat}}(\mathbf{2^{\alpha n}\tilde{O}(m)}) \geq 1 - \alpha - \varepsilon$ or* **CktSat**(n,m) *can be decided by deterministic circuits of size $2^{n/(1+\varepsilon/\alpha)}\text{poly}(m)$.*

Remarks: In other words, if the success probability is better than $2^{-(1-\alpha)n+o(n)}$, the theorem implies that **CktSat**(n,m) can be decided by deterministic circuits of size $2^{cn}\text{poly}(m)$ where $c = 1/(1+\frac{\varepsilon}{\alpha}) < 1$. We note that the standard success probability boosting technique (as opposed to the Exponential Amplification Lemma) would give deterministic circuits of size $2^{(1-\varepsilon)n}\text{poly}(m)$. It is easy to see that $c < (1-\varepsilon)$ as long as $\alpha < 1$.

Acknowledgments: The authors would like to thank Boaz Barak, Chris Calabro, Russell Impagliazzo, Mike Saks, Salil Vadhan, Avi Wigderson, and Ryan Williams for helpful discussions. We would also like to thank the reviewers for valuable comments.

4. REFERENCES

[1] Karl R. Abrahamson, Rodney G. Downey, and Michael R. Fellows. Fixed-parameter tractability and completeness iv: On completeness for w[p] and pspace analogues. *Ann. Pure Appl. Logic*, 73(3):235–276, 1995.

[2] Miklós Ajtai, Ravi Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. In *STOC '01*, 601–610, 2001.

[3] Richard Beigel. Finding maximum independent sets in sparse and general graphs. In *SODA '99*, 856–857, 1999.

[4] Richard Beigel and David Eppstein. 3-coloring in time $o(1.3446^n)$: A no-mis algorithm. In *FOCS '95*, 444–452, 1995.

[5] Richard Bellman. Dynamic programming treatment of the traveling salesman problem. *Journal of the ACM*, 9(1):61–63, 1962.

[6] Andreas Björklund, Thore Husfeldt, Petteri Kaski, and Mikko Koivisto. Fourier meets möbius: Fast subset convolution. In *STOC '07*, 67–74, 2007.

[7] Andreas Björklund, Thore Husfeldt, Petteri Kaski, and Mikko Koivisto. Computing the tutte polynomial in vertex-exponential time. In *FOCS '08*, 677–686, 2008.

[8] Andreas Björklund, Thore Husfeldt, and Mikko Koivisto. Set partitioning via inclusionŰexclusion. *SIAM Journal of Computing*, 39(2):546–563, 2009.

[9] Hans Bodlaender and Dieter Kratsch. An exact algorithm for graph coloring with polynomial memory. Technical Report UU-CS-2006-015, Utrecht University, 2006.

[10] Richard P. Brent, Fred G. Gustavson, and David Y. Y. Yun. Fast solution of toeplitz systems of equations and computation of padé approximants. *J. Algorithms*, 1(3):259–295, 1980.

[11] J. M. Byskov. *Exact Algorithms for Graph Colouring and Exact Satisfiability.* PhD thesis, Aarhus University, August 2005.

[12] Jesper Makholm Byskov. Algorithms for k-colouring and finding maximal independent sets. In *SODA '03*, 456–457, 2003.

[13] Jianer Chen, Xiuzhen Huang, Iyad A. Kanj, and Ge Xia. Strong computational lower bounds via parameterized complexity. *J. Comput. Syst. Sci.*, 72(8):1346–1367, 2006.

[14] Evgeny Dantsin, Andreas Goerdt, Edward A. Hirsch, Ravi Kannan, Jon Kleinberg, Christos Papadimitriou, Prabhakar Raghavan, and Uwe Schöning. A deterministic $(2-2/(k+1))^n$ algorithm for k-sat based on local search. *Theoretical Computer Science*, 289(1):69–83, 2002.

[15] David Eppstein. Improved algorithms for 3-coloring, 3-edge-coloring, and constraint satisfaction. In *SODA '01*, 329–337, 2001.

[16] David Eppstein. Small maximal independent sets and faster exact graph coloring. *Journal of Graph Algorithms and Applications*, 7:131–140, 2003.

[17] David Eppstein. Quasiconvex analysis of multivariate recurrence equations for backtracking algorithms. *ACM Trans. Algorithms*, 2(4):492–509, 2006.

[18] F. Fomin, F. Grandoni, and D. Kratsch. Measure and conquer : a simple $o(2^{0.288n})$ independent set algorithm. In *SODA '06*, 18–25, 2006.

[19] Jens Gramm, Edward A. Hirsch, Rolf

Niedermeier, and Peter Rossmanith. Worst-case upper bounds for max-2-sat with an application to max-cut. *Discrete Applied Mathematics*, 130(2):139–155, 2003.

[20] Michael Held and Richard M. Karp. A dynamic programming approach to sequencing problems. In *Proceedings of the 1961 16th ACM national meeting*, 71.201–71.204, 1961.

[21] R. Impagliazzo and R. Paturi. The complexity of k-sat. *Journal of Computer and Systems Sciences*, 62(2):367–375, March 2001.

[22] R. Impagliazzo, R. Paturi, and F. Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63:512–530, 1998.

[23] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Cryptography with constant computational overhead. In *STOC '08*, 433–442, 2008.

[24] E. Kaltofen and A. Lobo. On rank properties of toeplitz matrices over finite fields. In *ISSAC '96*, 241–249, 1996.

[25] Richard M. Karp. Dynamic programming meets the principle of inclusion and exclusion. *Operations Research Letters*, 1:49–51, April 1982.

[26] Ioannis Koutis. Faster algebraic algorithms for path and packing problems. In *ICALP '08*, 575–586, 2008.

[27] E. Lawler. A note on the complexity of the chromatic number problem. *Information Processing Letters*, 5(3):66–67, 1976.

[28] Dániel Marx. Can you beat treewidth? In *FOCS '07*, 169–179, 2007.

[29] Dániel Marx. On the optimality of planar and geometric approximation schemes. In *FOCS '07*, 338–348, 2007.

[30] Daniele Micciancio and Panagiotis Voulgaris. A deterministic single exponential time algorithm for most lattice problems based on voronoi cell computations. In *STOC '10*, 2010. To appear.

[31] Daniele Micciancio and Panagiotis Voulgaris. Faster exponential time algorithms for the shortest vector problem. In *SODA '10*, 1468–1480, 2010.

[32] Victor Pan. *Structured Matrices and Polynomials: Unified Superfast Algorithms*. Birkhäuser, 2001.

[33] R. Paturi, P. Pudlák, and F. Zane. Satisfiability coding lemma. *Chicago Journal of Theoretical Computer Science*, December 1999.

[34] Nicholas Pippenger. Fast simulation of combinational logic circuits without random-access storage. In *Fifteenth Allerton Conference* , 25–33, 1977.

[35] Nicholas Pippenger and Michael J. Fischer. Relations among complexity measures. *J. ACM*, 26(2):361–381, 1979.

[36] J. M. Robson. Algorithms for maximum independent sets. *Journal of Algorithms*, 7:425–440, September 1986.

[37] Michael E. Saks. Space efficient algorithms for np-hard sequencing and partition problems. manuscript, November 1998.

[38] Claus-Peter Schnorr. Satisfiability is quasilinear complete in nql. *J. ACM*, 25(1):136–145, 1978.

[39] U. Schöning. A probabilistic algorithm for k-sat and constraint satisfaction problems. In *FOCS '99*, 410–414, 1999.

[40] Richard Edwin Stearns and Harry B. Hunt. Power indices and easier hard problems. *Mathematical Systems Theory*, 23(4):209–225, 1990.

[41] Patrick Traxler. The time complexity of constraint satisfaction. In *2008 Dagstuhl Workshop on Moderately Exponential-time Algorithms*, 2008.

[42] L. Valiant and V. Vazirani. NP is as easy as detecting unique solutions. *Theoretical Computer Science*, 47:85–93, 1986.

[43] Dieter van Melkebeek. A survey of lower bounds for satisfiability and related problems. *Electronic Colloquium on Computational Complexity (ECCC)*, 14(099), 2007.

[44] Ryan Williams. Finding paths of length k in $O^*(2^k)$ time. *Information Processing Letters*, 109(6):315–318, 2009.