IEICE TRANS. INF. & SYST., VOL.E98–D, NO.1 JANUARY 2015

PAPER

A Satisfiability Algorithm for Some Class of Dense Depth Two Threshold Circuits

Kazuyuki AMANO[†a)], ***Member and*** **Atsushi SAITO**[†], ***Nonmember***

SUMMARY Recently, Impagliazzo et al. constructed a nontrivial algorithm for the satisfiability problem for *sparse* threshold circuits of depth two which is a class of circuits with cn wires. We construct a nontrivial algorithm for a larger class of circuits. Two gates in the bottom level of depth two threshold circuits are dependent, if the output of the one is always greater than or equal to the output of the other one. We give a nontrivial circuit satisfiability algorithm for a class of circuits which *may not be sparse* in gates with dependency. One of our motivations is to consider the relationship between the various circuit classes and the complexity of the corresponding circuit satisfiability problem of these classes. Another background is proving strong lower bounds for TC^0 circuits, exploiting the connection which is initiated by Ryan Williams between circuit satisfiability algorithms and lower bounds.

key words: *satisfiability, exact algorithm, threshold circuit*

1. Introduction

Satisfiability problem gives both an integral view on theory of NP-complete problems which is firstly defined in [6] and [14], and one of the most useful methods for constraint satisfaction problems in engineering and other practical fields. In particular, heuristic ways on CNF SAT are applied in the practical area of various combinatorial search problems such as boolean circuit design verification.

There are several well known computational problems related to satisfiability problems. The first one is satisfiability for CNF formulas and its generalization, because CNF is one of fundamental concepts about boolean functions. For example, Santhanam [17] gives an algorithm with a nontrivial exponent for linear size formulas of AND and OR gates with fan-in two. The second one is MAX-k-SAT, the optimization version of k-CNF SAT. Even for MAX-3-SAT, no algorithms with constant savings over brute force search are known while such an algorithm is constructed for MAX-2-SAT in [19]. The third one is Integer Linear Programming (ILP) that is very useful in expressing combinatorial optimization problems both in theory and practice.

Satisfiability for depth two threshold circuits contains these problems as special cases. Satisfiability for CNF formula can be solved by algorithms solving satisfiability for depth two threshold circuits. We should note that we do not obtain a nontrivial algorithm for depth two threshold circuit satisfiability algorithm as a corollary of the result in [17]. The reason is that known transformation from a linear size threshold circuit to formula over AND and OR gates yields a quadratic blow-up of size. MAX-k-SAT, the optimization version of k-CNF SAT, can be computed by algorithms solving satisfiability for depth two threshold circuits, since we can regard the top threshold gate as a counting device of the number of satisfied CNFs and an objective function in an optimization problem. Finally, 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 an AND gate. So understanding satisfiability of depth two threshold circuits could give us various view points on both theoretical and practical areas including the above three problems.

In the paper [13], Impagliazzo et al. constructed the first nontrivial algorithm with constant savings in the exponent over brute force search for the satisfiability of *sparse* depth two threshold circuits which has cn-wires for every constant c. As a consequence, they also got a similar result for linear-size ILP. Here we say an algorithm is nontrivial, if its running time is bounded above by $2^n/w(n)$ where n is the number of input variables and $w(n)$ is a super-polynomial function in n. Note that 2^n is just the number of assignments to n input variables. Their main subroutine is an algorithm for the Vector Domination Problem: given n vectors in $\mathbb{R}^d$, decide whether there is a pair of vectors such that the first vector is larger than the second vector in each coordinate. Relationship between this problem and satisfiability problem is studied in [19].

The Strong Exponential Time Hypothesis (SETH) is a well known conjecture about limitations of efficiency of satisfiability algorithms. The statement of SETH is that for every $\delta < 1$ there is a k such that k-SAT cannot be solved in time $O(2^{\delta n})$. In particular, an algorithm with constant savings for depth two threshold circuits of super linear size would violate SETH [9], since k-CNF for all k can be reduced through Sparsification Lemma [10] to superlinear size depth two threshold circuits [3]. Some algorithms solving CNF-SAT and MAX-SAT with constant savings when the formula is linear size are given in [17] and [7]. Assuming the SETH, we can not to solve satisfiability problem for super linear size depth two threshold circuits with constant savings as a direct extension of the result in [13].

Thus one of natural directions relating with this result is extending classes of input circuits and constructing an algorithm with constant savings under the SETH for such classes. Considering algorithms for a class of circuits of

Manuscript received April 23, 2014.
Manuscript revised August 29, 2014.

[†]The authors are with the Department of Computer Science, Gunma University, Kiryu-shi, 376–8515 Japan.
a) E-mail: amano@cs.gunma-u.ac.jp

DOI: 10.1587/transinf.2014EDP7127

polynomial size is also crucial to circuit complexity theory. Ryan Williams proved that nontrivial improvement over the brute force search for the general boolean circuit satisfiability problem implies circuit lower bounds [20]. He also applied this technique to prove super polynomial size bounds for ACC^0 circuits using a novel nontrivial satisfiability algorithm for ACC^0 circuits, solving a long standing open problem [21]. Because of the connection to circuit lower bounds the power of threshold circuits is extensively studied in [2] and [5]. The class TC^0, which is a class of constant depth polynomial size threshold circuits, is a well known natural circuit class larger than ACC^0. Current understanding of the limitations of bounded depth threshold circuits is, however, inadequate. Exponential lower bounds for such circuits are only showed for the limited class of depth two and bounded weight [8]. For larger depth circuits, it is barely super linear lower bounds that we obtained on the number of wires [12]. Very recently, Ryan Williams obtained a remarkable result toward the bounded depth threshold circuit lower bounds: circuit lower bounds for the class $\mathsf{ACC}^0 \circ \mathsf{THR}$, using a sophisticated modification of the previous results [22], but nontrivial satisfiability algorithm for polynomial size depth two threshold circuits is not given. For all above reasons, it is significant to give algorithms for an explicit class which is a subclass of depth two threshold circuits of super linear size.

In this paper, we construct a nontrivial algorithm for a larger subclass of depth two threshold circuits with polynomial number of gates. Two gates in the bottom level of depth two threshold circuits are dependent, if the output of the one is always greater than or equal to the output of the other one. We give a nontrivial circuit satisfiability algorithm for a class of circuits which *may not be sparse* in gates with dependency and which have polynomial size. Lets consider the following parameterized problem: for given depth two threshold circuit C of size n^c which is sparse in independent gates such that there exists an unique *maximal* independent gate set I' of size greater than k , compute YES iff C is satisfiable, where c is a constant and parameter k in this problem k is the maximum size of independent gate set of C except I'. This improves the previous result in [13] in the sense that we can construct a nontrivial algorithm when we relax some condition on sparsity in input circuits. We can consider a family of circuits with high dependency which can compute all boolean functions.More details about this fact are mentioned in the later.

The rest of the paper is as follows. Firstly, we define several notions being necessary in other sections in Sect. 2 and formally state our results in Sect. 3. We give an overview of the entire algorithm in Sect. 4. In Sect. 4, we also define a problem: for given a circuit and a graph containing information about dependency of the circuit output YES if and only if the circuit is satisfiable. We give a constructive proof of a reduction from our original circuit satisfiability problem to this problem in Sect. 5. In Sect. 6, we solve the problem defined in Sect. 4. In Sect. 7, we give an algorithm whose subroutine is to solve the problem in Sect. 5 to solve the original satisfiability problem. Finally, we discuss future work in the last section.

2. Preliminaries

A threshold gate which outputs a boolean value has the label $w_1x_1 + \cdots w_nx_n \geq t$, where each of $w_1, \ldots, w_m$ and t is a real number, and $x_1, \ldots, x_n$ are input boolean variables. For all boolean inputs $(x_1, \ldots, x_m)$, it outputs 1 if and only if the statement of the label holds. We give a more precise definition as follows.

Definition 2.1: Let $x_1, \ldots, x_n$ be boolean variables. Let $w_1, \ldots, w_n, t$ be real numbers. We define a threshold gate as a gate computing a boolean function $THR_{w_1,\ldots,w_n,t}(x_1, \ldots, x_n)$ such that $THR_{w_1,\ldots,w_n,t}(x_1, \ldots, x_n) = 1 \iff \Sigma_{i=1}^{n} w_i x_i \geq t$. A depth two threshold circuit is a circuit which has two layers of threshold gates: the top gate and bottom gates. We assume that there may be some wire from an input variable to the top gate, and we call such wire a direct wire.

Definition 2.2:
(1) Two gates G_1, G_2 at the bottom level have *dependency*, if $\forall x \in \{0,1\}^n[G_1(x) \leq G_2(x)] \vee \forall x \in \{0,1\}^n[G_2(x) \leq G_1(x)]$. In other words, one of two preimages $G_1^{-1}(1), G_2^{-1}(1)$ is a subset of the other one.
(2) A subset of bottom gates is called *independent gate set*, if any two gates in the set do not have dependency. A circuit may contain several independent gate sets.

Definition 2.3: A depth two threshold circuit C is *sparse*, if $\sum_{G \in \mathcal{B}}$ fan-in of $G \leq dn$, where d is a constant and $\mathcal{B}$ is the set of all bottom level gates in C.

We define an extension of this notion which involves dependency in circuits.

Definition 2.4: A depth two threshold circuit C is *sparse in independent gates*, if there exists some constant d for an arbitrary independent gate sets I in C (at the bottom level) $\sum_{G \in I}$ fan-in of $G \leq dn$. The constant d is called a *sparse constant* of C.

2.1 Problems We Consider

Let's consider the following parameterized circuit SAT problem.

Definition 2.5:
Name of Problem: k-THR-SAT
Given: Depth two threshold circuit C of size n^c which is sparse in independent gates, where c is a constant.
Parameter: k: Maximum size of independent gate sets of C which may depend on the number of input variables n.
Compute: YES iff C is satisfiable.

Definition 2.6:
Name of Problem: k-THR-SAT with unique exception
Given: Depth two threshold circuit C of size n^c which is

sparse in independent gates such that there exists an unique *maximal* independent gate set I' of size greater than k, where c is a constant.
Parameter: k: Maximum size of independent gate sets of C except I'.
Compute: YES iff C is satisfiable.

Note that we obtain an instance of k-THR-SAT by eliminating all gates in I' and wires connecting to them from C.

2.2 Motivation of Our Setting

In this subsection, we describe several facts on circuits with high dependency. We think that these explain why the investigation of threshold circuits parameterized by its dependency is interesting.

Let k-THR be a layer of threshold gates whose maximum independent gate set size is k, and let THR $\circ$ k-THR denote the class of depth two circuits where the top gate is an arbitrary threshold gate and the bottom level is k-THR.

It is clear that the class THR $\circ$ k-THR with $k = 2^n$ can compute all boolean functions by emulating DNF formulas. A bit surprisingly, THR $\circ$ k-THR is universal even for $k = 1$. We describe below the construction of such a circuit for an arbitrary given function.

Let $f(x_{n-1}, \ldots, x_1, x_0)$ be a boolean function on n variables. For simplicity, we assume $f(0, 0, \ldots, 0) = 0$. For $0 \leq j \leq 2^n - 1$, let y_j denote the binary representation of j of length n, i.e., $y_j := (x_{n-1}, \ldots, x_1, x_0)$ with $\sum_{i=0}^{n-1} x_i 2^i$. Let G_j be the threshold gate whose output is 1 iff $\sum_{i=0}^{n-1} 2^i x_i \geq j$. The bottom level of a circuit is consisting of $\mathcal{G} = \{G_j \mid f(y_j) \neq f(y_{j-1})\ (1 \leq y \leq 2^n - 1)\}$. Obviously, there is no pair of independent gates in $\mathcal{G}$. The top gate outputs 1 iff $\sum_{G_j \in \mathcal{G}} w_j G_j \geq 1$ where the weight w_j is $f(y_j) - f(y_{j-1})$ which is 1 or -1. In fact, the value of f is *equal* to $\sum_{G_j \in \mathcal{G}} w_j G_j$. We note that the top gate can be replaced by a symmetric gate (i.e., a gate whose output depends only on the sum of its inputs) that outputs 1 iff $\sum_{G_j \in \mathcal{G}} G_j$ is odd. This says that SYM $\circ$ k-THR is also universal.

We see by these examples that limiting dependency affects not the universality but the complexity, i.e., the number of gates or wires in a circuit. As was described in Introduction, for every $\delta < 1$, the existence of $2^{\delta n}$ time algorithm for k-THR-SAT of superlinear size for $k = \omega(n)$ would refute SETH. It is clear from the definition that the class of functions that can be computed by THR $\circ$ k-THR circuits of size $s(n)$ contains the one computed by THR $\circ$ k'-THR circuits of the same size for every $k' \leq k$. Hence it is interesting to see the largest value of k such that k-THR-SAT admits an algorithm with constant savings as well as to study how the time complexity of circuit satisfiability problem varies as the dependency of input circuits does.

3. Our Results

We show the following main theorem, which is about a construction of nontrivial satisfiability algorithm for depth two threshold circuits with bounded size of independent gate sets.

Theorem 3.1: There is a satisfiability algorithm for k-THR-SAT with unique exception that runs in time $O(2^{(1-s)n})$, where $s = 1/d^{O(d^2)}$, and $k \leq n^\gamma$ for an arbitrary real constant $0 < \gamma < 1$ and d is a sparse constant of a given circuit.

In the rest of the sections, our main goal is to prove the following **Lemma 3.1** and we obtain **Theorem 3.1** from **Lemma 3.1**

Lemma 3.1: There is a randomized satisfiability algorithm for k-THR-SAT with unique exception *in which all random bits are created by independently tossing a coin*, and the algorithm runs in time $O(2^{(1-s)n})$, where $E[s] = 1/d^{O(d^2)}$, and $k \leq n^\gamma$ for an arbitrary real constant $0 < \gamma < 1$ and d is a sparse constant of a given circuit.

In what follows, we give a way to obtain **Theorem 3.1** from **Lemma 3.1**. A way to get a deterministic algorithm from a two sided error algorithm with error probability at most $1/3$ is given as follows. This method is generally called the conditional expectation method.

We consider a randomized algorithm that uses m random bits. We can regard all its sequences of coin tosses as corresponding to a binary tree of depth m. We know that most paths from the root to the leaf are *good*, that is, give a correct answer. It is natural and simple thought to try and find such a path by walking down from the root and making *good* choices at each step. Equivalently, we try to find a good sequence of coin flips with considering each *single bit*.

We consider formally this intuition. Fix a randomized algorithm A and an input x, and let m be the number of random bits used by A on input x. For $1 \leq i \leq m$ and $r_1, r_2, \ldots, r_m \in \{0, 1\}$, we define $P(r_1, \ldots, r_i)$ as the fraction of continuations of a randomized computation that are good sequences of coin tosses. A precise definition is as follows: if $R_1, \ldots, R_m$ are uniform and independent random bits, then $P(r_1, \ldots, r_i)$ is defined as $\Pr_{R_1, \ldots, R_m}[A(x, R_1, \ldots, R_m) \text{ is correct} \mid \bigwedge_{j=1}^{i} \text{“}R_j = r_j\text{”}] = \mathop{E}_{R_{i+1}}[P(r_1, \ldots, r_i, R_{i+1})]$

By averaging argument, there exists an $r_{i+1} \in \{0, 1\}$ such that $P(r_1, \ldots, r_i, r_{i+1}) \geq P(r_1, r_2, \ldots, r_i)$. Thus, at node $(r_1, \ldots, r_i)$, *we pick* r_{i+1} *which maximizes* $P(r_1, \ldots, r_{i+1})$. Finally, we obtain $r_1, \ldots, r_m$ such that $P(r_1, r_2, \ldots, r_m) \geq P(r_1, r_2, \ldots, r_{m-1}) \geq \cdots \geq P(r_1) \geq P \geq 2/3$, where P is the fraction of good path from the root. Therefore, we have $P(r_1, \ldots, r_m) = 1$, because $P(r_1, \ldots, r_m)$ is either 0 or 1.

For an implementation of this argument, we just construct a deterministic algorithm to compute $P(r_1, r_2, \ldots, r_i)$ for each i. Note that if we show an algorithm in which all random bits are created by independently tossing a biased coin then an implementation is given. By the Chernoff bound, we can construct a randomized algorithm which repeats the algorithm in **Lemma 3.1** constant times and runs with error probability at most $1/3$. Thus, using the condi-

tional expectation method, we obtain **Theorem 3.1** by repeating the algorithm in **Lemma 3.1** constant times.

Note that the statement of **Theorem 3.1** improves the following previous result by Impagliazzo et al. [13] in the sense that we can construct a nontrivial algorithm when we relax some condition on sparsity in input circuits.

Theorem 3.2 ([13]): There is a depth two threshold circuit SAT algorithm with n variables and dn wires that runs in time $O(2^{(1-s)n})$, where $s = 1/d^{O(d^2)}$ and d is an arbitrary constant.

We first give a rough and qualitative sketch of the outline of the algorithm in [13]. For given depth two sparse threshold circuits, they give three reductions: the first one transforms an arbitrary ILP instance with small number of inequalities to an instance of vector domination problem, and the second one transforms any depth two circuit with small number of gates to an union of ILP instances with small number of inequalities, and the third one transforms a depth two threshold circuit with linear number of wires to an union of depth two circuits with small number of gates. Constructing an algorithm with constant savings in the exponent, we can test satisfiability of depth two sparse threshold circuits with our setting. Our meaning of *small number* is the one that the number of gates or inequalities is less than the number of variables.

The random restriction technique is used to construct these reduction procedures. The second reduction uses restrictions to output wires of bottom gates in depth two threshold circuits with small number of gates. The third reduction uses restrictions to input variables, and we can decrease the number of bottom gates because of this restriction. For each restriction to output wires of bottom gates we obtain an ILP instance with small number of inequalities by the second reduction, and for each restriction to input variables we obtain a depth two circuit with small number of gates by the third one.

Restricting to the output wire of a bottom gate means obtaining a linear inequality whose variables and coefficients and the threshold value agree with the label of the bottom gate. Let's consider when we obtain a depth two threshold circuit with small number of gates. The number of brute force restrictions to bottom gates of which the number is less than the number of variables is still less than the number of brute force restrictions to input variables. Because of this saving, we can constantly save the complexity of the exponent of the running time.

Restricting variables for the third reduction involves a little technical argument. We take a random subset of variables and assign a boolean string to these variables and let the other unchosen variables remaining. When for a random subset of input variables these variables are fixed, we consider the following two cases for an arbitrary bottom gate. In the first case the gate has at most one unfixed fan-in. In the second case the gate has at least two unfixed input fan-ins. In the former case, such kind of gates do not cause any trouble for the reduction, because we can eliminate these gates and decrease the number of bottom gates. In the latter case, however, we cannot take such straight forward argument. It is the sparsity of circuits that gives the nice property that there are not so many such bad gates.

We mention how we obtain an extension of [13] from our setting. In our setting, the number of restrictions to bottom level gates is bounded above because of dependency of bottom gates. We first define some partial order on the set of bottom gates. Hasse diagrams of this relation are useful to formalize the notion of dependency of bottom gates in a circuit. Next, we define a mid-point problem: for given a pair of a circuit and a Hasse diagram relating with the circuit, output YES if and only if the circuit is satisfiable. Our main subroutine is a randomized algorithm solving this problem. Because of an upper bound on the expected number of restrictions to bottom level gates, the running time of the randomized algorithm is faster than the complexity of the trivial exhaustive search. In other words, the expected exponent of the running time is faster than the one of the trivial exhaustive search. Our main subgoal is to obtain an upper bound on the expected exponent of the running time of a randomized algorithm to check satisfiability. We show several lemmas about the bounds on the number of restrictions. We finally design a randomized algorithm contains several subroutines: the reduction procedure from satisfiability problem for given depth two threshold circuits to the mid-point problem and the algorithm solving an intermediate problem. We obtain a deterministic algorithm by repeating this randomized algorithm constant times, using the method called the conditional expectation method.

4. An Overview of the Entire Algorithm in Lemma 3.1

4.1 Partial Order on Bottom Gates

We express structures on dependency of bottom level gates using directed graphs. We first introduce a partial order representing the dependency of threshold gates.

Definition 4.1: Let C be a depth two threshold circuit. The binary relation $\preceq$ on the set of bottom level gates of C is defined as follows: $G_1 \preceq G_2 \underset{\text{def}}{\iff} G_1^{-1}(1) \subseteq G_2^{-1}(1)$ for all $G_1, G_2 \in \mathcal{G}$, where $\mathcal{G}$ is the set of bottom gates.

We define a problem using the partial order stated in the above.

Definition 4.2:
Name of Problem: L'
Given:$\langle C, H\rangle$ satisfying the following conditions.

- C is an instance of k-THR-SAT with unique exception
- There is *no* pair of gates G_1, G_2 s.t. $G_1^{-1}(1) = G_2^{-1}(1)$
- H is a Hasse diagram of a partial ordered set of bottom level gates in C according to the order $\preceq$.

Output:YES iff C is satisfiable

We will give a procedure for the problem L' and this procedure is a critical subroutine of the algorithm constructed in

Lemma 3.1. The algorithm in **Lemma 3.1** first transforms a given instance of k-THR-SAT to an instance of the problem L', and then solves the satisfiability problem using some structure of Hasse diagram.

The first reduction is given by the following lemma.

Lemma 4.1: There is a reduction which reduces k-THR-SAT with unique exception to the problem L' defined above and the reduction runs in deterministic time $O(poly(n)T_{ZOLP}[2, n])$, where $T_{ZOLP}[m, n]$ is the time complexity of the 0-1 Linear Programming with m constraints and n variables.

Roughly speaking, the reduction described in this lemma generates a Hasse diagram by checking the dependency of every pair of bottom gates by solving an ILP with two constraints. The proof of the lemma is postponed to the next section.

4.2 Restriction to the Bottom Gates and Reduction to ILP

Intuitively, when the dependency of a circuit C is limited, the output of C is determined by fixing the output of a small number of bottom gates. By using this property, we can build a set $\mathcal{S}$ of small number of ILP instances such that C is satisfiable iff at least one instance in $\mathcal{S}$ is feasible and each instance has a small number of constraints. In fact, our algorithm solves k-THR-SAT by solving such set of ILP instances. The following lemma, which will be proved in Sect. 6, states this more formally.

Definition 4.3: For a depth two threshold circuit C, the set $X(C)$ is defined as $\{(y_1, \ldots, y_{n^c}) \in \{0,1\}^{n^c} : y_i$ is the output of the i-th bottom gate in C, when C runs for an arbitrary input $x \in \{0,1\}^n\}$.

Lemma 4.2: Let C be an instance of k-THR-SAT with unique exception. There is a set $\mathcal{S}$ of ILP instances with n variables satisfying the following three conditions: **(1)** It holds that $C^{-1}(1) = \bigcup_{S \in \mathcal{S}} F(S)$, where $F(S)$ is the set of feasible solutions of $S \in \mathcal{S}$, **(2)** the set $\mathcal{S}$ contains at most $|X(C)|$ ILP instances and **(3)** each instance in $\mathcal{S}$ has at most $2k + |I'|$ constraints, where I' is unique exceptional independent gate set in C.

We will use this reduction in the main algorithm which will be described in Sect. 6.

4.3 The Entire Overview

The construction of an algorithm in **Lemma 3.1** is as follows:

1. Call the reduction procedure in **Lemma 4.1** to transform the given instance of k-THR-SAT with unique exception to an instance of the problem L'.
2. Find the exceptional unique independent set I'.
3. Run the main algorithm on the input $\langle C, H\rangle$ and I'.

We note that we can find I' in step 2. as follows. First, let a positive integer l be 1, and repeat increasing l by one until there uniquely exist an independent set of size l. Next, search the unique maximal independent set of size greater than l. Note that the repeating process stops in at most k steps. In step 3, restriction methods to both input variables and outputs of bottom gates and the reduction to ILP in **Lemma 4.2** are used.

5. Partial Order in Circuits and Reduction Lemma

At first we give the following lemma on a binary relation on the set of bottom gates of a depth two threshold circuit mentioned in the previous section.

Lemma 5.1: Assume that C is an instance of k-THR-SAT with unique exception, and that there is *no* two gates G_1, G_2 in C such that $G_1^{-1}(1) = G_2^{-1}(1)$. Then, there exists some partial ordered set $(\mathcal{G}, \preceq)$, where $\mathcal{G}$ is a set of bottom gates of C such that the maximum size of an independent set of Hasse diagram H of $(\mathcal{G}, \preceq)$ is k.

Proof First, we prove the existence of a partial ordered set. For any instance of k-THR-SAT with unique exception the following holds. If there is *no* two equivalent gates at the bottom level in the instance, then there is a partial ordered set $(\mathcal{G}, \preceq)$ such that for all $G_1, G_2 \in \mathcal{G}$, $G_1 \preceq G_2 \iff G_1^{-1}(1) \subseteq G_2^{-1}(1)$, where $\mathcal{G}$ is a set of bottom gates. To prove this statement we show that the relation $\preceq$ is *reflective*, *asymmetric* and *transitive*.

It is clear that $G_1^{-1}(1) \subseteq G_1^{-1}(1)$ and that $G_1^{-1}(1) \subseteq G_2^{-1}(1) \wedge G_2^{-1}(1) \subseteq G_3^{-1}(1) \Rightarrow G_1^{-1}(1) \subseteq G_3^{-1}(1)$ for all gates G_1, G_2, G_3. Thus the relation is reflective and transitive.

Finally, $G_i^{-1}(1) \subseteq G_j^{-1}(1) \wedge G_j^{-1}(1) \subseteq G_i^{-1}(1) \Rightarrow i = j$, because there is *no* pair of two gates which is equivalent. Thus the relation is asymmetric.

Next we argue the maximum size of independent sets in a Hasse diagram. Let $H = (V, E)$ be a Hasse diagram of C stated above. It holds that $G_i^{-1}(1) \neq G_j^{-1}(1)$ if and only if either **(1)** $G_i^{-1}(1) \subsetneq G_j^{-1}(1)$ or **(2)** $G_j^{-1}(1) \subsetneq G_i^{-1}(1)$ or **(3)** $G_i^{-1}(1) \setminus G_j^{-1}(1) \neq \emptyset \wedge G_j^{-1}(1) \setminus G_i^{-1}(1) \neq \emptyset$. Thus, for any bottom gates G, G', the following three conditions are equivalent.
(i) $\neg(G \preceq G') \wedge \neg(G' \preceq G)$.
(ii) $(G, G') \notin E \wedge (G', G) \notin E$.
(iii) G and G' do not have dependency.
Hence, an arbitrary maximum independent set in H corresponds to some maximum independent gate set in C by the definition of $\preceq$. □

In the rest of this section, we give the proof of the lemma describing the reduction from k-THR-SAT to the problem L'.

Lemma 4.1(restated) There is a reduction which reduces k-THR-SAT with unique exception to the problem L' defined above and the reduction runs in deterministic time $O(poly(n)T_{ZOLP}[2, n])$, where $T_{ZOLP}[m, n]$ is the time complexity of the 0-1 Linear Programming with m constraints

and n variables.

Proof Let $P(x), Q(x)$ be constraints depending on $x = (x_1, \ldots, x_n) \in \{0,1\}^n$. Note that $\forall x[P(x) \Rightarrow Q(x)]$ is equivalent to $\forall x[\neg P(x) \vee Q(x)]$ and is equivalent to $\neg[\exists x[P(x) \wedge \neg Q(x)]]$ and that $\exists x[P(x) \wedge \neg Q(x)]$ is the YES-condition of the Integer Linear Programming. Thus we have the following procedure for the reduction. In this procedure, the 0-1 Linear Programming with two constraints and n variables is solved in step 4. and the other steps are computed in polynomial time. Thus, the running time of this procedure is $O(poly(n)T_{ZOLP}[2,n])$. □

The reduction procedure

1. Let $V = \{G_1, \ldots, G_{n^c}\}$ be a set of bottom gates and let D be $\emptyset$.
2. For all pairs of bottom level gates G, G' do the following steps 3.,4.,5.
   3. Let the label of G and G' be $\sum_{i \in S} a_i x_i \geq b$ and $\sum_{i \in S'} a'_i x_i \geq b'$ respectively, where S, S' are the sets of indices of variables connecting to G, G' respectively.
   4. Solve the following instances of Integer 0-1 Linear Programming with *two* constraints, that is, (1) $\sum_{i \in S} a_i x_i \geq b$ and (2) $\sum_{i \in S'} a'_i x_i < b'$
   5. If there does *not* exist $(x_1, \ldots, x_n) \in \{0,1\}^n$ satisfying the constraints (1) and (2) then $D := D \cup \{(G, G')\}$(because of $G^{-1}(1) \subseteq G'^{-1}(1)$)
6. For all G, G' such that $(G, G') \in D \wedge (G', G) \in D$ do the following steps 7., 8.
   7. for each bottom gate U let y_U be the output wire of U. The label $w_G y_G + w_{G'} y_{G'} + \sum_{U \neq G, G'} w_U y_U \geq t_{TOP}$ in the TOP gate is replaced with $(w_G + w_{G'}) y_G + \sum_{U \neq G, G'} w_U y_U \geq t_{TOP}$
   8. G' and all input and output wires of G' are removed from C. Replace V with the set of bottom gates in C. For any bottom gate U, if $(U, G') \in D \vee (G', U) \in D$ then $D := D \setminus \{(U, G'), (G', U)\}$.
9. Output the result $\langle C, H = (V, D)\rangle$

6. Main Algorithm

At first we recap a sketch of the outline of the algorithm in [13]. Remind three reductions for given depth two sparse threshold circuits: the first one transforms an arbitrary ILP instance with small number of inequalities to an instance of vector domination problem, and the second one transforms any depth two circuit with small number of gates to an union of ILP instances with small number of inequalities, and the third one transforms given instance to an union of depth two circuits with small number of gates.

We show several lemmas about the bounds of the number of restrictions. We first define several terms being necessary for formal statements of these lemmas.

Definition 6.1: A directed graph $H = (V, E)$ is an Induced Hasse Diagram (abbreviated I.H.D) of a circuit C which is an instance of k-THR-SAT with unique exception, if H is the output $\langle C, H\rangle$ of the procedure in the proof of **Lemma 4.1**.

Definition 6.2:
Validly Ordered Restriction: Let C be an instance of k-THR-SAT and let $H = (V, E)$. A coloring $\chi : V \mapsto \{0,1\}$ is called a *validly ordered restriction* (abbreviated V.O.R), if $\forall (u, v) \in E, \chi(u) \leq \chi(v)$. For a set $I \subset V$, $\chi(I)$ denotes the set $\{\chi(v) \mid v \in I\}$ as usual.
Min-Set, Max-Set for a V.O.R: Let χ be a V.O.R for an arbitrary I.H.D $H = (V, E)$.

S is a min-set of H for V.O.R χ, if $S = \{u_{min} \in V \cap \chi^{-1}(1) : \forall v \in V \setminus \{u_{min}\}[v \preceq u_{min} \Rightarrow \chi(v) = 0]\}$

S is a max-set of H for V.O.R χ, if $S = \{u_{max} \in V \cap \chi^{-1}(0) : \forall v \in V \setminus \{u_{max}\}[u_{max} \preceq v \Rightarrow \chi(u) = 1]\}$
The covering condition: Let H be an I.H.D and χ be a validly ordered restriction of H. Let I_1, I_0 be independent sets in H.

The pair of independent sets (I_1, I_0) satisfies the *covering condition* for H, if the following condition holds.
Condition: For any $v \in V \setminus (I_1 \cup I_0)$ in H, either $\exists u_1 \in I_1, u_1 \preceq v$ or $\exists u_0 \in I_0, v \preceq u_0$ according to the order $\preceq$ of H.

We count the number of validly ordered restrictions, and a lemma in this section is about an upper bound on the number of these restrictions. Bottom gates in min-set or max-set are critical to design our algorithm for satisfiability. Satisfiability of a circuit depends on information about bottom gates which are in min-set or max-set of the circuit, when output of bottom gates are fixed. In other words, we can decide satisfiability, even if we consider only some local information about bottom gates of a circuit and ignore the other gates. The covering condition is a condition stating the concept of min-set and max-set from another viewpoint, and is used in our algorithm in this section.

Definition 6.3: Let X'_H be a set of validly ordered restrictions of H. We define $\mathcal{I}_H$ as a set of pairs of independent sets in H which satisfies the covering condition, that is, $\mathcal{I}_H := \{(I_1, I_0) \subseteq V \times V : I_1, I_0$ are independent sets satisfying the covering condition in $H\}$.

The following lemma is a main lemma of this section, and rough meaning of this lemma is that we can construct a satisfiability algorithm using exhaustive search for all independent gate sets.

Definition 4.3(restated) For a depth two threshold circuit C, the set $X(C)$ is defined as $\{(y_1, \ldots, y_{n^c}) \in \{0,1\}^{n^c} : y_i$ is the output of the i-th bottom gate in C, when C runs for an arbitrary input $x \in \{0,1\}^n\}$.

Lemma 6.1: For an arbitrary instance $\langle C, H\rangle \in L'$, let I' be

the unique maximal independent set of size greater than k. Then, $|X(C)| \leq 2^{|I'|}k^2n^{O(k)}$.

We first show the following lemma to prove **Lemma 6.1**, which reduces counting the number of restrictions for a circuit to counting the number of structures in a graph.

Lemma 6.2: There is a bijection $\mu_H : X'_H \ni \chi \mapsto (I_1, I_0) \in \mathcal{I}_H$ such that if $\mu_H(\chi) = (I_1, I_0)$ then I_1 is the min-set of H for χ and I_0 is the max-set of H for χ.

First, we show two claims. The lemma easily follows from these claims.

Claim 6.1: Let H be an I.H.D and χ be a validly ordered restriction of H. Let I_1, I_0 be min-set and max-set of H for χ respectively. Then (I_1, I_0) is a pair of independent sets such that the covering condition holds for H.

Proof We give a proof by contradiction.

For any fixed χ which assign 0 or 1 to vertices of H and for min-set I_0 and max-set I_1 of H for χ, adding all edges which is in $I_0 \times I_1$ preserves validity of χ. In other words, after adding all edges which is in $I_0 \times I_1$, χ is still a valid ordered restriction. Let $H' = (V, E')$ be this I.H.D which is *obtained by adding edges to H*. Thus I_0, I_1 are maximal independent sets in H'.

Assume that I_0, I_1 do not satisfy the covering condition. Then either **case1** or **case2** holds for H'.

case1. $\exists u_1 \in I_1, u_1 \preceq v$ and $\exists u_0 \in I_0, v \preceq u_0$, for some $v \in V \setminus (I_1 \cup I_0)$ in H'.

In this case, $u_1 \preceq u_0$ contradicts to the assumption that $u_1 \in I_1, u_0 \in I_0$.

case2. $\forall u_1 \in I_1, \neg(u_1 \preceq v)$ and $\forall u_0 \in I_0, \neg(v \preceq u_0)$, for some $v \in V \setminus (I_1 \cup I_0)$ in H'.

In this case, since I_1, I_0 are maximal independent sets, for all $v' \in V \setminus (I_1 \cup I_0)$ it holds that $\forall u_1 \in I_1, \neg(u_1 \preceq v') \Rightarrow v' \preceq u_1$ and $\forall u_0 \in I_0, \neg(v' \preceq u_0) \Rightarrow u_0 \preceq v'$. Thus we obtain that there exists a vertex $v \in V \setminus (I_0 \cup I_1)$ such that $u_0 \preceq v \preceq u_1$, contradicting to $u_1 \in I_1$ and $u_0 \in I_0$. In other words, for any $c \in \{0, 1\}$ we obtain that $\chi(v) = c$ contradicts to $u_c \in I_c$. □

Claim 6.2: For any (I_1, I_0) which is a pair of independent sets in H satisfying the covering condition, there uniquely exists validly ordered restriction $\chi : V \to \{0, 1\}$ such that I_1, I_0 are min-set and max-set of H for χ, respectively.

Proof For any validly ordered restriction χ it holds that I_1 is max-set implies $\chi(I_1) = \{1\}$, and I_0 is min-set implies $\chi(I_0) = \{0\}$. First, we consider vertices in $I_0 \cup I_1$. Since I_1, I_0 are min-set and max-set of H for χ respectively, assign zero-one value to vertices in $I_1 \cup I_0$ such that $\chi(I_1) = \{1\}$ and $\chi(I_0) = \{0\}$. Finally we consider the other vertices. For any $v \in V \setminus (I_1 \cup I_0)$, the value $\chi(v)$ is uniquely determined by the definition of covering condition. □

Proof of Lemma 6.2 For any $\chi \in X'_H$, there uniquely exists $(I_1, I_0) \subseteq V \times V$ such that I_1 is the min-set of H for χ and I_0 is the max-set of H for χ. By **Claim 6.1**, there is a map $\mu_H : X'_H \to \mathcal{I}_H, \mu_H(\chi) = (I_0, I_1)$ such that if $\mu_H(\chi) = (I_1, I_0)$ then I_1 is the min-set of H for χ and I_0 is the max-set of H for χ. By **Claim 6.2**, this map μ_H is a bijective map. □

Proof of Lemma 6.1 Fix an assignment $\chi|_{I'} : I' \to \{0, 1\}$. Let $H = (V, E)$.

We assign 1 to any vertex $u \in V$ such that there is some vertex $v \in \chi|_{I'}^{-1}(1) \subseteq I'$ such that $v \preceq u$. We assign 0 to any vertex $u \in V$ such that there is some vertex $v \in \chi|_{I'}^{-1}(0) \subseteq I'$ such that $u \preceq v$.

Remove all vertices *whose assignment is fixed* and all edges connecting to them. Thus we obtain a subgraph H' of H such that size of any independent vertex set in H' is at most k because we assume the existence of unique exception. The output pattern set $X(C)$ is a subset of validly ordered restrictions of H.

By **Lemma 6.2** there is a bijective map $\mu_{H'} : X'_{H'} \to \mathcal{I}_{H'}$. Thus for any H' the cardinality $|X'_{H'}|$ is bounded above by $|\mathcal{I}_{H'}|$. Since $|X(C)| \leq 2^{|I'|} \max_{H'} |X'_{H'}|$ and $|\mathcal{I}_{H'}| \leq \left(\sum_{i=1}^{k} \binom{n^c}{i}\right)^2 \leq \left(\sum_{i=1}^{k} n^{ci}\right)^2 \leq \left(kn^{ck}\right)^2 = k^2n^{O(k)}$, we obtain the desired bound: $|X(C)| \leq 2^{|I'|}k^2n^{O(k)}$. □

The following lemma is similar to a part of work in [13], which gives a reduction from an instance of circuit satisfiability to a union of ILPs.

Lemma 4.2(restated) Let C be an instance of k-THR-SAT with unique exception. There is a set $\mathcal{S}$ of ILP instances with n variables satisfying the following three conditions: **(1)** It holds that $C^{-1}(1) = \bigcup_{S \in \mathcal{S}} F(S)$, where $F(S)$ is the set of feasible solutions of $S \in \mathcal{S}$, **(2)** the set $\mathcal{S}$ contains at most $|X(C)|$ ILP instances and **(3)** each instance in $\mathcal{S}$ has at most $2k + |I'|$ constraints, where I' is unique exceptional independent gate set in C.

Proof For any circuit C and each element in $X(C)$, we obtain the following transformation from a circuit to an ILP instance, according to the following three kinds of gates.

(i) For gates whose output is fixed to 1 with weights $w_1, \ldots, w_n$ and threshold t, we have $\sum_{i=1}^{n} w_i x_i \geq t$.

(ii) For gates whose output gate is fixed to 0 we require $\sum_{i=1}^{n} w_i x_i < t$, which is equivalent to $\sum_{i=1}^{n} -w_i x_i \geq -t + \min_i w_i$.

(iii) For the top gate, let $v_1, \ldots, v_n$ be the weights of the direct wires, and s be the threshold of the top level gate, and w_{FIX} be the sum of the weights of the gates whose output is fixed to 1. Then we require $\sum_{i=1}^{n} v_i x_i \geq s - w_{FIX}$.

Thus, the set of these instances satisfies the conditions **(1)** and **(2)**. Moreover, the following **Fact 6.1** and the definition of min-set and max-set implies that the dependency of gates in C gives at most $2k + |I'|$ constraints, and yields the existence of a set $\mathcal{S}$ of ILP instances with n variables satisfying the condition **(3)**, because it holds that for each restriction to gates in I', the circuit C' which is obtained by removing all gates in I' and all wires connecting to them

from C has min-set and max-set of size at most k. □

Fact 6.1: For boolean functions $P_1(x), \ldots, P_m(x) : \{0,1\}^n \to \{0,1\}$, the following statements hold.
(1) If $\forall x \in \{0,1\}^n, P_1(x) = 1 \Rightarrow P_2(x) = 1 \Rightarrow \cdots \Rightarrow P_m(x) = 1$ then, it holds that $\forall x \in \{0,1\}^n, P_1(x) = 1 \wedge P_2(x) = 1 \wedge \cdots \wedge P_m(x) = 1 \iff \forall x \in \{0,1\}^n, P_1(x) = 1$.
(2) If $\forall x \in \{0,1\}^n, P_m(x) = 0 \Rightarrow P_{m-1}(x) = 0 \Rightarrow \cdots \Rightarrow P_1(x) = 0$ then, it holds that $\forall x \in \{0,1\}^n, P_m(x) = 0 \wedge P_{m-1}(x) = 0 \wedge \cdots \wedge P_1(x) = 0 \iff \forall x \in \{0,1\}^n, P_m(x) = 0$.

When for a random subset of input variables these variables are fixed, we consider the following two cases for an arbitrary bottom gate. The gate has at most one unfixed input wire in one case, and the gate has at least two unfixed input wires in the other case. In the former case, such gates are not harmful for our argument about the reduction, because we can eliminate these gates and decrease the number of bottom gates. In the latter case, however, we cannot use such straightforward argument. We define more precisely such gates to which we cannot directly apply gate elimination argument.

Definition 6.4: BAD gate
Let U be a subset of variables. BAD gate on U is a bottom level gate that depends on at least two variables in U.

In other words, if a gate is *not* BAD then we can eliminate it or replace it with a direct wire.

Lemma 6.3 ([13]): Consider a depth two threshold circuit with n variables and dn wires. Let $\delta > 0$ be an arbitrary positive real number and let $\tilde{U}$ be a random set of variables such that each variable is not in $\tilde{U}$ with some probability p independently. There exists a $p = 1/d^{O(d^2)}$ such that *the expected number of BAD gates on* $\tilde{U}$ is at most $3\delta pn$, where d is a sparse constant.

The following corollary is easily obtained from this lemma.

Corollary 6.1 ([13]): Consider a depth two threshold circuit C with n variables, which is a part of $\langle C, H\rangle \in L'$. Let $VAR[I']$ be a set of variables connecting to the gates corresponding to I', which is unique exceptional independent gate set in C. Let $\delta > 0$ be an arbitrary positive real number and let $\tilde{U}$ be a random set of variables such that each variable is not in $\tilde{U}$ with some probability p independently. Let $I'_{\tilde{U}}$ be a subset of I' which are also BAD gates on $\tilde{U}$. There exists a $p = 1/d^{O(d^2)}$ such that $E[|I'_{\tilde{U}}|] \le 3\delta p|VAR[I']|$, where d is a sparse constant.

Let $C|_{\rho[\tilde{U}]}$ be a circuit obtained by the operation for a circuit C that all variables in $\tilde{U}$ is fixed to an arbitrary assignment $\rho[\tilde{U}] : \tilde{U} \to \{0,1\}$ and any gate, which is not BAD, is eliminated from C or replaced with direct wires in C.

Corollary 6.2: Consider a depth two threshold circuit with n variables, which is a part of $\langle C, H\rangle \in L'$. Let $\delta > 0$ be an arbitrary positive real number and let $\tilde{U}$ be a random set of variables such that each variable is not in $\tilde{U}$ with some probability $p = 1/d^{O(d^2)}$, where d is a sparse constant.

$E[\log |X(C|_{\rho[\tilde{U}]})|] \le 3\delta pn + O(k \log_2 n + \log_2 k)$, for any assignment $\rho[\tilde{U}] : \tilde{U} \to \{0,1\}$.

Proof By **Lemma 6.3**, we obtain that $\log |X(C|_{\rho[\tilde{U}]})| \le |I_{\tilde{U}'}| + O(k \log_2 n + \log_2 k)$. Hence **Corollary 6.1** and linearity of expectation give us the desired bound. □

Finally we consider an algorithm to solve L' under given the unique exception of each instance. Let $VAR[I']$ be a set of variables connecting to the gates corresponding to I'.

We give a description of the main algorithm as follows, using for all above ingredients. Note that all random bits are created by tossing a coin independently in the following algorithm.

Description of the main algorithm
Given: An instance $\langle C, H\rangle$, and the exceptional unique independent set I' in H. **Output:**YES if and only if $\langle C, H\rangle$ is a YES instance of L'.

1. Choose a random subset $\tilde{U} \subset VAR[I']$ such that each variable is *not* in $\tilde{U}$ with probability p independently.
2. For each boolean assignment to $\tilde{U}$, fix the value of input variables in $\tilde{U}$, and do the following steps 3. and 4.
    3. Eliminate any gate in I' whose output is totally fixed. Replace any gate whose output value is the value of some input variable x with direct wire connecting to x. Let V_D be a set of variable indices directly connecting to the top gate. Let $D_i (i \in V_D)$ be the sum of weights at the top gate such that these weights are coefficients of outputs of bottom gates replaced with direct wires connecting to the i-th variable.
    4. For each restriction μ to outputs of *remaining* bottom gates in I' do the following steps 5., 6., and 7.
        5. Assign 0 to the output of an *arbitrary bottom* gate G, if there is some $G' \in I'$ such that the output of G' is fixed to 0 and $G \preceq G'$. Assign 1 to the output of an *arbitrary bottom* gate G, if there is some $G' \in I'$ such that the output of G' is fixed to 1 and $G' \preceq G$.
        6. Remove all gates whose outputs are totally fixed from the given circuit which is a part of given instance of L'. Let H' be an I.H.D. which is obtained from H by this removing operation.
        7. Find k as follows. Let l be 1. Repeat increasing l by one until there uniquely exist an independent set of size l. Let k be l. For each pair of gate sets (I_0, I_1) in H' such that $|I_0|, |I_1| \le k$, if (I_0, I_1) is a pair of *independent gate sets* in H' and satisfies *the covering condition*, then solve ILP for an instance

which is obtained from the top gate and bottom gates in $I_0 \cup I_1 \cup I'$ and is constituted by the following three kinds of inequalities (i), (ii), and (iii). If satisfying 0-1 vector is found then HALT and return "YES".

(i) For bottom gates whose output is fixed to 1 with weights $w_1, \ldots, w_n$ and threshold t, the corresponding inequality is $\sum_{i=1}^{n} w_i x_i \geq t$.

(ii) For bottom gates whose output gate is fixed to 0, the corresponding inequality is $\sum_{i=1}^{n} -w_i x_i \geq -t + \min_i w_i$.

(iii) For the top gate, let $v_1, \ldots, v_n$ be the weights of the direct wires, and s be the threshold of the top level gate, and w_{FIX} be the sum of the weights of the gates whose output is fixed to 1. Then for the top gate, the corresponding inequality is $\sum_{i=1}^{n} v_i x_i \geq s - w_{FIX}$.

8. HALT and return "NO"

Note that three kinds of inequalities in this step. appear in the proof of **Lemma 4.2** and that the iterating increment of l in this step stops in at most k steps. Note also that **Lemma 6.1** which gives an upper bound on the number of restrictions in steps 2. and 4. is a key to obtain a constant saving in the exponent in the running time of our algorithm.

7. Analysis of the Expected Savings

In this section our goal is the following lemma about the expected savings.

Lemma 3.1(restated) There is a randomized satisfiability algorithm for k-THR-SAT with unique exception *in which all random bits are created by independently tossing a coin*, and the algorithm runs in time $O(2^{(1-s)n})$, where $E[s] = 1/d^{O(d^2)}$, and $k \leq n^{\gamma}$ for an arbitrary real constant $0 < \gamma < 1$ and d is a sparse constant of a given circuit.

We remind the construction of an algorithm in **Lemma 3.1** as follows.
1. Call the reduction procedure in **Lemma 4.1** to transform the given instance of k-THR-SAT with unique exception to an instance of the problem L'.
2. Find the exceptional unique independent set I'.
3. Run the main algorithm on the input $\langle C, H \rangle$ and I'.

Let $T_{\tilde{U}}(n)$ be the running time of the Main Algorithm. To consider savings of this algorithm, we only analyze the exponent $\log T_{\tilde{U}}(n)$ of the running time of the Main Algorithm, because the time complexity of entire procedure is at most $3\max\{2^{\varepsilon n}, 2^{o(n)}, T_{\tilde{U}}(n)\}$ for some positive constant $\varepsilon < 1$. Note that three quantities $2^{\varepsilon n}$, $2^{o(n)}$, and $T_{\tilde{U}}(n)$ are respectively corresponding to three steps **1.**, **2.**, and **3.** in the above algorithm and that by summing up these three terms we obtain the bound. We also note that we can find I' in step 2. by the exhaustive search for all i $(1 \leq i \leq k+1)$ and for all i-sets of the vertex set of H and searching maximal independent set of size greater than k. We use the following result.

Corollary 7.1 ([13]): Consider a 0-1 Integer Linear Program on n variables and $m(n)$ inequalities Let λ be $m(n)/n$. Then we can find a solution in time.

$$2^{n/2}\left(\binom{(1/2+\lambda)n}{\lambda n} poly(n)\right) \leq 2^{(1/2+\lambda(\log(e)+\log(1+1/2\lambda)))n} \cdot poly(n).$$

Note that this algorithm is faster than 2^n for $\lambda < 0.136$ and has some positive constant saving C' such that running time is $2^{(1-C')n}$.

Firstly we prove **Lemma 3.1** assuming the following Claim.

Claim 7.1: There exists a constant N_0 for any $\delta > 0$, $E[\log T_{ZOLP}[2k + |I'_{\tilde{U}}|, |R|]] \leq 0.5pn + 3N_0\delta pn + o(n)$, where R is a set of remaining variables in the main algorithm.

Proof of Lemma 3.1 Let R be a set of remaining variables in the main algorithm. The expectation of the exponent of the time complexity is bounded above as follows.

$E[\log T_{\tilde{U}}] \leq E[\log(|\{0,1\}^{|\tilde{U}|}| \cdot |X(C_{\rho[\tilde{U}]})| \cdot T_{ZOLP}[2k + |I'_{\tilde{U}}|, |R|] \cdot poly(|R|))]$, where $|\{0,1\}^{|\tilde{U}|}|$ is the number of assignments for brute force restriction to a random subset of variables $\tilde{U}$. We mention how each operation in the main algorithm contributes to the above expectation. We remind that there are two loops in the description of the main algorithm: the inner loop and the outer loop. Note that the term $|X(C_{\rho[\tilde{U}]})|$ corresponds to restricting procedure to bottom gates in the steps 4., 5., and 6. and that $T_{ZOLP}[2k + |I'_{\tilde{U}}|, |R|]$ corresponds to solving ILP problem in the step 7., where $|I'_{\tilde{U}}|$ is the size of the unique exception when a random subset of input variables is chosen and input variables in the subset are fixed. Thus by all these observations we obtain the above bound.

By the linearity of expectation, $E[\log |\{0,1\}^{|\tilde{U}|}|] + \log |X(C_{\rho[\tilde{U}]})| + \log T_{ZOLP}[2k + |I'_{\tilde{U}}|, |R|]] = E[\log |\{0,1\}^{|\tilde{U}|}|] + E[\log |X(C_{\rho[\tilde{U}]})|] + E[\log T_{ZOLP}[2k + |I'_{\tilde{U}}|, |R|]]$.

We show upper bounds on the above three terms. Firstly, note that $E[|\tilde{U}|] = (1-p)n$. By **Corollary 6.2**, we obtain the following upper bound. $E[\log |X(C|_{\rho[\tilde{U}]})|] \leq 3\delta pn + O(k \log_2 n + \log_2 k)$. By **Claim 7.1**, we obtain the following bound. $E[T_{ZOLP}[2k + |I'_{\tilde{U}}|, |R|])] \leq 0.5pn + 3N_0\delta pn + o(n)$.

By summing up these three bounds, we obtain $E[\log |\{0,1\}^{|\tilde{U}|}|] + E[\log |X(C_{\rho[\tilde{U}]})|] + E[\log T_{ZOLP}[2k + |I'_{\tilde{U}}|, |R|]] \leq (1-p)n + 0.5pn + 3(N_0+1)\delta pn + o(n)$. There is some δ, which is a sufficient small constant such that for some positive constant C'' we obtain $E[\log |X(C_{\rho[\tilde{U}]})|] + E[\log T_{ZOLP}[2k + |I'_{\tilde{U}}|, |R|]] \leq (1 - (C'' - o(1)))pn$. Therefore, we obtain a bound $E[\log T_{\tilde{U}}] \leq (1-p)n + (1 - (C'' - o(1)))pn + O(\log n) = (1 - (C'' - o(1))p)n$. The lemma follows from $p = 1/d^{O(d^2)}$. □

Finally, we give a proof of **Claim 7.1** and completes the entire proof of **Lemma 3.1**.

Proof of Claim 7.1 We will use **Corollary 7.1** to obtain the desired upper bound. Note that $|R|$ is the number of remaining variables and that $|I'_{\tilde{U}}| + 2k$ is the number of constraints. Recall that λ in **Corollary 7.1** is defined as the number of constraints divided by the number of variables. Thus $\lambda = \frac{|I'_{\tilde{U}}|+2k}{|R|}$.

Firstly, we consider the following two cases; (i)$\lambda \geq 1/2$, and (ii) $\lambda < 1/2$.

Let's consider the case (i) $\lambda \geq 1/2$. In this case, $1 \geq 1/2\lambda$ and then $\log(1 + 1/2\lambda) \leq 1$.

By **Corollary 7.1**, $\log T_{ZOLP}[|I'_{\tilde{U}}| + 2k, |R|]$ is $0.5|R| + \lambda(\log(e)+1)|R| \leq 0.5|R| + \frac{|I'_{\tilde{U}}|+2k}{|R|}(\log(e)+1)|R|$. Since **Corollary 6.1** implies $E[|I'_{\tilde{U}}|] \leq 3\delta pn$, it holds that

$$\begin{aligned} &E[\log T_{ZOLP}[|I'_{\tilde{U}}| + 2k, |R|]] \\ &\quad \leq 0.5E[|R|] + (\log(e) + 1)E[|I'_{\tilde{U}}|] + o(n) \\ &\quad \leq 0.5pn + (\log(e) + 1)3\delta pn + o(n). \end{aligned}$$

Let's consider the case (ii) $\lambda < 1/2$. In this case it holds that $|I'_{\tilde{U}}| \leq \frac{1}{2}|R| - 2k$ and $\log(1 + 1/2\lambda) \leq \log(1/\lambda)$. Therefore by **Corollary 7.1**,

$$\begin{aligned} &\log T_{ZOLP}[|I'_{\tilde{U}}| + 2k, |R|] \\ &\leq 0.5|R| + \lambda(\log(e) + \log(1/\lambda))|R| \\ &\leq 0.5|R| + \frac{|I'_{\tilde{U}}| + 2k}{|R|}(\log(e) - \log(|I'_{\tilde{U}}| + 2k) + \log|R|)|R| \end{aligned}$$

Note that $-\log\lambda = -\log\frac{|I'_{\tilde{U}}|+2k}{|R|} = -\log(|I'_{\tilde{U}}| + 2k) + \log|R|$.

In this case, we consider the following two subcases; (ii-a) for all real constant $\beta > 0$, $|I'_{\tilde{U}}| + 2k \leq \beta|R|$, and (ii-b) there exists some real constant β_0 $(0 < \beta_0 < \frac{1}{2})$, $|I'_{\tilde{U}}| + 2k > \beta_0|R|$.

(ii-a) for all real constant β $(0 < \beta < \frac{1}{2})$, $|I'_{\tilde{U}}| + 2k \leq \beta|R|$.

Then, $\log T_{ZOLP}[|I'_{\tilde{U}}| + 2k, |R|] \leq \log T_{ZOLP}[\beta|R|, |R|]$. Thus, we have

$$\begin{aligned} &E[\log T_{ZOLP}[|I'_{\tilde{U}}| + 2k, |R|]] \\ &\leq E[\log T_{ZOLP}[\beta|R|, |R|]] \\ &\leq (1/2 + \beta(\log(e) + \log(1 + 1/2\beta)))E[|R|] \\ &= (0.5 + \beta(\log(e) + \log(1 + 1/2\beta)))pn. \end{aligned}$$

(ii-b) there exists some real constant β_0 $(0 < \beta_0 < \frac{1}{2})$, $|I'_{\tilde{U}}| + 2k > \beta_0|R|$.

In this case, $\beta_0|R| < |I'_{\tilde{U}}| + 2k < \frac{1}{2}|R|$. Thus, there exists some real constant α $(\beta_0 < \alpha < \frac{1}{2})$ such that $|I'_{\tilde{U}}| + 2k = \alpha|R|$.

Hence $0.5|R| + \frac{|I'_{\tilde{U}}|+2k}{|R|}(\log(e) - \log(|I'_{\tilde{U}}|+2k) + \log|R|)|R| = 0.5|R| + (|I'_{\tilde{U}}| + 2k)(\log(e) - \log(\alpha|R|) + \log|R|) = 0.5|R| + (|I'_{\tilde{U}}| + 2k)(\log(e) + \log(\alpha^{-1}))$. Remind that $|I'_{\tilde{U}}| \leq \frac{1}{2}|R| - 2k$ and $\log(1 + 1/2\lambda) \leq \log(1/\lambda)$.

Therefore,

$$\begin{aligned} &E[\log T_{ZOLP}[|I'_{\tilde{U}}| + 2k, |R|]] \\ &= 0.5E[|R|] + (\log(e) + \log(\alpha^{-1}))E[|I'_{\tilde{U}}|] + 2k(\log(e) + \log(\alpha^{-1})) \\ &\leq 0.5pn + (\log(e) + \log(\alpha^{-1}))(3\delta p)n + 2k(\log(e) + \log(\alpha^{-1})) \end{aligned}$$

In the case (ii-a), for any $\delta > 0$, let β be a constant such that $\beta(\log(e) + \log(1 + 1/2\beta)) \leq 3\delta$. Thus, in the case (ii-a), $T_{ZOLP}[|I'_{\tilde{U}}| + 2k, |R|]] \leq 0.5pn + 3\delta pn + o(n)$.

Let N_0 be $\max\{\log(e) + 1, \log(e) + \log(\alpha^{-1})\}$. Note that N_0 does not depend on δ because α does not depend on δ. Therefore, there exists some constant N_0 for any $\delta > 0$ it holds that $E[\log T_{ZOLP}[|I'_{\tilde{U}}| + 2k, |R|]] \leq 0.5pn + 3N_0\delta pn + o(n)$. □

8. Concluding Remark

In this section, we mention future work: finding nonuniform circuit classes having super polynomial size lower bounds against NEXP. We denote $O(k)$-THR as a layer of threshold circuits such that the size of maximum independent gate sets is at most $O(k)$, where k may depend on the number of input variables.

Let $\mathsf{SYM} \circ \underbrace{O(k)\text{-THR} \circ \cdots \circ O(k)\text{-THR}}_{d}$ be a class of circuits which have the symmetric gate at the top that connects to d layers of threshold gates defined in the above. Let $\mathsf{ACC} \circ \mathsf{THR} \circ \underbrace{O(k)\text{-THR} \circ \cdots \circ O(k)\text{-THR}}_{d}$ be a class of circuits which have $\mathsf{ACC} \circ \mathsf{THR}$ circuit at the top and also have d layers of threshold gates defined in the above. It is clear that this class is larger than $\mathsf{ACC} \circ \mathsf{THR}$ for which super polynomial size lower bounds against NEXP are proved [22]. Proving size lower bounds for these circuit classes against NEXP by extending the methods developed in this paper is an interesting future work.

References

[1] A. Biere, A. Cimatti, E. Clarke, and Y. Zhu, "Symbolic modelchecking without BDDs," in Tools and Algorithms for the Construction and Analysis of Systems, pp.193–207, 1999.

[2] R.B. Boppana and M. Sipser, "The complexity of finite functions," in Handbook of theoretical computer science (vol.a), pp.757–804, MIT Press, Cambridge, MA, USA, 1990.

[3] C. Calabro, "The exponential complexity of satisfiability problems," PhD thesis, University of California, San Diego, 2009.

[4] C. Calabro, R. Impagliazzo, and R. Paturi, "The complexity of satisfiability of small depth circuits," Parameterized and Exact Computation: 4th International Workshop, IWPEC 2009, Copenhagen, Denmark, Sept. 2009, Revised Selected Papers, pp.75–85, Springer-Verlag, Berlin, Heidelberg, 2009.

[5] A.K. Chandra, L. Stockmeyer, and U. Vishkin, "Constant depth reducibility," SIAM J. on Comput, vol.13, no.2, pp.423–439, 1984.

[6] S.A. Cook, "The complexity of theorem-proving procedures," Proceedings of the Third Annual ACM Symposium on Theory of Computing, pp.151–158, 1971.

[7] E. Dantsin and A. Wolpert, "Max-SAT for formulas with constant clause density can be solved faster than in $O(2^n)$ time," in Armin Biere and CarlaP. Gomes, editors, Theory and Applications of Satisfiability Testing - SAT 2006, Lecture Notes in Computer Science, vol.4121, pp.266–276, Springer Berlin Heidelberg, 2006.

[8] A. Hajnal, W. Maass, P. Pudlak, M. Szegedy, and G. Turan, "Threshold circuits of bounded depth," Proceedings of the 28th Annual Symposium on Foundations of Computer Science, FOCS'87, pp.99–110, IEEE Computer Society, Washington, DC, USA, 1987.
[9] 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, pp.237–240, 1999.
[10] R. Impagliazzo, R. Paturi, and F. Zane, "Which problems have strongly exponential complexity?," J. 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.
[11] R. Impagliazzo, W. Matthews, and R. Paturi, "A satisfiability algorithm for AC0," Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms, pp.961–972, 2012.
[12] 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.
[13] R. Impagliazzo, R. Paturi, and S. Schneider, "A satisfiability algorithm for sparse depth two threshold circuits," FOCS'13, pp.479–488, 2013.
[14] L. Levin, "Universal sorting problems," Problems of Information Transmission, vol.9, pp.265–266, 1973.
[15] I. Lynce and J. Marques-Silva, "Efficient haplotype inference with Boolean satisfiability," National Conference on Artificial Intelligence, pp.104–109, July 2006.
[16] R. Santhanam, "Fighting perebor: New and improved algorithms for formula and qbf satisfiability," Proc. 2010 IEEE 51st Annual Symposium on Foundations of Computer Science, FOCS10, pp.183–192, IEEE Computer Society, Washington, DC, USA, 2010.
[17] 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.
[18] A. Smith, A.G. Veneris, M.F. Ali, and A. Viglas, "Fault diagnosis and logic debugging using Boolean satisfiability," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol.24, no.10, pp.1606–1621, 2005.
[19] R. Williams, "A new algorithm for optimal 2-constraint satisfaction and its implications," Theoretical Computer Science, vol.348, pp.357–365, 2005.
[20] R. Williams, "Improving exhaustive search implies superpolynomial lower bounds," Proc. 42nd ACM symposium on Theory of computing, STOC'10, pp.231–240, ACM, New York, NY, USA, 2010.
[21] R. Williams, "Non-uniform ACC circuit lower bounds," Proc. Twenty-Sixth Annual IEEE Conference on Computational Complexity, pp.115–125, 2011.
[22] R. Williams, "New algorithms and lower bounds for circuits with linear threshold gates," Proc. 46th ACM symposium on Theory of computing, STOC'14, pp.194–202, 2014.

Kazuyuki Amano received his Ph.D in 1996 from Tohoku University. Since 2006, he has been with the Gunma University, where he is currently a professor in the Department of Computer Science. His research interest includes computational complexity and combinatorics. He is a member of IEICE and LA.

Atsushi Saito received both his bachelor's degree in 2010 and his master's in 2012 from the University of Electro-Communications, and is currently a graduate student of Department of Computer Science of Gunma University.