

A Satisfiability Algorithm for Sparse Depth Two Threshold Circuits

Russell Impagliazzo
University of California, San Diego
 La Jolla, CA, USA
 Email: russell@cs.ucsd.edu

Ramamohan Paturi
University of California, San Diego
 La Jolla, CA, USA
 Email: paturi@cs.ucsd.edu

Stefan Schneider
University of California, San Diego
 La Jolla, CA, USA
 Email: stschnei@cs.ucsd.edu

Abstract—We give a nontrivial algorithm for the satisfiability problem for threshold circuits of depth two with a linear number of wires which improves over exhaustive search by an exponential factor. The independently interesting problem of the feasibility of sparse 0-1 integer linear programs is a special case. To our knowledge, our algorithm is the first to achieve constant savings even for the special case of Integer Linear Programming. The key idea is to reduce the satisfiability problem to the *Vector Domination* problem, the problem of checking whether there are two vectors in a given collection of vectors such that one dominates the other component-wise.

Our result generalizes to formulas of arbitrary constant depth. We also provide a satisfiability algorithm with constant savings for depth two circuits with symmetric gates where the total weighted fan-in is at most linear in the number of variables.

One of our motivations is proving strong lower bounds for TC0 circuits, exploiting the connection (established by Williams) between satisfiability algorithms and lower bounds. Our second motivation is to explore the connection between the expressive power of the circuits and the complexity of the corresponding circuit satisfiability problem.

Keywords—Threshold Circuits, Satisfiability Algorithms

I. INTRODUCTION

Satisfiability testing is both a canonical NP-complete problem [1], [2] and one of the most successful general approaches to solving real-world constraint satisfaction problems. In particular, optimized CNFSAT heuristics are used to address a variety of combinatorial search problems successfully in practice, such as circuit and protocol design verification. The exact complexity of the satisfiability problem is also central to complexity theory, as demonstrated by Williams [3], who has shown that any improvement (by even a superpolynomial factor compared to exhaustive search) for the satisfiability problem for general circuits implies circuit lower bounds. Furthermore he has successfully used the connection to prove superpolynomial size bounds for ACC⁰ circuits using a novel nontrivial satisfiability algorithm for ACC⁰ circuits, solving a long standing open problem [4].

This raises the questions: For which circuit models do nontrivial satisfiability algorithms exist? How does the amount of improvement over exhaustive search relate to the expressive power of the model (and hence to lower bounds)? Can satisfiability heuristics for stronger models than CNF be useful for real-world instances?

Both the connection to circuit lower bounds and to heuristic search algorithms point to threshold circuits as the model to study next. Bounded depth polynomial size threshold circuits TC⁰ are the next natural circuit class stronger than ACC⁰. TC⁰ is a powerful bounded depth computational model. It has been shown that basic operations like addition, multiplication, division, and sorting can be performed by bounded depth polynomial size threshold circuits [5], [6]. In contrast, unbounded fan-in bounded depth polynomial size circuits over the standard basis (even when supplemented with mod p gates for prime p) cannot compute the majority function [6]. However, our understanding of the limitations of bounded depth threshold circuits is extremely weak. Exponential lower bounds for such circuits are only known for the special case of depth two and bounded weight [7]. For larger depth circuits, barely superlinear lower bounds are known on the number of wires [8].

Satisfiability for depth two threshold circuits contains as special cases some well known problems of both theoretical and practical significance. CNFSAT is one such special case, since both conjunctions and disjunction are a special case of threshold gates. MAX- k -SAT, the optimization form of k -CNF satisfiability, is another special case, since the top threshold gate can count the number of satisfied clauses for an assignment. Even for MAX-3-SAT, no algorithms with a constant factor savings over exhaustive search are known (although such an algorithm is provided for MAX-2-SAT in [9]). Another special case is Integer Linear Programming (ILP), a problem that is very useful in expressing optimization problems both in theory and practice. Testing the feasibility for a 0-1 ILP is equivalent to testing the satisfiability of a circuit with two levels, the bottom consisting of threshold gates and the top level being a conjunction. So both theoretical and real-world motivation points us to trying to understand the satisfiability problem for depth two threshold circuits.

Santhanam [10] gives an algorithm with constant savings for linear size formulas of AND and OR gates with fan-in two. However, this does not directly give an algorithm for depth two threshold circuits, as converting a linear size threshold circuit into a formula over AND and OR gates gives quadratic size.

In all of these related problems, a key distinction is between the cases of *linear size* and *superlinear size* circuits.

In particular, an algorithm with constant savings for depth two threshold circuits of superlinear size would refute the *Strong Exponential Time Hypothesis* (**SETH**) [11], since k -CNF for all k can be reduced (via Sparsification Lemma [12]) to superlinear size depth two threshold circuits [13]. (**SETH** says that for every $\delta < 1$, there is a k such that k -SAT cannot be solved in time $O(2^{\delta n})$.) However, for CNFSAT and MAXSAT, algorithms with constant savings are known when the formula is *linear size* [14], [15], [16]. So, short of refuting **SETH**, the best we could hope for is to extend such an improvement to the linear size depth two threshold circuit satisfiability problem.

In this paper, we give the first improved algorithm, which obtains a constant savings in the exponent over exhaustive search for the satisfiability of cn -wire, depth two threshold circuits for every constant c . As a consequence, we also get a similar result for linear-size ILP. In follow-up work [17] we extend our algorithm for the special case of 0-1 ILP to a linear number of constraints (as opposed to a linear number of wires). The algorithm can also be extended to threshold formulas of constant depth. Under **SETH**, this is qualitatively the best we could hope for, but we expect that further work will improve our results quantitatively. For example, our savings is exponentially small in c , whereas in, e.g., the satisfiability algorithm of [18] for constant depth and-or circuits, it is polylogarithmic in c . We consider this just a first step towards a real understanding of the satisfiability problem for threshold circuits, and hope that future work will get improvements both in depth and in savings.

In the second part of this paper, we give a satisfiability algorithm for depth two circuits consisting of *symmetric gates*. While symmetric gates are more expressive than threshold gates, our algorithm requires that the weights are small integers. In particular, the algorithm requires the *weighted number of wires*, the sum of the absolute weights, to be linearly bounded. Note that a single symmetric gate with exponential weights can represent any Boolean function. It is therefore natural to consider symmetric gates with bounded weights.

While we do not obtain any new circuit lower bounds, there is some chance that this line of work could eventually yield such bounds. For example, if there is an algorithm for any constant depth threshold circuit with super-inverse-polynomial savings in c , then $\text{NEXP} \not\subseteq \text{TC}^0$ by applying [3].

Our main sub-routine is an algorithm for the *Vector Domination Problem*: given n vectors in \mathbb{R}^d , is there a pair of vectors so that the first is larger than the second in every coordinate? We show that, when $d < c \log n$ for a constant c , this problem can be solved in subquadratic time. In contrast, Williams [9] shows that solving even the Boolean special case of vector domination with a subquadratic algorithm when $d = \omega(\log n)$ would refute **SETH**. We think the

Vector Domination Problem may be of independent interest, and might be used to reason about the likely complexities of other geometric problems within polynomial time.

II. NOTATION

Let V be a set of variables with $|V| = n$. An *assignment* on V is a function $V \rightarrow \{0, 1\}$ that assigns every variable a Boolean value. A *restriction* is an assignment on a set $U \subseteq V$. For an assignment α and a variable x , $\alpha(x)$ denotes the value of x under the assignment α .

A *threshold gate* on n variables x_1, \dots, x_n is defined by *weights* $w_i \in \mathbb{R}$ for $1 \leq i \leq n$ and a *threshold* t . The output of the gate is 1, if $\sum_{i=1}^n w_i x_i \geq t$ and 0 otherwise. The *fan-in* of the threshold gate is the number of nonzero weights. We call a variable an input to a gate if the corresponding weight is nonzero. We also extend the definition of a threshold gate to d -ary symmetric gates whose inputs and outputs are d -ary.

For a collection of threshold gates, the *number of wires* is the sum of their fan-ins. A *depth two threshold circuit* consists of a collection of m threshold gates (called the *bottom-level gates*) on the same n variables and a threshold gate (called the *top-level gate*) on the outputs of the bottom-level gates plus the variables. The output of the circuit is the output of the top-level gate. We call a variable with nonzero weight at the top-level gate a *direct wire*. For a d -ary depth two threshold circuit, the gates are d -ary gates and the top-level gate only outputs Boolean values. The number of wires of a depth two threshold circuit is the number of wires of the bottom-level gates. We call a threshold circuit *sparse* if the the number of wires is linear in the number of variables.

A *satisfiability algorithm* for depth two threshold circuits is an algorithm that takes as input a depth two threshold circuit and outputs an assignment such that the circuit evaluates to 1 under the assignment.

A *linear function* on a variable set x_1, \dots, x_n is a function $g(x_1, \dots, x_n) = \sum_{i=1}^n w_i x_i$, where $w_i \in \mathbb{R}$ are called the *coefficients*. The *size* of a linear function is the number of nonzero coefficients. A *linear inequality* is an inequality of the form $g(x_1, \dots, x_n) \geq t$.

An algorithm for the *Integer Linear Programming problem* (ILP) takes as input a collection of linear inequalities on variables x_1, \dots, x_n and outputs an assignment $\{x_1, \dots, x_n\} \rightarrow \mathbb{Z}$ such that all inequalities are satisfied. We call an inequality of the form $0 \leq x_i \leq d - 1$ a *capacity constraint*. In a 0-1 ILP problem each variables is constrained to be 0 or 1.

We use $\tilde{O}(f(n))$ to denote the asymptotic growth of a function f ignoring polynomial factors. Informally, we say an algorithm is *nontrivial*, if its time is significantly better than exhaustive search. If \mathcal{A} is a satisfiability algorithm for circuits with n variables with run time $\tilde{O}(2^{(1-s)n})$, we call s the *savings* of the algorithm over exhaustive search.

III. RESULTS AND TECHNIQUES

The main contribution of this paper is a nontrivial satisfiability algorithm for sparse threshold circuits of depth two. More precisely, we prove the following:

Theorem III.1. *There is a satisfiability algorithm for depth two threshold circuits on n variables with cn wires that runs in time $\tilde{O}(2^{(1-s)n})$ where*

$$s = \frac{1}{c^{O(c^2)}}$$

While the proof in Section V assumes Boolean inputs for simplicity, the proof easily extends to threshold circuits with d -ary inputs, yielding the following corollary.

Corollary III.2. *There is a satisfiability algorithm for depth two threshold circuits on n d -ary variables with cn wires that runs in time $\tilde{O}(d^{(1-s)n})$ where*

$$s = \frac{1}{c^{O(c^2)}}$$

In the following, we provide a high level description of our algorithm. Intuitively, there are two extreme cases for the bottom layer of a linear size threshold circuits of depth two.

The first extreme case is when we have a linear number of gates each with bounded fan-in k . This case is almost equivalent to MAX- k -SAT and can be handled in a way similar to [19], [16]. Consider the family of k -sets of variables given by the support of each bottom-level gate. A probabilistic argument shows that, for some constant c , there exists a subset of about $n - n/(ck)$ variables U so that at most one element from each of the k -sets in the family is outside of U . Then for any assignment to the variables in U , each bottom-level gate becomes either constant or a single literal, and the top-level gate becomes a threshold function of the remaining inputs. To check if a threshold function is satisfiable, we set each variable according to the sign of its weight.

The second extreme case is when we have a relatively small number of bottom-level gates, say, at most ϵn , but some of them might have a large fan-in. In this case, we could first reduce the problem to 0-1 ILP by guessing the truth value of all bottom-level gates and the top gate, and then verifying the consistency of our guesses. Each of our guesses are threshold functions of the variables, so testing consistency of our guesses is equivalent to testing whether the feasible region of about ϵn linear inequalities has a Boolean solution.

We then reduce such an ILP to the Vector Domination problem. To do this, we partition the variables arbitrarily into two equal size sets. For each assignment to the first set, we compute a vector where the i 'th component corresponds to the weighted sum contributed by the first set of variables to the i 'th threshold gate. For the second set of variables, we

do the same, but subtract the contribution from the threshold for the gate. It is easy to see that the vectors corresponding to a satisfying assignment are a dominating pair. Since there are $N = O(2^{n/2})$ vectors in our set, and each vector is of dimension $d = \epsilon n = 2\epsilon \log N$, to get constant savings, we need a Vector Domination algorithm that is subquadratic when the dimension is much less than the logarithm of the number of vectors. The last step is to give such an algorithm, using a simple but delicate divide-and-conquer strategy.

Finally, to put these pieces together, we need to reduce the arbitrary case to a ‘‘convex combination’’ of the two extreme cases mentioned above. To do this, we use the Fan-In Separation Lemma which asserts that there must be a relatively small value of k so that there are relatively few gates of fan-in bigger than k but less than ck , for some constant c . We show that, as in the first extreme case, for a random subset U of variables, the gates with fan-in less or equal to k almost entirely simplify to constants or literals after setting the variables in U . Our selection of k ensures that the number of gates of fan-in greater than k is small relative to the number of remaining variables. So we can apply the method outlined for the second extreme case. The Fan-In Separation Lemma is where our savings becomes exponentially small. Unfortunately, this lemma is essentially tight, so a new method of handling this step would be needed to make the savings polynomially small.

The results translate directly to the feasibility version of sparse integer linear programs with capacity constraints, since it can be expressed as a depth two threshold circuit with an AND gate as the top-level gate. In this formulation, the number of wires are equivalent to the number of nonzero coefficients. We get

Corollary III.3. *Let $\{g_1 \geq a_1, \dots, g_m \geq a_m\}$ be a collection of linear inequalities in variables x_1, \dots, x_n with at most cn nonzero coefficients total. There is an algorithm that finds an integer solution to the linear inequalities with capacity constraints $0 \leq x_i \leq d - 1$ for all i in time $\tilde{O}(d^{(1-s)n})$ for*

$$s = \frac{1}{c^{O(c^2)}}$$

The following two sections contain the details of the proof. Section IV defines the *Vector Domination problem* and, for small dimension, gives an algorithm faster than the trivial quadratic time. The feasibility of a 0-1 ILP with a small number of inequalities is then reduced to the Vector Domination problem, yielding an algorithm for such 0-1 ILP with constant savings. A reduction from depth two threshold circuits to 0-1 ILP concludes that section. In Section V, we show how to reduce the cn -wire depth two threshold circuits satisfiability problem to the special case with a small number of bottom-level gates relative to the number of variables. Section VI discussed a generalization of the algorithm to formulas of arbitrary constant depth.

Section VII considers a generalization of our result to symmetric gates. The satisfiability algorithm for depth two circuits consisting of symmetric gates uses a random restriction technique similar to the algorithm for depth two threshold circuits. However, instead of reducing the problem to 0-1 ILP, the satisfiability problem is reduced to a system of linear equations. While this algorithm generalizes to symmetric gates, it does not allow for arbitrary real weights, but requires that the weights are integer and the *weighted number of wires*, the sum of the absolute weights, is linearly bounded in the number of variables. The result is as follows.

Theorem III.4. *There is a satisfiability algorithm for depth two circuits with symmetric gates and weighted number of wires cn that runs in time $\tilde{O}(2^{(1-s)n})$ where*

$$s = \frac{1}{c^{O(c^2)}}$$

IV. VECTOR DOMINATION PROBLEM

In this section we discuss the *Vector Domination* problem and give an algorithm faster than the trivial $O(n^2)$ for small dimension.

Definition IV.1. Given two sets of d -dimensional real vectors A and B , the *Vector Domination Problem* is the problem of finding two vectors $u \in A$ and $v \in B$ such that $u_i \geq v_i$ for all $1 \leq i \leq d$.

Chan [20] uses the Vector Domination problem to give an algorithm for the all-pairs shortest path problem. The result uses the same algorithm as ours with a different analysis. We present our analysis here for completeness.

Lemma IV.1. *Let $d \in \mathbb{N}$ and $A, B \subseteq \mathbb{R}^d$ with $|A| + |B| = n$. There is an algorithm for the Vector Domination problem that runs in time*

$$O\left(\binom{d + \log n + 2}{d + 1} n\right)$$

Proof: The claim is trivial for $n = 1$ or $d = 1$.

Otherwise, let a be the median of the first coordinates of $A \cup B$. We partition the set A into three sets A^+ , $A^=$ and A^- , where A^+ contains all vectors $u \in A$ such that $u_1 > a$, $A^=$ contains all vectors such that $u_1 = a$ and A^- contains all vectors such that $u_1 < a$. We further partition B into set B^+ , $B^=$ and B^- in the same way. A vector $u \in A$ can only dominate a vector $v \in B$ in one of three cases:

- 1) $u \in A^+$ and $v \in B^+$
- 2) $u \in A^-$ and $v \in B^-$
- 3) $u \in A^= \cup A^+$ and $v \in B^= \cup B^-$

For the first two cases we have $|A^+| + |B^+| \leq \frac{n}{2}$ and $|A^-| + |B^-| \leq \frac{n}{2}$ as we split at the median. For the third case, we know $u_1 \geq v_1$, hence we can recurse on vectors of dimension $d - 1$. Since finding the median takes time $O(n)$ we get for the running time of n vectors of dimension d

$$T(n, d) = 2T\left(\frac{n}{2}, d\right) + T(n, d - 1) + O(n)$$

To solve this recurrence relation, we want to count the number of nodes in the recurrence tree with $n' = \frac{n}{2^i}$ and $d' = d - j$. There are $\binom{i+j}{j} 2^i$ possible paths from the root node to such a node, as in every step we either decrease n or d , and there are $\binom{i+j}{j}$ possible combinations to do so, and if we decrease n there are two possible children. Since computing the median of $\frac{n}{2^i}$ numbers takes time $O(\frac{n}{2^i})$ the total time is upper bounded by

$$\begin{aligned} \sum_{\substack{0 \leq i \leq \log n \\ 0 \leq j \leq d}} \binom{i+j}{j} 2^i O\left(\frac{n}{2^i}\right) &= \sum_{0 \leq i \leq \log n} \binom{i+d+1}{d} O(n) \\ &= \left(\binom{\log n + d + 2}{d + 1} - \binom{d + 1}{d + 1} \right) O(n) \\ &= \left(\binom{d + \log n + 2}{d + 1} - 1 \right) O(n) \end{aligned}$$

We can reduce 0-1 ILP with few inequalities to the Vector Domination Problem.

Corollary IV.2. *Consider a 0-1 Integer Linear Program on n variables and δn inequalities for some $\delta > 0$. Then we can find a solution in time*

$$2^{n/2} \binom{(1/2 + \delta)n}{\delta n} \text{poly}(n)$$

Note that this algorithm is faster than 2^n for $\delta < 0.136$.

Proof: Separate the variable set into two sets S_1 and S_2 of equal size. We assign every assignment to the variables in S_1 and S_2 a δn -dimensional vector where every dimension corresponds to an inequality. Let α be an assignment to S_1 and let $\sum_{j=1}^n w_{i,j} x_i \geq t_j$ be the j -th inequality for all j . Let $a \in \mathbb{R}^{\delta n}$ be the vector with $a_j = \sum_{x_i \in S_1} w_{i,j} \alpha(x_i)$ and let A be the set of $2^{n/2}$ such vectors. For an assignment β to S_2 , let b be the vector with $b_j = t_j - \sum_{x_i \in S_2} w_{i,j} x_i(\beta)$ and let B be the set of all such vectors b .

An assignment to all variables corresponds to an assignment to S_1 and an assignment to S_2 , and hence to a pair $a \in A$ and $b \in B$. The pair satisfies all inequalities if and only if a dominates b . Since $|A| + |B| = 2^{n/2+1}$ and the dimension is δn , we can solve the domination problem in time

$$O\left(\binom{n/2 + \delta n + 3}{\delta n + 1} 2^{n/2+1}\right)$$

We now reduce the satisfiability of a depth two threshold circuit with δn bottom-level gates and any number of direct wires to the union of $2^{\delta n}$ ILP problems.

Corollary IV.3. *Consider a depth two threshold circuit on n variables and δn bottom-level gates for some $\delta > 0$. We allow an arbitrary number of direct wires to the top-level gate. Then there is a satisfiability algorithm that runs in time*

$$2^{\delta n} 2^{n/2} \binom{(1/2 + \delta)n}{\delta n} \text{poly}(n)$$

Note that this algorithm is faster than 2^n for $\delta < 0.099$.

Proof: For every subset U of bottom-level gates, we solve the satisfiability problem under the condition that only the bottom-level gates of U are satisfied. For an assignment to satisfy both the circuit and the condition that only gates in U are satisfied, it must satisfy the following system of inequalities:

- 1) For gates in U with weights w_1, \dots, w_n and threshold t , we have $\sum_{i=1}^n w_i x_i \geq t$.
- 2) For gates not in U we require $\sum_{i=1}^n w_i x_i < t$, which is equivalent to $\sum_{i=1}^n -w_i x_i \geq -t + \varepsilon$ for some small ε .
- 3) Let v_1, \dots, v_n be the weights of the direct wires and let s be the threshold of the top-level gate. Further let w_U be the sum of the weights of the gates in U . Then $\sum_{i=1}^n v_i x_i \geq s - w_U$.

Note that this system contains $\delta n + 1$ inequalities, and the additional dimension adds only a polynomial factor to the time.

Since we need to solve a system of inequalities for every possible subset of bottom-level gates to be satisfied, we have an additional factor of $2^{\delta n}$, which gives the running time as claimed. ■

Williams [9] introduced the reductions used in Corollaries IV.2 and IV.3. He considered a special case of the Vector Domination problem (called the Cooperative Subset Query problem) where the entries in the vectors are 0 and 1 instead of arbitrary real numbers. Applying the reduction from Corollary IV.2 to CNFSAT, he concludes that an algorithm for solving the Cooperative Subset Query problem with $d = \omega(\log n)$ that runs in time $O(f(d)n^\delta)$ for some $\delta < 2$ and a time-constructible f gives a CNFSAT algorithm in time $O(f(m)2^{(\delta/2)^n})$ where m is the number of clauses.

V. FAN-IN SEPARATION

In this section we reduce the satisfiability of a depth two threshold circuit with cn wires to depth two threshold circuits with at most δn bottom-level gates by considering all possible assignments to a random subset U of variables. The goal of the restriction is to eliminate all but a small fraction of gates. U will consist of all but an $O(1/(ck))$ fraction of the variables where k is chosen such that there are only a small number of gates of fan-in larger than k relative to the number of remaining variables. The Fan-In Separation Lemma shows how to find such a k .

Lemma V.1 (Fan-In Separation Lemma). *Let \mathcal{F} be a family of sets such that $\sum_{F \in \mathcal{F}} |F| \leq cn$. Further let $a > 1$ and $\varepsilon > 0$ be parameters. There is an $k \leq a^{c/\varepsilon}$ such that*

$$\sum_{\substack{F \in \mathcal{F} \\ k < |F| \leq ka}} |F| \leq \varepsilon n$$

Proof: Assume otherwise for the sake of contradiction. For $0 \leq i \leq \frac{c}{\varepsilon}$, let f_i be the sum of $|F|$ where $a^i < |F| \leq a^{i+1}$. By assumption we have $f_i > \varepsilon n$ for all i . Hence $\sum_{i=0}^{c/\varepsilon} f_i > cn$, which is a contradiction. ■

Lemma V.2. *Consider a depth two threshold circuit with n variables and cn wires. Let $\delta > 0$ and let U be a random set of variables such that each variable is in U with some probability $1 - p$ independently. There exists a $p = \frac{1}{c^{O(c^2)}}$ such that the expected number of bottom-level gates that depend on at least two variables not in U is at most $3\delta pn$.*

Proof: Let $\varepsilon = \frac{\delta^2}{c}$ and $a = \frac{c^2}{\delta^2}$ and let k be the smallest value such that there are at most εn wires as inputs to gates with fan-in between k and ka . Further let $p = \frac{\delta}{ck}$.

Using the Fan-In Separation Lemma we get $k \leq \left(\frac{c^2}{\delta^2}\right)^{c^2/\delta^2}$. We distinguish three types of bottom-level gates: Small gates, with fan-in at most k , medium gates with fan-in between k and ka , and large gates with fan-in at least ka . For each type of gates, we argue that the expected number of gates that depend on at least two variables not in U is bounded by δpn .

For medium gates, the total number of wires is bounded by $\frac{\delta^2}{c}n$ and each gate contains at least k wires. Hence the number of medium gates is bounded by $\frac{\delta}{ck}\delta n = \delta pn$.

Large gates contain at least ka wires, hence the number of large gates is bounded by $\frac{c}{ka}n = \frac{\delta}{ck}\delta n = \delta pn$.

For small gates, we argue as follows. Let m be the number of small gates and let l_1, \dots, l_m be their fan-ins. Let X_i denote the event that gate i depends on at least two variables not in U and let X be the number of such events. We have $P(X_i) \leq \binom{l_i}{2}p^2 \leq l_i^2 p^2$ and therefore

$$E[X] = \sum_{i=1}^m P(X_i) \leq \sum_{i=1}^m l_i^2 p^2 \leq p^2 kcn = \delta pn$$

■

Lemma V.3. *There is a satisfiability algorithm for depth two threshold circuits with cn wires that runs in time $2^{(1-s)n}$ for $E[s] = \frac{1}{c^{O(c^2)}}$.*

Proof: Let $\delta = \frac{1}{48}$ and U as well as other parameters be as above. For every assignment to U , we have a depth two threshold circuit with pn variables and $3\delta pn$ bottom-level gates in expectation. Since $3\delta < 0.099$, we can decide the satisfiability of such a circuit using Corollary IV.3 with constant savings. Let s' be the savings with our parameters.

Let T be the time for carrying out the entire procedure. Since we are interested in the expected savings we consider the logarithm of the time and get

$$E[\log(T)] = (1 - p)n + (1 - s')pn = (1 - s')pn$$

and the lemma follows from $p = \frac{1}{c^{O(c^2)}}$. ■

Since s is bounded by above by 1, we can repeat the process a constant number of times until we find a restriction

such that the savings is at least its expectation. This gives us our main result Theorem III.1.

VI. GENERALIZATION TO FORMULAS

In this section we discuss an extension of our main result to linear size, constant depth *threshold formulas*. A formula is a circuit such that the output of every gate is an input to at most one other gate. A formula can be viewed as a tree where the internal nodes correspond to gates and the leaves to bottom variables. Note that a circuit of depth two is always a formula. The proof is a direct generalization of our main proof.

Corollary VI.1. *There is a satisfiability algorithm for depth d threshold formulas with cn wires that runs in time $\tilde{O}(2^{(1-s)n})$ where*

$$s = \frac{1}{((d-1)c)^{O(((d-1)c)^2)}}$$

Proof sketch: The algorithm chooses a random restriction such that at most δn gates depend on more than one variable after restriction, where $\delta < \frac{1}{16}$ as before. As in the original proof, we take into account that there is only a single top-level gate, which does not simplify after restriction. The main difference to our main proof is the notion of the fan-in. Instead of considering the number of inputs to a gate, consider the *size* of a gate. The size of a gate is the size of the subtree rooted at that gate. It is also an upper bound to the number of variables the gate depends on.

For all $i \leq d$, the sum of sizes of all gates at depth i is at most cn , since the circuit is a tree with at most cn wires. Hence the sum of sizes of all gates (minus the top-level gate) is at most $(d-1)c$.

Using the Fan-In Separation Lemma we can select a set U of size pn where $p = \frac{1}{((d-1)c)^{O(((d-1)c)^2)}}$ such that the number of gates that depend on at least two variables not in U is at most δn . We can then write each remaining gate as a linear inequality, which allows us to apply Corollary IV.2. ■

VII. GENERALIZATION TO SYMMETRIC GATES

In this section we discuss a second extension, to *symmetric gates*. A gate is symmetric if the output depends only on the weighted sums of the inputs. In particular, threshold gates are symmetric. The proof of our main result does not directly generalize to symmetric gates, but we give a different algorithm to decide the satisfiability of depth two circuits consisting of symmetric gates that follows similar ideas as our main proof. For this algorithm we do however require that the weights are integer and small. Specifically, we define the *weighted fan-in* of a gate as the sum of the absolute weights and the *weighted number of wires* as the sum of the fan-ins of all the gates. The result applies to circuits with a weighted fan-in of cn .

The main difference between the two algorithms is the problem we reduce it to after applying a restriction. In our main result, we reduce the satisfiability of the simplified circuit to a (small) system of linear inequalities. Here, we reduce to a system of linear equations. We first give an algorithm for linear equations.

Lemma VII.1. *There is an algorithm to find a Boolean solution to a system of linear equations on variables $\{x_1, \dots, x_n\}$ in time $\tilde{O}(2^{n/2})$.*

Proof: We first reduce the problem to subset sum. Let $w_{i,j}$ be the weight of x_i in the j -th equation and let r_j be right-hand side of the j -th equation. Further let $D = n \cdot \max_{i,j} \{w_{i,j}, r_j\}$. We define $s_i = \sum_j w_{i,j} D^j$ and $s = \sum_j r_j D^j$. Then there is a solution to the system of linear equations if and only if there is a subset of the s_i that sums to s .

To solve the subset sum problem, partition the set of s_i into two sets of equal size and list all $2^{n/2}$ possible subset sums each. We can then sort the lists in time $O(2^{n/2}n)$ and determine if there is a pair of numbers that sums to s . ■

We reduce the satisfiability problem of depth two threshold circuits with small integer weights to a system of linear equations to get the following result.

Theorem VII.2 (Theorem III.4 restated). *There is a satisfiability algorithm for depth two circuits with symmetric gates and weighted number of wires cn that runs in time $\tilde{O}(2^{(1-s)n})$ where*

$$s = \frac{1}{c^{O(c^2)}}$$

As before, we pick a random restriction with some parameter p , such that most gates depend on at most one variable.

Given an assignment, we distinguish between the Boolean output of a gate and the *value*, which is defined as the weighted sum of the inputs. Note that the value uniquely defines the output of a symmetric gate. Unlike our main proof, we guess the value of the remaining gates, including the top-level gate. Given a value for every gate, we can write a system of linear equation that is satisfied if and only if the assignment is consistent with the values of the gates. We then solve the system of linear equations on n Boolean variables in time $\tilde{O}(2^{n/2})$ using Lemma VII.1.

While Lemma VII.1 gives some savings over brute force, we need to solve a separate system of linear equations for every set of possible values of the gates. We call the number of systems the *overhead*. To get any savings, we need the overhead for guessing the values to be smaller than the savings achieved with solving the system of linear equations. For this, it is crucial that both the number of remaining gates and the number of values that can be obtained is small. Here we use the requirement that the weights are small. Section VII-A contains the details of this calculation.

One possible approach would be to select p using a fan-in separation technique. However, we only achieved savings that are doubly exponentially small in c using this approach. To get better savings, it is useful to model the interplay between the parameter p and the circuit as an explicit zero-sum game, where the first player's (the algorithm designer) pure strategies are the values of p and the second player's (the circuit designer) pure strategies are circuits where all the gates have the same fan-in. The payoff is the difference between the saving of solving the subset sum problem and the overhead of guessing the values of the gates.

The mixed strategies of the circuit designer are circuits of symmetric gates with a weighted number of wires of at most cn , where each such circuit is viewed as a distribution of the total number of wires among gates of different weighted fan-in. The mixed strategies of the algorithm designer are distributions on the values of p . We then apply the Min-Max theorem to lower bound the expected value of the game by exhibiting a distribution (with finite support) on the values of p . We search through the values in the support of the distribution to find a p that produces the expected value. This novel game-theoretic analysis yields an overall savings which is only single exponentially small in c . Section VII-B contains the details of the Min-Max approach.

A. The Algorithm

We develop the algorithm of Theorem VII.2 in three stages. In this section, we consider p a parameter and present a satisfiability algorithm for depth two circuits with symmetric gates and weighted number of wires of cn . We further assume that all the bottom-level gates have the same weighted fan-in f . The algorithm achieves savings $s_{p,f}$ and for certain combinations of p and f the savings might be negative. In the second stage we extend the algorithm to circuits with varying fan-in and show that the savings of the algorithm is a convex combination of $s_{p,f}$. In the last stage, in Section VII-B we show how to select a p such that the savings is at least $\frac{1}{c^{O(c^2)}}$ for any distribution on f .

As we are mainly interested in the savings, we look at the logarithm of the time complexity and bound its expectation.

Lemma VII.3. *Let $0 \leq p \leq 1$ be a parameter and C be a depth two circuit with symmetric gates, variables $V = \{x_1, \dots, x_n\}$, a weighted number of wires of cn , and weighted fan-in f for all bottom-level gates. There is an algorithm that decides the satisfiability for such C with time complexity T such that $\mathbf{E}[\log(T)] = (1 - s_{p,f})n$ for*

$$s_{p,f} = \begin{cases} \frac{p}{4} & \text{if } pf < \frac{1}{4c} \\ \frac{p}{2} - \frac{c}{f} \log(8cpf) & \text{otherwise} \end{cases}$$

Proof: We select a random subset $U \subseteq V$ such that a variable is in U with probability $(1-p)$ independently. We note that $\mathbf{E}[|U|] = (1-p)n$. For each of the $2^{|U|}$ assignments to U , we solve the satisfiability problem of the simplified

circuit. Bottom-level gates where all inputs are in U are removed and the top-level gate is adjusted appropriately. Gates that only depend on one input are replaced by a direct wire to the top-level gate with an appropriate weight and adjustment to the top-level gate. For all gates with at least two remaining inputs, we guess the value of the gate and express the gate as a linear equation. Similarly, we guess the value of the top-level gate to get another linear equation. We then solve the resulting system of linear equations on $n' = n - |U|$ variables in time $\tilde{O}(2^{n'/2})$ using Lemma VII.1.

The critical part of the analysis is bounding the overhead from guessing the values of the gates. We first bound the number of distinct values a gate can take. The top-level gate can only take polynomially many different values. Consider a bottom-level gate with weighted fan-in $l \geq 2$ after applying an assignment to the variables in U . The number of possible inputs, and hence the number of possible values is bounded by 2^l . On the other hand, since the value is an integer between $-l$ and l , the number of possible values for the gates is also upper bounded by $2l + 1$. Hence, we use $\min\{2^l, 2l + 1\}$ as an upper bound for the number of values of a bottom-level gate with weighted fan-in l .

Our overhead crucially depends on the number of exceptional gates, gates that depend on more than one variable after applying an assignment to the variables in U . Intuitively, if the fan-in of a gate is small, then we expect that it will be simplified to depend on at most one variable after assigning values to the variables in U . On the other hand, there cannot be too many gates of large fan-in. While this intuition is simple, it is tricky to make it work for us in the general context. At this stage, our focus is on estimating the savings $s_{p,f}$ for the probability parameter p and weighted fan-in f .

Let H be a random variable denoting the number of possible values the remaining gates can obtain. Our estimation of H and $s_{p,f}$ involves two cases. Let $t = \frac{1}{4c}$. We first consider the case $pf < t$. Let $U' \subseteq V - U$ be the set of variables that appear in exceptional gates. Our goal is to upper bound $\mathbf{E}[\log(H)] \leq \mathbf{E}[|U'|]$.

Consider a bottom-level gate. Let X be the random variable denoting the number of its inputs not in U . Let $f' \leq f$ be the fan-in of the gate (remember f is the *weighted* fan-in), and let X be the random variable denoting the number of its inputs not in U . The distribution of X is $\text{Bin}(f', p)$, hence we have $\mathbf{E}[X] = f'p$. Let the random variable Y denote the number of variables that the gate can contribute to U' . Since U' is the set of variables appearing in exceptional gates, we have $Y = X$ for $X \geq 2$ and $Y = 0$

otherwise. Hence

$$\begin{aligned}\mathbf{E}[Y] &= \mathbf{E}[X] - \mathbf{P}[X = 1] \leq f'p - f'p(1-p)^{(f'-1)} \\ &\leq f'p(1 - (1-p)^{f'}) \leq f'p(1 - (1-f'p)) \\ &= (f'p)^2 \leq (fp)^2\end{aligned}$$

by Bernoulli's inequality. Hence, for any variable x which is an input to the gate, the probability x belongs to U' is at most $\frac{\mathbf{E}[Y]}{f} \leq p^2 f \leq \frac{p}{4c}$. Since the total number of wires is bounded by cn , we have $\mathbf{E}[\log(H)] \leq \mathbf{E}[|U'|] = \frac{p}{4}n$.

For the logarithm of the time complexity this yields

$$\begin{aligned}\mathbf{E}[\log(T)] &= \mathbf{E}[|U|] + \mathbf{E}\left[\frac{1}{2}(n - |U|)\right] + \frac{p}{4}n + O(\log n) \\ &\leq n\left(1 - \frac{p}{4}\right) + O(\log n)\end{aligned}$$

where the logarithmic summand stems from guessing the value of the top-level gate. We have $s_{p,f} = \frac{p}{4}$.

We now consider the case $pf \geq t$. Suppose the i -th gate has l_i inputs that are not in U . The expected value of l_i is pf . There are at most $2l_i + 1$ possible values for the gate. Since all the bottom-level gates have the same weighted fan-in f , the number of bottom-level gates is at most cn/f and $\mathbf{E}[\sum_{i=1}^{cn/f} l_i] = pcn$. We bound the expected logarithm of the number of possible values of all gates by

$$\begin{aligned}\mathbf{E}\left[\log\left(\prod_{i=1}^{cn/f} (2l_i + 1)\right)\right] &= \frac{cn}{f} \sum_{i=1}^{cn/f} \mathbf{E}\left[\left(\log(2l_i + 1) \frac{f}{cn}\right)\right] \\ &\leq \frac{cn}{f} \log(2pf + 1) \\ &\leq \frac{cn}{f} \log(8cpf)\end{aligned}$$

where we use the concavity of the logarithm function in the penultimate step and the fact $pf \geq \frac{1}{4c}$ in the last step.

For the logarithm of the time complexity we get,

$$\begin{aligned}\mathbf{E}[|U|] + \mathbf{E}\left[\frac{1}{2}(n - |U|)\right] &+ cn/f \log(8cpf) + O(\log n) \\ &\leq n\left(1 - \left(\frac{p}{2} - \frac{c}{f} \log(8cpf)\right)\right) + O(\log n)\end{aligned}$$

with savings $s_{p,f} = \frac{p}{2} - \frac{c}{f} \log(8cpf)$. \blacksquare

We now extend the algorithm to circuits with varying fan-in and show that the logarithm of the time complexities is lower bounded by a convex combination of the savings $s_{p,f}$. We model the cn -wire circuits of varying weighted fan-in by a distribution \mathcal{F} on wires. For each weighted fan-in f , the wire distribution \mathcal{F} specifies the number $c_f n$ of wires of bottom-level gates of weighted fan-in f . We denote the savings of our algorithm on circuits with wire distribution \mathcal{F} by $s_{p,\mathcal{F}}$.

Lemma VII.4. *Let $0 \leq p \leq 1$ be a parameter and C be a depth two circuit with symmetric gates, n variables and a weighted number of wires of cn , where the wires*

are distributed according to \mathcal{F} . There is a satisfiability algorithm for such C with time complexity T such that $\mathbf{E}[\log(T)] = (1 - s_{p,\mathcal{F}})n$ for

$$s_{p,\mathcal{F}} \geq \sum_{f=1}^n \frac{c_f}{c} s_{p,f}$$

Proof: The algorithm is the same as above. The logarithm of the overhead of guessing the values for all bottom-level gates with fan-in f is $\log(H_f) = \frac{c_f n}{f} \log(8cpf)$ if $pf \geq t$ and $\log(H_f) = \frac{c_f}{c} \frac{p}{4} n$ otherwise. Solving the system of linear equations and using linearity of expectation then yields the savings as claimed. \blacksquare

B. The Algorithm as a Zero-Sum Game

The time complexity of the algorithm in Section VII-A depends crucially on choosing a suitable parameter p . Instead of trying to directly determine a good parameter p by analyzing the wire distribution of the circuit, we apply a trick from game theory.

A zero-sum game with two players **A** and **C** is a game where both players pick a strategy and the outcome is determined by a function of the two strategies. Player **A** tries to maximize the outcome, while player **C** tries to minimize it. The Min-Max Theorem states that it does not matter which player moves first, as long as we allow mixed strategies for the players.

We model the task of choosing the parameter p as the following zero-sum game: Player **A**, the algorithm designer, picks some probability p , and player **C**, the circuit designer, picks a value f . The outcome is $s_{p,f}$, the savings of the algorithm. The algorithm designer tries to maximize the savings, and the circuit designer tries to minimize it. The wire distribution of a circuit is a mixed strategy for the circuit designer. A mixed strategy for the algorithm designer **A** would be a distribution on the probabilities.

A direct approach for designing the algorithm would be to select the parameter p by analyzing the circuit. Specifically, given the wire distribution of the circuit \mathcal{F} , the algorithm designer picks a p and the outcome $s_{p,\mathcal{F}}$ is a convex combination of the values $s_{p,f}$. Using the Min-Max Theorem we turn this game around: The algorithm designer picks a mixed strategy and the circuit designer responds with a pure strategy f , a circuit where all bottom-level gates have weighted fan-in f . The following lemma shows that there is a good strategy for the algorithm designer.

Lemma VII.5. *There is a distribution \mathcal{D} on parameters p such that for all f ,*

$$\mathbf{E}_{p \sim \mathcal{D}}[s_{p,f}] \geq \frac{1}{c^{O(c^2)}}$$

Proof: Let \mathcal{D} be the following distribution on p : For $I = O(c^2 \log(c))$ with suitable constants, and $1 \leq i \leq I$, we set $p = 2^{-i}$ with probability $A \cdot 2^{-(I-i+1)}$, where

$A = \frac{1}{\sum_{i=1}^I 2^{-(I-i+1)}}$ is the normalization factor. We know that $1 \leq A \leq 2$. The expectation of p is $\mathbf{E}[p] = AI2^{-I-1}$.

Let f be any pure strategy of the circuit designer and $J = \log(f)$. The expected outcome of the game for these strategies is

$$\mathbf{E}_{p \sim \mathcal{D}}[s_{p,f}] = \sum_{i=1}^I 2^{-(I-i+1)} s_{2^{-i}, 2^J}.$$

To lower bound the expected outcome, we use a case analysis on the savings similar to the one in Section VII-A. Let $t = \frac{1}{4c}$ as defined in the previous section. Let $I' \leq I$ be the largest value such that for $i \leq I'$, we have $2^{J-i} \geq t$ and for $I' < i \leq I$ we have $2^{J-i} < t$.

Using the savings from Lemma VII.3, we have $s_{2^{-i}, 2^J} = 2^{-i-1} - \frac{c}{2^J} \log(c2^{J-i+1})$ for $2^{J-i} \geq t$ and $s_{2^{-i}, 2^J} = 2^{-i-2}$ otherwise. The expected savings is then

$$\begin{aligned} \mathbf{E}_{p \sim \mathcal{D}}[s_{p,f}] &= \sum_{i=1}^I 2^{-(I-i+1)} s_{2^{-i}, 2^J} \\ &= \sum_{i=1}^{I'} 2^{-(I-i+1)} \left(2^{-i-1} - \frac{c}{2^J} \log(c2^{J-i+3}) \right) \\ &\quad + \sum_{i=I'+1}^I 2^{-(I-i+1)} 2^{-i-2} \\ &\geq \sum_{i=1}^{I'} 2^{-(I+3)} \\ &\quad - \sum_{i=1}^{I'} 2^{-(I-i+1)} \frac{c}{2^J} \log(c2^{J-i+3}) \\ &= \frac{1}{2^{I+1}} \left(\frac{I}{4} - c \sum_{i=1}^{I'} 2^{-(J-i)} \log(c2^{J-i+3}) \right) \end{aligned}$$

Let $j = \lceil (J-i) \rceil$. By the definition of I' we have $j \geq \log(t) = -\log(c) - 2$. Hence

$$\begin{aligned} \sum_{i=1}^{I'} 2^{-(J-i)} \log(c2^{J-i+3}) &\leq \sum_{j=\log(t)}^{\infty} 2^{-j} (j + \log(8c)) \\ &\leq 8c \log(8c) + \sum_{j=1}^{\infty} j 2^{-j} + \log(8c) \\ &= O(c \log(c)) \end{aligned}$$

Hence for $I = O(c^2 \log(c))$ we get the claim. \blacksquare

We now conclude that for every f there is a $p = 2^{-i}$ for $1 \leq i \leq I$, such that $s_{p,f} \geq \frac{1}{c^{O(c^2)}}$. Using that for every mixed strategy for f , the savings is a convex combination of the savings for pure strategies, we conclude the same for any strategy on f .

This gives us the final algorithm: Given a circuit C with wire distribution \mathcal{F} , evaluate $\mathbf{E}_{f \sim \mathcal{F}}[s_{p,f}]$ with $p = 2^{-i}$ for

each $1 \leq i \leq I$ as above and use the optimal p for the random restriction.

The savings is tight in the sense that there is a mixed strategy on f such that the expected savings is at most $1/2^{\Omega(c)}$.

Lemma VII.6. *There is a wire distribution \mathcal{F} such that for any p*

$$\mathbf{E}_{f \sim \mathcal{F}}[s_{p,f}] \leq \frac{1}{2^{\Omega(c)}}$$

Proof: Let p be the strategy of the algorithm designer and let \mathcal{F} be the distribution such that for $1 \leq j \leq c$, $c_{2^j} = 1$ and $c_f = 0$ for any other f . By lemma VII.4 we have

$$\mathbf{E}_{f \sim \mathcal{F}}[s_{p,f}] = \sum_{j=1}^c \frac{1}{c} s_{p, 2^j}$$

We argue that for large c and $p \geq \frac{1}{2^c}$, the savings is negative. Assume $p \geq \frac{1}{2^c}$. There is some $j^* \leq c$ such that for $f = 2^{j^*}$, $1 \leq pf \leq 2$. Using that for any p and f , the savings $s_{p,f}$ is upper bounded by $\frac{p}{2}$ we get

$$\begin{aligned} \mathbf{E}_{f \sim \mathcal{F}}[s_{p,f}] &= \sum_{j=1}^c \frac{1}{c} s_{p, 2^j} \\ &\leq \frac{p}{2} - \frac{1}{c} s_{p, 2^{j^*}} \\ &= \frac{p}{2} - \frac{1}{c} \frac{c}{2^{j^*}} \log(cp2^{j^*+3}) \\ &\leq \frac{p}{2} (1 - (\log(8c) + 1)) \end{aligned}$$

For large c , the expectation is therefore negative. On the other hand, if $p \leq \frac{1}{2^c}$, then $\mathbf{E}_{f \sim \mathcal{F}}[s_{p,f}] \leq \frac{1}{2^{c-1}}$. \blacksquare

VIII. CONCLUSION

In this paper, we present the first nontrivial algorithm for deciding the satisfiability of cn -wire threshold circuits of depth two. The same result also applies to the special case of 0-1 Integer Linear Programming with sparse constraints. The algorithm improves over exhaustive search by a factor 2^{sn} where $s = 1/c^{O(c^2)}$.

This paper also presents a satisfiability algorithm for depth two circuits consisting of symmetric gates with small integer weights.

Several straightforward open questions remain. Can we further improve the savings? The savings in our algorithm is exponentially small in c , while the best known savings for cn -size \mathbf{AC}^0 circuits is only polylogarithmically small in c [18]. Can we decrease this gap? If not, can we explain it in terms of the expressive power of the circuits?

Our algorithm handles only linear size threshold circuits of depth two. Can we obtain nontrivial satisfiability algorithms for slightly more relaxed models? For example, it would be very interesting to extend the result to larger depth circuits. It would also be nice to generalize the algorithm to

deal with depth two threshold circuits with linearly many gates (as opposed to a linear number of wires).

It would also be interesting to relax the restriction on the number of wires. Unfortunately, as discussed earlier, it is not possible to obtain a constant savings algorithm for depth two threshold circuits of superlinearly many wires under **SETH**.

It would be extremely interesting to find a subquadratic algorithm for the Vector Domination Problem for dimension $\omega(\log n)$, which would imply the refutation of **SETH**.

Our algorithm is a “Split and List” algorithm [9], split the variable set into subsets and list all assignments to the subsets. As such, it inherently takes exponential space. Can we reduce the space requirement to polynomial space?

ACKNOWLEDGMENT

We thank Dominik Scheder, Shachar Lovett, Ryan Williams and the anonymous reviewers for various helpful comments.

This research is supported by NSF grant CCF-1213151 from the Division of Computing and Communication Foundations. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation

REFERENCES

- [1] S. Cook, “The complexity of theorem-proving procedures,” in *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, 1971, pp. 151–158.
- [2] L. Levin, “Universal sorting problems,” *Problems of Information Transmission*, vol. 9, pp. 265–266, 1973.
- [3] R. Williams, “Improving exhaustive search implies superpolynomial lower bounds,” in *Proceedings of the 42nd ACM symposium on Theory of computing*, ser. STOC ’10. New York, NY, USA: ACM, 2010, pp. 231–240. [Online]. Available: <http://doi.acm.org/10.1145/1806689.1806723>
- [4] —, “Non-Uniform ACC Circuit Lower Bounds,” in *Proceedings of the Twenty-Sixth Annual IEEE Conference on Computational Complexity*, 2011.
- [5] A. K. Chandra, L. Stockmeyer, and U. Vishkin, “Constant depth reducibility,” *SIAM Journal on Computing*, vol. 13, no. 2, pp. 423–439, 1984.
- [6] R. B. Boppana and M. Sipser, “Handbook of theoretical computer science (vol. a),” J. van Leeuwen, Ed. Cambridge, MA, USA: MIT Press, 1990, ch. The complexity of finite functions, pp. 757–804. [Online]. Available: <http://dl.acm.org/citation.cfm?id=114872.114886>
- [7] A. Hajnal, W. Maass, P. Pudlak, M. Szegedy, and G. Turan, “Threshold circuits of bounded depth,” in *Proceedings of the 28th Annual Symposium on Foundations of Computer Science*, ser. SFCS ’87. Washington, DC, USA: IEEE Computer Society, 1987, pp. 99–110. [Online]. Available: <http://dx.doi.org/10.1109/SFCS.1987.59>
- [8] R. Impagliazzo, R. Paturi, and M. E. Saks, “Size–depth tradeoffs for threshold circuits,” *SIAM J. Comput.*, vol. 26, no. 3, pp. 693–707, 1997, preliminary version published in STOC 1993.
- [9] R. Williams, “A new algorithm for optimal 2-constraint satisfaction and its implications,” *Theoretical Computer Science*, vol. 348, pp. 357–365, 2005.
- [10] R. Santhanam, “Fighting peregbor: New and improved algorithms for formula and qbf satisfiability,” in *Proceedings of the 2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, ser. FOCS ’10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 183–192. [Online]. Available: <http://dx.doi.org/10.1109/FOCS.2010.25>
- [11] R. Impagliazzo and R. Paturi, “The complexity of k -SAT,” *Journal of Computer and Systems Sciences*, vol. 62, no. 2, pp. 367–375, March 2001, preliminary version in *14th Annual IEEE Conference on Computational Complexity*, pages 237–240, 1999.
- [12] R. Impagliazzo, R. Paturi, and F. Zane, “Which problems have strongly exponential complexity?” *Journal of Computer and System Sciences*, vol. 63, pp. 512–530, 1998, preliminary version in *39th Annual IEEE Symposium on Foundations of Computer Science*, pp 653-662, 1998.
- [13] C. Calabro, “The exponential complexity of satisfiability problems,” Ph.D. dissertation, University of California, San Diego, 2009.
- [14] R. Schuler, “An algorithm for the satisfiability problem of formulas in conjunctive normal form,” *Journal of Algorithms*, vol. 54, no. 1, pp. 40–44, 2005.
- [15] E. Dantsin and A. Wolpert, “Max-sat for formulas with constant clause density can be solved faster than in $\mathcal{O}(2^n)$ time,” in *Theory and Applications of Satisfiability Testing - SAT 2006*, ser. Lecture Notes in Computer Science, A. Biere and C. Gomes, Eds. Springer Berlin Heidelberg, 2006, vol. 4121, pp. 266–276.
- [16] P. Austrin, S. Benabbas, A. Chattopadhyay, and T. Pitassi, 2012, personal Communication.
- [17] R. Impagliazzo, R. Paturi, and S. Schneider, “0-1 Integer Linear Programming with a Linear Number of Constraints,” 2013, unpublished manuscript.
- [18] R. Impagliazzo, W. Matthews, and R. Paturi, “A Satisfiability Algorithm for \mathbf{AC}^0 ,” in *Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms*, 2012.
- [19] C. Calabro, R. Impagliazzo, and R. Paturi, “The complexity of satisfiability of small depth circuits,” in *Parameterized and Exact Computation: 4th International Workshop, IWPEC 2009, Copenhagen, Denmark, September 10-11, 2009, Revised Selected Papers*. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 75–85.
- [20] T. M. Chan, “All-pairs shortest paths with real weights in $\mathcal{O}(n^3/\log n)$ time,” in *Algorithms and Data Structures*. Springer, 2005, pp. 318–324.