A Faster Algorithm for 0-1 Integer Programming from Communication Complexity

Deepanshu Kush
Department of Mathematics, IIT Bombay, Mumbai, India
deepkush@iitb.ac.in

Srikanth Srinivasan
Department of Mathematics, IIT Bombay, Mumbai, India
srikanth@math.iitb.ac.in

Abstract

We consider the 0-1 Integer Programming problem, where we are given a collection of linear inequalities (with integer coefficients) on n variables and the question is to check if there is a Boolean assignment to the variables that satisfies all of them. This problem generalizes the CNF-SAT problem since each clause of a CNF formula can be easily translated into a constraint of the above form.

We give a deterministic algorithm running in time $2^{n-n/O(\log(m\cdot M))}$ where M is an upper bound on the bit-complexity of the integer coefficients of the linear inequalities and m is the number of linear inequalities (for $m, M \leq 2^{n^{o(1)}}$). In particular, when $M = \mathrm{poly}(n)$ (i.e. the coefficients are in the range $[-2^{\mathrm{poly}(n)}, 2^{\mathrm{poly}(n)}]$) and $m = \mathrm{poly}(n)$ (i.e. polynomially many inequalities), our algorithm runs in time $2^{n-n/O(\log n)}$, matching the running times of the best CNF-SAT algorithm (due to Schuler (Journal of Algorithms, 2005)) in this range of parameters.

As far as we know, the previous best algorithm for this problem is that of Williams (STOC 2014), which runs in time $2^{n-n/O(\log M\cdot(\log m)^5)}\cdot\mathrm{poly}(Mmn)$, which for the particular range of parameters mentioned above is $2^{n-n/O((\log n)^6)}$. (Impagliazzo, Lovett, Paturi and Schneider (ECCC 2014) give a faster algorithm in the special case that $m = O(n)$.)

The algorithm is obtained by a reduction to the Orthogonal Vectors problem, which has been studied extensively. The reduction is obtained via a simple 1-sided error MA communication complexity protocol for any linear threshold function.

We also observe that any 2-sided error MA protocol (and hence in particular 2-sided error randomized communication complexity protocol) can be converted into a 1-sided error MA protocol with only a slight degradation of parameters. In principle, this idea could be useful in obtaining satisfiability algorithms for more general classes of constraints.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms → Mathematical Optimization → Discrete optimization

Keywords and phrases 0-1 Integer Programming, Satisfiability with Linear Threshold Constraints, MA Protocols in algorithm design

Digital Object Identifier 10.4230/LIPIcs...

1 The 0-1 Integer Programming problem

The 0-1 Integer Programming (I.P.) problem is defined as follows.

Input: A set of m linear inequalities (denoted by $C_1, \ldots, C_m$ respectively) in n variables $x_1, \ldots, x_n$, say

$$\sum_{j\in[n]} w_{i,j}x_j \geq \theta_i \tag{1}$$

Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

for $i \in [m]$, where $w_{i,j}, \theta_i \in \mathbb{Z}$ for all i, j. We call w_{ij} the "weights" corresponding to C_i and θ_i the "threshold value" corresponding to C_i. Throughout, we assume that M is an upper bound on the bit complexity of the numbers that appear in the inequalities (i.e. all the numbers are in the range $[-2^M, 2^M]$).

Output: A Boolean (i.e. $\{0,1\}$-valued) assignment to the variables $x_1, \ldots, x_n$ that satisfies all the inequalities, if one exists.[1]

We will call the tuple (n, m, M) the *parameters* of an instance of the 0-1 I.P. problem. Note that the size s of an instance with parameters (n, m, M) is lower bounded by $\Omega(n + m + M)$ and upper bounded by $O(nmM)$.

This is a well-known NP-complete problem, as is easy to prove by a simple reduction from CNF-SAT: each clause of a CNF formula can be easily transformed to an inequality of the above form. For instance, a clause of the form $x_1 \vee \neg x_2 \vee x_3$ can be expressed via the inequality

$$x_1 + (1 - x_2) + x_3 \geq 1, \text{ or equivalently, } x_1 - x_2 + x_3 \geq 0.$$

As a result, we do not expect to have polynomial-time algorithms for the 0-1 I.P. problem. Nevertheless, the problem is an important one in many computational settings, and it is desirable to have as fast an improvement over the trivial brute-force algorithm as possible. (Note that the brute-force algorithm for the 0-1 I.P. problem would take time $2^n \cdot \text{poly}(nmM)$.)

Such investigations have been conducted for many problems over the past two decades, and there is by now a large body of results in the area (see, e.g. [12] for an introduction). For example, in the case of the CNF-SAT problem, we have algorithms that count the number of satisfying assignments of a given CNF instance on n variables and m clauses in time $2^{n-n/O(\log(m/n))}$ [29, 6, 10, 18, 2, 7]. Significant improvements over these algorithms are related via the Strong Exponential Time Hypothesis [19] to improving the complexity of many other problems [34].

The 0-1 I.P. problem has also received quite a bit of attention. We only survey the very recent history and refer the reader to [13] for a more detailed account. Impagaliazzo, Paturi and Schneider [20] considered this problem in the setting when the number of non-zero $w_{i,j}$ is at most cn for some $c > 0$ and gave an algorithm that runs in time 2^{n-sn} where s is exponentially small in c [2]. This was strengthened by Impagliazzo, Lovett, Paturi and Schneider [17], who obtained a satisfiability algorithm that runs in time 2^{n-sn} for $s = 1/\operatorname{poly}(m/n)$. An algorithm with a worse running time but smaller space requirements was given by Bansal, Garg, Nederlof and Vyas [5].

While the algorithm of Impagliazzo et al. [17] has truly exponential savings when m is linear in n, it does not yield any improvement over brute force search when $m = n^2$ (say). A better algorithm for this range of parameters was devised by Williams [33], who showed that the 0-1 I.P. problem can be solved by an algorithm in time $2^{n-n/O((\log M)\cdot(\log m)^5)}$, which in the setting when $M = \text{poly}(n)$ and $m = \text{poly}(n)$ yields a running time of $2^{n-n/O(\log n)^6}$.

In this paper, we give an improved algorithm that matches the running time of the best CNF-SAT algorithm [29] in the above range of parameters. The main theorem is as follows.

[1] More precisely, this is the *feasibility* question for 0-1 Integer programs. However, the more general problem of *optimizing* a linear function over the solutions of a system of linear inequalities reduces to the feasibility question using a simple binary search. See, e.g., [33].

[2] The algorithm in [20] also works for a more general family of satisfiability problems for *depth-*2 *threshold circuits.*

▶ **Theorem 1** (Main theorem). *Assume n, m, M are growing integer parameters where $m, M \leq 2^{n^{o(1)}}$. There is a deterministic algorithm which solves the 0-1 I.P. problem in time $2^{n-n/O(\log(mM))}$ where the parameters of the input instance are (n, m, M).*

1.1 Techniques.

The proof idea is based on a template provided in a result of Williams [33]. Given a function $F : \{0,1\}^n \to \{0,1\}$ whose satisfiability we would like to check (in our setting, F is a conjunction of linear inequalities), we divide the input variables $x_1, \ldots, x_n$ into the two sets $x_1, \ldots, x_{n/2}$ and $x_{n/2+1}, \ldots, x_n$, which we denote $x_{\leq n/2}$ and $x_{>n/2}$ respectively. We then consider the Boolean $2^{n/2} \times 2^{n/2}$ matrix M, whose rows and columns are indexed by assignments σ and τ to $x_{\leq n/2}$ and $x_{>n/2}$ respectively; the entry $M(\sigma, \tau)$ of M is 1 iff $F(\sigma \cup \tau) = 1$.

Obviously, the matrix M cannot be constructed in time less than 2^n, but Williams' idea is to obtain some sort of "low-rank" decomposition for M, which can then be used to quickly check for satisfiability using algebraic algorithms such as the Fast Fourier Transform or Coppersmith's [9] rectangular Matrix Multiplication algorithm.

In [33], Williams carried out this idea for 0-1 I.P. by using circuit complexity ideas to obtain suitable bounded-depth circuits for F and then converting those to polynomials, which can then be quickly evaluated using the Fast Fourier Transform.

Here, we follow a slightly different route, inspired by a later result of Abboud, Williams and Yu [2]. In this paper, the authors study the *Orthogonal Vectors problem*, defined as follows. The input to the problem is two lists of vectors $A, B \subseteq \{0,1\}^d$ of size N each and the question is to check if there are vectors $u \in A$ and $v \in B$ that do not share an index where they are both 1 (we say that such a pair of vectors is orthogonal). The trivial algorithm for this problem runs in time $O(dN^2)$ and it remains an open question to give a $N^{2-\Omega(1)}$-time algorithm for this problem when $d = \omega(\log N)$. This problem is closely related to the Strong Exponential Time Hypothesis [32] and has been studied extensively in recent years (see the survey [34] for a comprehensive list).

Abboud et al. [2] give the best-known algorithm for the Orthogonal Vectors problem. Further, they use this algorithm to give improved satisfiability algorithms for the *Symmetric Boolean CSP problem*: this is an extension of CNF-SAT where each "clause" is allowed to check, on a subset of the input variables, for any function that is *symmetric*, i.e. only depends on the number of input bits in that subset that are set to 1. The satisfiability algorithm is obtained via a reduction to the Orthogonal Vectors problem for $N = 2^{n/2}$ and small d (this can be seen as a "low-rank decomposition" in the above sense). We describe this reduction below.

The function F in this setting is a conjunction of symmetric functions $F_1, \ldots, F_m : \{0,1\}^n \to \{0,1\}$. We split the input x into $x_{\leq n/2}$ and $x_{>n/2}$ as above. Each F_i is a symmetric function on a subset $x^{(i)} = x^{(i)}_{\leq n/2} \cup x^{(i)}_{>n/2}$ of the variables. We define $\{0,1\}$-vectors u_σ and v_τ for each assignment σ to $x_{\leq n/2}$ and τ to $x_{>n/2}$ such that u_σ and v_τ are orthogonal iff $\sigma \cup \tau$ is a satisfying assignment for F. Taking $A = \{u_\sigma \mid \sigma\}$ and $B = \{v_\tau \mid \tau\}$ then yields the reduction.

It remains to describe how u_σ and v_τ are constructed. Since each F_i is symmetric, the set of inputs on which it is *not* satisfied can be partitioned into sets of the form $S_p \times T_q$ ($p, q \in \{0, \ldots, n\}$) where S_p denotes the set of assignments σ of $x_{\leq n/2}$ so that the assignment to $x^{(i)}_{\leq n/2}$ has Hamming weight p, T_q is the set of assignments τ so that the assignment to $x^{(i)}_{>n/2}$ has Hamming weight q, and F_i rejects inputs of Hamming weight $p + q$. To construct

u_σ and v_τ, we set $d = m \cdot (n+1)^2$ which is identified with $[m] \times \{0, \ldots, n\} \times \{0, \ldots, n\}$. For (i, p, q) such that F_i rejects inputs of Hamming weight $p + q$, set the (i, p, q)th co-ordinate of u_σ (resp. v_τ) to 1 iff $\sigma \in S_p$ (resp. $\tau \in T_q$); all other co-ordinates are set to 0. It can now easily be seen that $\sigma \cup \tau$ is a satisfying assignment iff u_σ and v_τ are orthogonal. This finishes the argument.

To use this idea for the 0-1 I.P. problem, we abstract the above argument. Note that we only used the fact that for each i, the non-satisfying assignments of F_i can be written as a small union of explicit sets of the form $S \times T$ where S and T are sets of assignments to $x_{\leq n/2}$ and $x_{>n/2}$ respectively. This is true more generally of functions whose complements have small *non-deterministic communication complexity* [4]. Unfortunately, however, this is not true for functions defined by linear inequalities, which we call *linear threshold functions*.

Nevertheless, it is known by a result of Nisan [26] that any linear threshold function (and its complement, which is also a linear threshold function) has an efficient *randomized* communication protocol. On the face of it, it seems that we could hope to use this to obtain *randomized* algorithms for the 0-1 I.P. problem by using the randomized protocol as a randomized reduction to the Orthogonal Vectors problem (exactly as above). However, this does not work, because the randomized protocols for threshold functions necessarily have 2-*sided error* and hence can give false positives as well as false negatives. This means that the randomized algorithm may misclassify non-satisfying assignments as satisfying assignments, which is quite bad since there could, in the worst case, be only one satisfying assignment and $2^n - 1$ non-satisfying assignments. In such a scenario, it is not clear how to filter out the satisfying assignment efficiently.

At this point, what comes to our aid is non-determinism. As noted above, even an efficient *non-deterministic* communication protocol would result in an efficient reduction to Orthogonal Vectors. This also turns out to be true for non-deterministic randomized communication protocols, or MA communication protocols, and randomized reductions. Further, if the MA communication protocol happens to have 1-sided error (suitably defined), then the reduction may misclassify satisfying assignments as non-satisfying (with a small probability) but *not vice-versa.*

We show how to obtain a MA communication protocol of small communication complexity with 1-sided error for any linear threshold function. This turns to be just a simple truncation of Nisan's randomized communication complexity protocol [26]. Turning this into a reduction as above, we get a randomized reduction to Orthogonal vectors. We then show how to efficiently derandomize it using *Small-bias spaces*, a standard tool in the derandomization literature [24], to obtain a deterministic reduction to the Orthogonal vectors problem. This yields the algorithm.

We then turn to MA communication protocols in general and ask if general MA communication protocols with 2-sided error can be converted to 1-sided error protocols of comparable efficiency. (As noted above, a 1-sided error protocol is crucial to our satisfiability algorithm.) It is a standard fact in Computational Complexity (see, e.g. [14, Section 3.3]) that every language $L \subseteq \{0,1\}^*$ in the (Turing Machine) complexity class MA has an MA protocol with 1-sided error. We observe that essentially the same proof carries through in the communication complexity setting. This is not too surprising: any reasonably black-box proof technique in computational complexity should be expected to yield the same results in the communication complexity setting.[3] However, we could not find this particular result anywhere in the literature, and given the usefulness of MA protocols as reductions to Orthogonal vectors (as

[3] This intuition was communicated to us by Mika Göös.

above) and other similar problems [1, 28, 8], we feel that it is worth pointing out.

1.2 Related work.

We note that MA communication protocols have been used as reductions in many recent results in the area [1, 21, 28, 8], starting with the work of Abboud, Rubinstein and Williams [1]. In these results, the MA protocols have been used to reduce CNF-SAT or the Orthogonal Vectors problem to (approximation variants of) harder problems. Here, we try to give an efficient reduction to the Orthogonal vectors problem from a more expressive satisfiability question.

2 Preliminaries

Given vectors $u, v \in \{0,1\}^d$ (for some d), we use $\langle u, v\rangle$ to denote their standard inner product (i.e. $\sum_{i\in[d]} u_i v_i$) modulo 2.

2.1 Orthogonal Vectors

Given two vectors $u, v \in \{0,1\}^d$, we say that they are *orthogonal* if $\bigvee_{i\in[d]} u_i \wedge v_i = 0$ (or equivalently, if there is no $i \in [d]$ such that $u_i = v_i = 1$). The Orthogonal Vectors problem is defined as follows.

Input: $A, B \subseteq \{0,1\}^d$ where $|A| = |B| = N$.
Output: A pair $u \in A$ and $v \in B$ such that u and v are orthogonal, if such a pair exists.

We will use the following deterministic algorithm of this problem due to Chan and Williams [7], building on an earlier randomized algorithm due to Abboud, Williams and Yu [2].

▶ **Theorem 2** (Chan-Williams [7]). *There is a deterministic algorithm for the Orthogonal Vectors problem that runs in time $N^{2-1/O(\log(d/\log N))}$, provided $d \leq 2^{(\log N)^{o(1)}}$.*

2.2 MA protocols in communication complexity

We recall (see, e.g., [15]) that an ε-error MA communication complexity protocol for a Boolean function $F : \{0,1\}^{n/2} \times \{0,1\}^{n/2} \to \{0,1\}$ is a communication protocol where the players Alice and Bob receive, in addition to their respective inputs $x, y \in \{0,1\}^{n/2}$, an additional input $w \in \{0,1\}^k$ (which we think of as a "proof string").

Given their inputs x, y and the proof string w, Alice and Bob execute a *randomized* protocol Π (defined as a distribution over deterministic protocols) which has the following properties:

- Completeness: $F(x,y) = 1 \Rightarrow \exists w \; s.t. \; \Pr_\Pi[\Pi(x,y,w) = 1] \geq 1 - \varepsilon$.
- Soundness: $F(x,y) = 0 \Rightarrow \forall w \;\; \Pr_\Pi[\Pi(x,y,w) = 1] \leq \varepsilon$.

▶ Remark. Typically, the error ε is taken to be a constant (say 1/4), but we will use it as a parameter.

Further, we say that an MA protocol has 1-sided error if we have instead that whenever $F(x,y) = 1$, there is a w such that $\Pr_\Pi[\Pi(x,y,w) = 1] = 1$.

The complexity of an MA protocol is defined to be the sum of k (the length of the proof string) and the total number of bits communicated by Alice and Bob in the worst case.

2.3 Small bias spaces and subspace avoidance

We use the notion of a *small-bias space*, which is a standard tool in the derandomization literature [24, 3].

▶ **Definition 3** ([24]). We say that a multiset $S \subseteq \{0,1\}^s$ is an ε-*bias space* if for any non-zero $u \in \{0,1\}^s$, we have

$$\left| \Pr_{a \in S}[\langle u, a \rangle = 1] - \frac{1}{2} \right| \leq \varepsilon.$$

Near-optimal explicit constructions of ε-bias spaces first appeared in the work of Naor and Naor [24]. Better (i.e. smaller) constructions are known [3, 31], but we do not need them here.

▶ **Theorem 4** (Naor and Naor [24]). *There is a deterministic algorithm which when given as input an $s \in \mathbb{N}$ and $\varepsilon \in (0, 1/2)$, produces a multiset $S \subseteq \{0,1\}^s$ of size* $\mathrm{poly}(s/\varepsilon)$ *that is an ε-biased space. Further, the algorithm runs in time* $\mathrm{poly}(s/\varepsilon)$.

The following is a standard fact about ε-bias spaces, which follows from [11, Lemma 2.5].

▶ Fact 5. Let $S \subseteq \{0,1\}^s$ be an ε-biased space. Viewing $\{0,1\}^s$ as $\mathbb{F}_2^s$, let V be an affine subspace of $\{0,1\}^s$ of codimension t. Then, $\Pr_{a \in S}[a \in V] \leq (1/2^t) + \varepsilon$.

3 An MA protocol with 1-sided error for any threshold function

We recall that a linear threshold function is a Boolean function $F : \{0,1\}^n \to \{0,1\}$ that can be specified by a linear inequality in the following sense.

$$F(x_1, \ldots, x_n) = 1 \Leftrightarrow \sum_{i=1}^{n} w_i x_i \geq \theta \tag{2}$$

for $w_1, \ldots, w_n, \theta \in \mathbb{R}$.

We consider the communication complexity of a linear threshold function where some $n/2$ input bits are given to Alice and the rest to Bob. Nisan [26] showed that any threshold function has an ε-error *randomized* communication complexity protocol of complexity $O(\log n)$. However, his protocol is a 2-sided error protocol[4] and this makes it unsuitable for our algorithmic application.

Nevertheless, the ideas behind Nisan's protocol easily yield a 1-sided MA protocol as we argue below.

▶ **Definition 6.** For a 0-1 vector v, let $v(i)$ denote its i^{th} bit and let $v(< i)$ denote its first $i - 1$ bits (which is just an empty string if $i = 1$). We define the inner product of two empty strings to be 0.

▶ **Theorem 7.** *Let F be as given above and assume that the w_i and θ are integers of bit complexity M (i.e. in the range $[-2^M, 2^M]$). Then, there is an ε-error MA protocol for F with complexity $O(\log((M + \log n)/\varepsilon)) = O(\log M + \log\log n + \log(1/\varepsilon))$. Further, the protocol has 1-sided error.*

[4] It is not hard to argue that there is no 1-sided error randomized protocol of comparable efficiency.

Proof. Suppose $X \cup Y$ is a partition of the variable set $\{x_1, \ldots, x_n\}$ into parts of equal sizes, where Alice is given the input bits in X and Bob the bits in Y. We assume that Alice and Bob are given as inputs partial assignments $\sigma : X \to \{0,1\}$ and $\tau : Y \to \{0,1\}$ respectively. Set $R = \lceil \log(n \cdot 2^{M+1}) \rceil$.

We define protocols $\Pi_{\underline{z}}$, indexed by $\underline{z}$ which is a collection of vectors $z_1, \ldots, z_t$ (t to be chosen in terms of ε later) each chosen independently and uniformly at random from $\{0,1\}^R$, as follows.

Protocol $\Pi_{\underline{z}}$ on input (σ, τ):

1. Prover provides: an index $i \in [R+1]$.
2. Alice computes $F_\sigma := \theta - \sum_{x_j \in X} w_j \sigma(x_j) + n \cdot 2^M$ and Bob computes $F_\tau := \sum_{x_j \in Y} w_j \tau(x_j) + n \cdot 2^M$, both R bit numbers. We consider F_σ, F_τ as elements of $\{0,1\}^R$ (using binary notation).
3. If the given index $i \in [R]$, they check if

   (i) $F_\sigma(i) < F_\tau(i)$, and

   (ii) $\langle F_\sigma(< i), z_j(< i) \rangle = \langle F_\tau(< i), z_j(< i) \rangle \ \forall j \in [t]$.

   If *both* these conditions hold, then they output that $\sigma \cup \tau$ *satisfies* F. If either of these conditions fails to hold, then they output that $\sigma \cup \tau$ does *not* satisfy F.
4. If $i = R+1$, then they only check condition (ii). If it holds, they output that $\sigma \cup \tau$ *satisfies* F. Otherwise, they output that $\sigma \cup \tau$ does *not* satisfy F.

Note that F_σ and F_τ are both in the range $[0, n \cdot 2^{M+1}]$ and thus, can each be represented using R bits. Further, F is satisfied by an assignment (σ, τ) if and only if $F_\sigma \leq F_\tau$. Also, observe that the protocol has complexity $O(\log R) + O(t)$, as both conditions can be checked by communicating $O(t)$ many bits and the length of the proof string $i \in [R+1]$ is $O(\log R) = O(\log(M + \log n))$.

Let us first verify the completeness property of this randomized protocol for F. Suppose $\sigma \cup \tau$ satisfies F (i.e. $F(\sigma, \tau) = 1$). Then $F_\sigma \leq F_\tau$. If $F_\sigma < F_\tau$, then there exists a bit $i \in [R]$ such that $F_\sigma(i) < F_\tau(i)$ and $F_\sigma(< i) = F_\tau(< i)$. This i is the most significant bit at which they differ. If $F_\sigma = F_\tau$, then for $i = R+1$, we trivially have $F_\sigma(< i) = F_\tau(< i)$. Thus, in either case there exists a proof string i such that $F_\sigma(< i) = F_\tau(< i)$ and hence, regardless of the choice of the random bits $\underline{z}$, the protocol outputs the correct answer.

Now let us verify the soundness property. Suppose we are given an instance (σ, τ) such that $\sigma \cup \tau$ does *not* satisfy F i.e. $F_\sigma > F_\tau$. We need to check that for every proof string $i \in [R+1]$, the randomized protocol errs with a small probability. Suppose the proof string $i \in [R]$. Observe that the protocol errs only when $F_\sigma(i) < F_\tau(i)$ and the corresponding inner products with $z_j(< i)$ match. But if for an $i \in [R]$ $F_\sigma(i) < F_\tau(i)$, then along with $F_\sigma > F_\tau$, this implies that $F_\sigma(< i) \neq F_\tau(< i)$. And if $i = R+1$, then $F_\sigma > F_\tau$ trivially implies that $F_\sigma(< i) \neq F_\tau(< i)$. Now using the fact that the probability that $\langle v, z \rangle = \langle u, z \rangle$ for distinct vectors v, u where z is chosen uniformly at random is precisely $1/2$ (which in turn follows easily from the Schwartz-Zippel Lemma), and the fact that the z_j are chosen independently, we deduce that the probability that $\langle F_\sigma(< i), z_j(< i) \rangle = \langle F_\tau(< i), z_j(< i) \rangle \forall j \in [t]$ is at most $\frac{1}{2^t}$. So we set $t = \lceil \log \frac{1}{\varepsilon} \rceil$ to obtain an error of at most ε.

Hence, this is an MA protocol with 1-sided error for F with complexity $O(\log(M + \log n)) + O(t) = O(\log((M + \log n)/\varepsilon))$. ◀

▶ Remark. It is known [23] that any Linear Threshold function on n variables can be represented with integer weights w_i ($i \in [n]$) and θ as above with $|w_i|, |\theta| \leq n^{O(n)}$. Thus, Alice and Bob can always agree on such a representation beforehand and then run the above protocol, which will have complexity at most $O(\log(n/\varepsilon))$.

Unfortunately, the proof of the above fact from [23] is not algorithmic, and hence we cannot assume this bound on weights in our algorithmic application given below.

4 The reduction to Orthogonal Vectors

We use the ideas behind the MA protocol for threshold functions in Section 3 to give a reduction from the 0-1 I.P. problem to the Orthogonal Vectors problem. An analogous statement could be made for other classes of functions that have efficient MA protocols, but we would need the protocol to satisfy some additional explicitness properties. Hence, we omit the general statement here.

We start with a *randomized reduction*, which we will later derandomize.

▶ **Lemma 8.** *There is a randomized algorithm which, when given as input an instance I of the* 0-1 *I.P. problem with parameters (n, m, M), produces an instance (A, B) of the Orthogonal vectors problem with $A, B \subseteq \{0,1\}^d$, $|A| = |B| = N = 2^{n/2}$, and $d = O(m(M + \log n))^2$ such that:*

- *If I is unsatisfiable, then with probability* 1*, there is no orthogonal pair of vectors $u \in A$ and $v \in B$.*
- *If I is satisfiable, then with probability at least* 1/2*, there is an orthogonal pair of vectors $u \in A$ and $v \in B$. Given such an orthogonal vector, a satisfying assignment for I can be recovered in time $2^{n/2} \cdot \mathrm{poly}(nmM)$.*

Further, the reduction runs in time $2^{n/2} \cdot \mathrm{poly}(nmM)$.

Proof. We use ideas from the proof of theorem 7 to construct this randomized algorithm. Suppose we are given an instance I consisting of m linear inequalities $C_1, \ldots, C_m$ (which we shall henceforth refer to as "clauses") defined by eq. (1), with parameters (n, m, M). Then algorithm 1 (described below) produces the desired instance (A, B) of the Orthogonal Vectors problem.

Let us now verify the correctness of algorithm 1.

- Suppose I is unsatisfiable. We show that $u_\sigma \in A$ and $v_\tau \in B$ are not orthogonal for every $\sigma : X \to \{0,1\}$ and $\tau : Y \to \{0,1\}$. Since $\sigma \cup \tau$ is not a satisfying assignment, there is a clause $C \in \{C_1, \ldots, C_m\}$ such that $C_\sigma > C_\tau$ (as defined as in line 3). But this means that there is a bit $i \in [R]$ (as these are R bit numbers) at which $C_\sigma(i) = 1, C_\tau(i) = 0$ and $C_\sigma(< i) = C_\tau(< i)$. In other words, i is the most significant bit at which they differ. Now, since $C_\sigma(< i)$ and $C_\tau(< i)$ are equal, for every set of random vectors $\underline{z}$, there is a choice of $\bar{b}$ given by $b_j = \langle C_\tau(< i), z_j(< i)\rangle = \langle C_\sigma(< i), z_j(< i)\rangle$ for $j \in [t]$ such that for this $\bar{b}$ and the previously specified i and C, we have $u_\sigma(C, i, \bar{b}) = v_\tau(C, i, \bar{b}) = 1$ due to the construction in line 7 and line 8, which immediately implies that u_σ and v_τ are not orthogonal.
- Suppose I is satisfiable and that $\sigma \cup \tau$ is a satisfying assignment. Then for every clause $C \in \{C_1, \ldots, C_m\}$, we must have $C_\sigma \le C_\tau$. We shall prove that u_σ and v_τ are *not* orthogonal with probability $\le \frac{mR}{2^t}$ (which is at most $\frac{1}{2}$ by the choice of t). Note that it is enough to show that for a given C and $i \in [R]$, the probability that there exists $\bar{b}$ such that $u_\sigma(C, i, \bar{b}) = v_\tau(C, i, \bar{b}) = 1$ is at most $\frac{1}{2^t}$; the desired upper bound on the error probability then directly follows from the union bound. By line 8, we need to find the

Algorithm 1 Reduction from the 0-1 I.P. problem to Orthogonal Vectors

```
Input: Boolean Variables $x_1, \dots, x_n$ and clauses $C_1, \dots, C_m$ given by integral weights $w_{ij}$
    and thresholds $\theta_i$ respectively; each from the range $[-2^M, 2^M]$
Output: An instance $(A, B)$ of the Orthogonal vectors problem with $A, B \subseteq \{0,1\}^d$, $|A| =$
    $|B| = N = 2^{n/2}$, and $d = O(m(M + \log n))^2$
1: Partition the set of variables into $X$ and $Y$, each of size $n/2$.
2: For a clause $C \in \{C_1, \dots, C_m\}$ given by weights $\{w_j\}$ and threshold value $\theta$ and a partial
   assignment $\sigma : X \to \{0,1\}$, define $C_\sigma := \theta - \sum_{x_j \in X} w_j \sigma(x_j)$ and for a partial assignment
   $\tau : Y \to \{0,1\}$, define $C_\tau := \sum_{x_j \in Y} w_j \tau(x_j)$.
3: For every $\sigma : X \to \{0,1\}$ and $\tau : Y \to \{0,1\}$, redefine $C_\sigma \leftarrow C_\sigma + n \cdot 2^M$ and
   $C_\tau \leftarrow C_\tau + n \cdot 2^M$ so that the newly defined numbers are both non-negative and in the
   range $[0, n \cdot 2^{M+1}]$.
4: Then $\sigma \cup \tau$ does not satisfy $C$ if and only if $C_\sigma > C_\tau$.
5: Let $R = \lceil \log(n \cdot 2^{M+1}) \rceil$. Then each $C_\sigma$ and $C_\tau$ can be represented as a binary string of
   length $R$.
6: Pick $t = \lceil \log m + \log R + 1 \rceil$ vectors $z_1, \dots, z_t$ independently from $\{0,1\}^R$ and uniformly
   at random. Let $\underline{z} = (z_1, \dots, z_t)$ denote this choice of random vectors.
7: For every clause $C$, a choice of $\bar{b} = (b_1, \dots, b_t) \in \{0,1\}^t$ and an $i \in [R]$, construct the
   sets $A_{\underline{z}}^{(C,i,\bar{b})}$ and $B_{\underline{z}}^{(C,i,\bar{b})}$, defined as follows:

       $A_{\underline{z}}^{(C,i,\bar{b})} := \{\sigma : X \to \{0,1\} | C_\sigma(i) = 1 \text{ and } \langle C_\sigma(< i), z_j(< i) \rangle = b_j \forall j \in [t]\}$

       $B_{\underline{z}}^{(C,i,\bar{b})} := \{\tau : Y \to \{0,1\} | C_\tau(i) = 0 \text{ and } \langle C_\tau(< i), z_j(< i) \rangle = b_j \forall j \in [t]\}.$

8: For a partial assignment $\sigma : X \to \{0,1\}$, define a 0-1 vector $u_\sigma$ of length $d = m \cdot R \cdot 2^t$
   (whose positions are indexed by tuples of the form $(C, i, \bar{b})$) as follows: $u_\sigma(C, i, \bar{b}) = 1 \Leftrightarrow$
   $\sigma \in A_{\underline{z}}^{(C,i,\bar{b})}$. Similarly, for a partial assignment $\tau : Y \to \{0,1\}$, define $v_\tau$ as follows:
   $v_\tau(C, i, \bar{b}) = 1 \Leftrightarrow \tau \in B_{\underline{z}}^{(C,i,\bar{b})}$.
9: Output the lists $A = \{u_\sigma | \sigma : X \to \{0,1\}\}$ and $B = \{v_\tau | \tau : Y \to \{0,1\}\}$.
```

probability that $(\sigma, \tau) \in \bigcup_{\bar{b}} A_{\underline{z}}^{(C,i,\bar{b})} \times B_{\underline{z}}^{(C,i,\bar{b})}$ given that $C_\sigma \leq C_\tau$. Stated in words, we need to find the probability of the event that $\langle C_\tau(< i), z_j(< i) \rangle = \langle C_\sigma(< i), z_j(< i) \rangle$ for all $j \in [t]$, given that $C_\sigma \leq C_\tau$ and $C_\sigma(i) = 1, C_\tau(i) = 0$ (the latter because of the way the sets $A_{\underline{z}}^{(C,i,\bar{b})}$ and $B_{\underline{z}}^{(C,i,\bar{b})}$ are defined; see line 7). Note that these conditions together imply that $C_\sigma(< i) \neq C_\tau(< i)$. Finally, the claim follows at once when we observe that the probability that $\langle v, z \rangle = \langle u, z \rangle$ for two distinct vectors v, u where z is chosen uniformly at random is precisely $1/2$.

◀

Next, we show that algorithm 1 can be easily derandomized.

▶ **Lemma 9.** *We can modify algorithm 1 so that it uses only $r = O(\log m + \log R)$ random bits at the expense of the success probability (in the case that I is satisfiable) dropping to $1/4$.*

Proof. Note that the only randomness used in the algorithm was the random choice of $\underline{z} = (z_1, \dots, z_t) \in \{0,1\}^{R \cdot t}$. Further, the only property of $\underline{z}$ that was used in the analysis was that when u and v are distinct vectors of length $(i - 1) \leq d$, the probability that $\langle u, z_j(< i) \rangle = \langle v, z_j(< i) \rangle$ for all $j \in [t]$ is at most $1/2^t$.

To (approximately) preserve this property, we claim that it suffices to choose $\underline{z}$ u.a.r. from an ε-biased space $S \subseteq \{0,1\}^{R\cdot t}$ (see Section 2.3) where $\varepsilon = 1/2^{t+1}$. To see this, fix any distinct vectors u, v of length $(i-1)$ and let w be their bitwise XOR (which is a non-zero vector). Note that $\langle u, z_j(< i)\rangle = \langle v, z_j(< i)\rangle$ for each $j \in [t]$ iff $\langle w, z_j(< i)\rangle = 0$ for each $j \in [t]$. This is further equivalent to having $\langle w_j, \underline{z}\rangle = 0$ for each $j \in [t]$, where $w_j \in \{0,1\}^{R\cdot t}$ is the non-zero vector that has w in the first $(i-1)$ positions in its jth block (i.e. the positions $(j-1)R+1, \ldots, (j-1)R+(i-1)$) and 0 elsewhere.

Since w is a non-zero vector, the vectors w_j $(j \in [t])$ span a subspace of dimension t. Thus, the probability that $\langle w_j, \underline{z}\rangle = 0$ for each $j \in [t]$ is equivalent to requiring that $\underline{z}$ lie in a subspace of $\{0,1\}^{R\cdot t}$ of codimension t. By Fact 5, this probability is bounded by $(1/2^t) + \varepsilon = 3/2^{t+1}$.

Repeating the argument in the proof of Lemma 8 with this choice of $\underline{z}$ and the above property, we see that when I is satisfiable, the probability that the algorithm does *not* output (A, B) with a pair of orthogonal vectors is at most $3mR/2^{t+1} \leq 3/4$. Hence, the success probability of the reduction is $1/4$. (The analysis in the case that I is unsatisfiable is unchanged.)

Finally, as we have explicit ε-biased sets of size $\mathrm{poly}(Rt/\varepsilon) = \mathrm{poly}(mR)$ (Theorem 4), the randomized algorithm only uses $O(\log m + \log R)$ many random bits. The running time remains $2^{n/2} \cdot \mathrm{poly}(nmM)$. ◀

The main theorem (Theorem 1) now follows easily.

Proof of Theorem 1. We can run the modified version of algorithm 1 from Lemma 9 which uses only r many random bits for all possible choices of them i.e. $2^r = \mathrm{poly}(mR)$ many times. This reduction would still take $O(2^{n/2} \cdot \mathrm{poly}(nmM))$ time (recall $R = O(M + \log n)$) and produce pairs of subsets (A_i, B_i) $(i \in [2^r])$ of $\{0,1\}^d$ such that: (a) if I is unsatisfiable, no (A_i, B_i) contains an orthogonal pair of vectors, (b) if I is satisfiable, then there is some $i \in [2^r]$ such that (A_i, B_i) contains an orthogonal pair of vectors; further, it follows from the proof of Lemma 8 that for any orthogonal pair (u_σ, v_τ), the Boolean assignment determined by σ and τ is a satisfying assignment to the set of linear inequalities.

Running the deterministic algorithm for Orthogonal Vectors in Theorem 2 on each pair of lists (A_i, B_i) of size $N = 2^{n/2}$ and with $d = O(m(M + \log n))^2$ yields a total running time of $2^{n - n/O(\log(d/n))} = 2^{n - n/O(\log(mM))}$. This yields a total running time of $2^r \cdot 2^{n-n/O(\log(mM))} = 2^{n-n/O(\log(mM))}$ for $m, M \leq 2^{n^{o(1)}}$. ◀

▶ Remark. Using the same or similar ideas, we can also handle other kinds of constraints that involve weighted sums of the input variables. Examples include arbitrary functions of weighted sums with small weights (already done by Abboud et al. [2]), weighted equalities or Exact Threshold functions as defined by [16] (which can be written as the conjunction of two inequalities) and complements of weighted equalities (which have efficient non-deterministic communication complexity protocols).

5 More on MA versus MA with 1-sided error

In this section, we show that any function that has efficient MA protocols (with possibly 2-sided error) also has efficient 1-sided error MA protocols. In particular, this result implies that any function that has efficient 2-sided error randomized protocols also has efficient 1-sided error MA protocols. This generalizes the fact we proved for threshold functions in

Section 3.[5] In principle, this idea can be used for checking satisfiability for conjunctions of more general classes of constraints than the threshold constraints we considered in Theorem 1.

The theorem we prove is as follows.

▶ **Theorem 10.** *Let $F : \{0,1\}^{n/2} \times \{0,1\}^{n/2} \to \{0,1\}$ be any function that has a $1/4$-error MA protocol of complexity k. Then F also has an $O(1/n^2)$-error MA protocol with 1-sided error of complexity $O(k \log n)$.*

The proof follows an idea of Lautemann [22] who used it to give an alternate proof of the fact that BPP $\subseteq \Sigma_2^P \cap \Pi_2^P$ (due originally to Gacs and Sipser [30]) in the Turing Machine setting. The idea is for the prover to convince the verifier that a computation accepts with high probability by showing that the set of "accepting" random strings can be "shifted" a few times to cover the entire universe of possible random strings. If most strings are accepting, this can indeed be done; otherwise, any small set of shifts can only cover a negligible fraction of the universe. This can be checked (with 1-sided error) by a verifier who simply chooses a uniformly random string and sees that it belongs to one of the given shifts.

Proof. We can assume that $k \leq n$, since otherwise the conclusion of the theorem is trivial. Let Π be an MA protocol of complexity k for F. We thus have

- $F(x,y) = 1 \Rightarrow \exists w \in \{0,1\}^k$ *s.t.* $\Pr_\Pi[\Pi(x,y,w) = 1] \geq 3/4$.
- $F(x,y) = 0 \Rightarrow \forall w \in \{0,1\}^k,\ \Pr_\Pi[\Pi(x,y,w) = 1] \leq 1/4$.

By repeating the randomized protocol $O(\log n)$ times and taking the majority vote, Alice and Bob can reduce the error to $1/n^2$. We call this protocol Π'. Note that Π' has the same proof length as Π (which is at most k) but has communication $O(k \log n)$.

We assume that the protocol Π' uses r random bits. It is a standard fact (implied by the proof of Newman's theorem [25]; see also [27, Theorem 3.5]) in randomized communication complexity that any randomized communication complexity protocol for a Boolean function with error ε can be assumed to use $O(\log n + \log(1/\varepsilon))$ bits. It is easy to observe that the same proof also works for MA protocols. So henceforth we will assume that $r \leq c \log n$ for some absolute constant c.

We now show how to obtain a 1-sided error protocol Π''. Given inputs x, y to Alice and Bob respectively, the protocol is as follows.
Protocol Π'' on input (x,y):

1. Prover provides: a string $w \in \{0,1\}^k$ and strings $z^{(1)}, \ldots, z^{(c)} \in \{0,1\}^r$.
2. Alice and Bob choose a string $z \in \{0,1\}^r$ u.a.r. and run the protocol Π' with random strings $z \oplus z^{(i)}$ (the bitwise XOR of z and $z^{(i)}$) for each $i \in [c]$. If Π' accepts on any of these random strings, Alice and Bob accept. Otherwise, they reject.

Clearly, the protocol has complexity $O(ck \log n) = O(k \log n)$.

We analyze the correctness of the protocol. Consider first the case when $F(x,y) = 0$. In this case, no matter which proof string w is supplied, a run of Π' on input (x,y,w) with a random string $\sigma \in \{0,1\}^r$ accepts with probability at most $O(1/n^2)$. Since z (as chosen by Alice and Bob in Step 2) has the uniform distribution, so does the string $z \oplus z^{(i)}$ for each

[5] There is a small caveat here to do with parameters. The MA protocol we gave in Section 3 for threshold functions is more efficient than the one that is implied by the results in this section. This results in a more efficient algorithm than we would obtain were we to use the results from this section instead.

$i \in [c]$. Thus, the probability that the protocol Π' accepts on each such string is at most $O(1/n^2)$. By a union bound, the probability that Π'' accepts is at most $O(c/n^2) = O(1/n^2)$.

Now consider the case when $F(x, y) = 1$. We know that there is a $w \in \{0,1\}^k$ such that Π' rejects only with probability $O(1/n^2)$. Let $A \subseteq \{0,1\}^r$ be the set of strings z such that Π' accepts (x, y, w) on random string z. We have $|A| \geq 2^r(1 - O(1/n^2))$. It suffices to show that there is a choice of $z^{(1)}, \ldots, z^{(c)} \in \{0,1\}^r$ such that for *every* $z \in \{0,1\}^r$, there is a $z^{(i)}$ such that $z \oplus z^{(i)} \in A$.

We show the existence of such a $z^{(1)}, \ldots, z^{(c)}$ by the probabilistic method. Choose $z^{(1)}, \ldots, z^{(c)}$ i.u.a.r. from $\{0,1\}^r$. Fix a $z \in \{0,1\}^r$. Since $z \oplus z^{(i)}$ has the uniform distribution over $\{0,1\}^r$ for each $i \in [c]$, we have

$$\Pr_{z^{(1)},\ldots,z^{(c)}}[\forall i \in [c], z \oplus z^{(i)} \notin A] = \prod_{i \in [c]} \Pr_{z^{(i)}}[z \oplus z^{(i)} \notin A] \leq \left(\frac{O(1)}{n^2}\right)^c < \frac{1}{n^c}.$$

By a union bound, the probability that there is a $z \in \{0,1\}^r$ such that for every $i \in [c]$, $z \oplus z^{(i)} \notin A$ is strictly less than $2^r/n^c \leq 1$ (since $r \leq c \log n$). This shows the existence of $z^{(1)}, \ldots, z^{(c)}$ as required. ◀

Acknowledgements. We thank Mika Göös for a helpful discussion on communication complexity classes.

References

1 Amir Abboud, Aviad Rubinstein, and R. Ryan Williams. Distributed PCP theorems for hardness of approximation in P. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 25–36, 2017. URL: `https://doi.org/10.1109/FOCS.2017.12`, `doi:10.1109/FOCS.2017.12`.

2 Amir Abboud, Richard Ryan Williams, and Huacheng Yu. More applications of the polynomial method to algorithm design. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 218–230, 2015. URL: `https://doi.org/10.1137/1.9781611973730.17`, `doi:10.1137/1.9781611973730.17`.

3 Noga Alon, Oded Goldreich, Johan Håstad, and René Peralta. Simple construction of almost k-wise independent random variables. *Random Struct. Algorithms*, 3(3):289–304, 1992. URL: `https://doi.org/10.1002/rsa.3240030308`, `doi:10.1002/rsa.3240030308`.

4 László Babai, Peter Frankl, and Janos Simon. Complexity classes in communication complexity theory (preliminary version). In *27th Annual Symposium on Foundations of Computer Science, Toronto, Canada, 27-29 October 1986*, pages 337–347, 1986. URL: `https://doi.org/10.1109/SFCS.1986.15`, `doi:10.1109/SFCS.1986.15`.

5 Nikhil Bansal, Shashwat Garg, Jesper Nederlof, and Nikhil Vyas. Faster space-efficient algorithms for subset sum and k-sum. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 198–209, 2017. URL: `http://doi.acm.org/10.1145/3055399.3055467`, `doi:10.1145/3055399.3055467`.

6 Chris Calabro, Russell Impagliazzo, and Ramamohan Paturi. A duality between clause width and clause density for SAT. In *21st Annual IEEE Conference on Computational Complexity (CCC 2006), 16-20 July 2006, Prague, Czech Republic*, pages 252–260, 2006. URL: `https://doi.org/10.1109/CCC.2006.6`, `doi:10.1109/CCC.2006.6`.

7 Timothy M. Chan and Ryan Williams. Deterministic apsp, orthogonal vectors, and more: Quickly derandomizing razborov-smolensky. In *Proceedings of the Twenty-Seventh*

Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016, pages 1246–1255, 2016. URL: `https://doi.org/10.1137/1.9781611974331.ch87`, `doi:10.1137/1.9781611974331.ch87`.

8 Lijie Chen. On the hardness of approximate and exact (bichromatic) maximum inner product. In *33rd Computational Complexity Conference, CCC 2018, June 22-24, 2018, San Diego, CA, USA*, pages 14:1–14:45, 2018. URL: `https://doi.org/10.4230/LIPIcs.CCC.2018.14`, `doi:10.4230/LIPIcs.CCC.2018.14`.

9 Don Coppersmith. Rapid multiplication of rectangular matrices. *SIAM J. Comput.*, 11(3):467–471, 1982.

10 Evgeny Dantsin and Edward A. Hirsch. Worst-case upper bounds. In *Handbook of Satisfiability*, pages 403–424. 2009. URL: `https://doi.org/10.3233/978-1-58603-929-5-403`, `doi:10.3233/978-1-58603-929-5-403`.

11 Anindya De, Omid Etesami, Luca Trevisan, and Madhur Tulsiani. Improved pseudorandom generators for depth 2 circuits. *Electronic Colloquium on Computational Complexity (ECCC)*, 16:141, 2009. URL: `http://eccc.hpi-web.de/report/2009/141`.

12 Fedor V. Fomin and Dieter Kratsch. *Exact Exponential Algorithms*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2010. URL: `https://doi.org/10.1007/978-3-642-16533-7`, `doi:10.1007/978-3-642-16533-7`.

13 Krasimira Genova and Vassil Guliashki. Linear integer programming methods and approaches - a survey. *Cybernetics and Information Technologies*, 11(1):3–25, 2011.

14 Oded Goldreich and David Zuckerman. *Another Proof That $\mathcal{BPP} \subseteq \mathcal{PH}$ (and More)*, pages 40–53. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. URL: `https://doi.org/10.1007/978-3-642-22670-0_6`, `doi:10.1007/978-3-642-22670-0_6`.

15 Mika Göös, Toniann Pitassi, and Thomas Watson. The landscape of communication complexity classes. In *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, pages 86:1–86:15, 2016. URL: `https://doi.org/10.4230/LIPIcs.ICALP.2016.86`, `doi:10.4230/LIPIcs.ICALP.2016.86`.

16 Kristoffer Arnsfelt Hansen and Vladimir V. Podolskii. Exact threshold circuits. In *Proceedings of the 25th Annual IEEE Conference on Computational Complexity, CCC 2010, Cambridge, Massachusetts, USA, June 9-12, 2010*, pages 270–279, 2010. URL: `https://doi.org/10.1109/CCC.2010.33`, `doi:10.1109/CCC.2010.33`.

17 Russell Impagliazzo, Shachar Lovett, Ramamohan Paturi, and Stefan Schneider. 0-1 integer linear programming with a linear number of constraints. *Electronic Colloquium on Computational Complexity (ECCC)*, 21:24, 2014. URL: `http://eccc.hpi-web.de/report/2014/024`.

18 Russell Impagliazzo, William Matthews, and Ramamohan Paturi. A satisfiability algorithm for ac^0. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 961–972, 2012. URL: `http://portal.acm.org/citation.cfm?id=2095193&CFID=63838676&CFTOKEN=79617016`.

19 Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-sat. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001. URL: `https://doi.org/10.1006/jcss.2000.1727`, `doi:10.1006/jcss.2000.1727`.

20 Russell Impagliazzo, Ramamohan Paturi, and Stefan Schneider. A satisfiability algorithm for sparse depth two threshold circuits. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 479–488, 2013. URL: `https://doi.org/10.1109/FOCS.2013.58`, `doi:10.1109/FOCS.2013.58`.

21 Karthik C. S., Bundit Laekhanukit, and Pasin Manurangsi. On the parameterized complexity of approximating dominating set. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29,*

2018, pages 1283–1296, 2018. URL: `http://doi.acm.org/10.1145/3188745.3188896`, `doi:10.1145/3188745.3188896`.

22 Clemens Lautemann. BPP and the polynomial hierarchy. *Inf. Process. Lett.*, 17(4):215–217, 1983. URL: `https://doi.org/10.1016/0020-0190(83)90044-3`, `doi:10.1016/0020-0190(83)90044-3`.

23 Saburo Muroga. *Threshold logic and its applications.* Wiley, 1971.

24 Joseph Naor and Moni Naor. Small-bias probability spaces: Efficient constructions and applications. *SIAM J. Comput.*, 22(4):838–856, 1993. URL: `https://doi.org/10.1137/0222053`, `doi:10.1137/0222053`.

25 Ilan Newman. Private vs. common random bits in communication complexity. *Inf. Process. Lett.*, 39(2):67–71, 1991.

26 Noam Nisan. The communication complexity of threshold gates. *Combinatorics, Paul Erdos is Eighty*, 1:301–315, 1993.

27 Anup Rao and Amir Yehudayoff. Communication complexity (early draft). URL: `https://homes.cs.washington.edu/~anuprao/pubs/book.pdf`.

28 Aviad Rubinstein. Hardness of approximate nearest neighbor search. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 1260–1268, 2018. URL: `http://doi.acm.org/10.1145/3188745.3188916`, `doi:10.1145/3188745.3188916`.

29 Rainer Schuler. An algorithm for the satisfiability problem of formulas in conjunctive normal form. *J. Algorithms*, 54(1):40–44, 2005. URL: `https://doi.org/10.1016/j.jalgor.2004.04.012`, `doi:10.1016/j.jalgor.2004.04.012`.

30 Michael Sipser. A complexity theoretic approach to randomness. In *STOC*, pages 330–335. ACM, 1983.

31 Amnon Ta-Shma. Explicit, almost optimal, epsilon-balanced codes. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 238–251, 2017. URL: `http://doi.acm.org/10.1145/3055399.3055408`, `doi:10.1145/3055399.3055408`.

32 Ryan Williams. Algorithms and resource requirements for fundamental problems. *PhD Thesis*, 2007.

33 Ryan Williams. New algorithms and lower bounds for circuits with linear threshold gates. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 194–202, 2014. URL: `http://doi.acm.org/10.1145/2591796.2591858`, `doi:10.1145/2591796.2591858`.

34 Virginia Vassilevska Williams. Hardness of easy problems: Basing hardness on popular conjectures such as the strong exponential time hypothesis (invited talk). In *10th International Symposium on Parameterized and Exact Computation, IPEC 2015, September 16-18, 2015, Patras, Greece*, pages 17–29, 2015. URL: `https://doi.org/10.4230/LIPIcs.IPEC.2015.17`, `doi:10.4230/LIPIcs.IPEC.2015.17`.