Sparser Random 3-SAT Refutation Algorithms and the Interpolation Problem

Extended Abstract*

Iddo Tzameret**

The Institute for Interdisciplinary Information Sciences (IIIS)
Tsinghua University, Beijing

Abstract. We formalize a combinatorial principle, called *the 3XOR principle*, due to Feige, Kim and Ofek [12], as a family of unsatisfiable propositional formulas for which refutations of small size in any propositional proof system that possesses the feasible interpolation property imply an efficient *deterministic* refutation algorithm for random 3SAT with n variables and $\Omega(n^{1.4})$ clauses. Such small size refutations would improve the state of the art (with respect to the clause density) efficient refutation algorithm, which works only for $\Omega(n^{1.5})$ many clauses [13]. We demonstrate polynomial-size refutations of the 3XOR principle in resolution operating with disjunctions of quadratic equations with small integer coefficients, denoted R(quad); this is a weak extension of cutting planes with small coefficients. We show that R(quad) is weakly automatizable iff R(lin) is weakly automatizable, where R(lin) is similar to R(quad) but with linear instead of quadratic equations (introduced in [25]). This reduces the problem of refuting random 3CNF with n variables and $\Omega(n^{1.4})$ clauses to the interpolation problem of R(quad) and to the weak automatizability of R(lin).

1 Introduction

In the well known *random 3-SAT model* one usually considers a distribution on formulas in conjunctive normal form (CNF) with m clauses and three literals each, where each clause is chosen independently with repetitions out of all possible $2^3 \cdot \binom{n}{3}$ clauses with n variables (cf. [1]). The *clause density* of such a 3CNF is m/n. When m is greater than cn for sufficiently large c, that is, when the clause density is greater than c, it is known (and easily proved for e.g. $c \geq 5.2$) that with high probability a random 3CNF is unsatisfiable.

A *refutation algorithm* for random kCNFs is an algorithm that receives a kCNF (with sufficiently large clause density) and outputs either "`unsatisfiable`" or "`don't know`"; if the algorithm answers "`unsatisfiable`"

[*] The full version is available at: http://arxiv.org/abs/1305.0948

[**] Supported in part by the National Basic Research Program of China Grant 2011CBA00300, 2011CBA00301, the National Natural Science Foundation of P. R. China; Grants 61033001, 61061130540, 61073174, 61373002.

then the kCNF is required to be indeed unsatisfiable; moreover, the algorithm should output “`unsatisfiable`” with high probability (namely, with probability $1 - o(1)$ over the input kCNFs).

We can view the problem of determining the complexity of (deterministic) refutation algorithms as an average-case version of the **P** vs. **coNP** problem: a polynomial-time refutation algorithm for random kCNFs (for a small enough clause density) can be interpreted as showing that “**P** = **coNP** in the average-case”; while a polynomial-time *nondeterministic* refutation algorithm (again, for a small enough clause density) can be interpreted as “**NP** = **coNP** in the average-case”.

Refutation algorithms for random kCNFs were investigated in Goerdt and Krivelevich [15] and subsequent works by Goerdt and Lanka [16], Friedman, Goerdt and Krivelevich [14], Feige and Ofek [13], Feige [11] and Coja-Oghlan et al. [8] (among other works). For random 3CNFs, the best (with respect to the clause density) polynomial-time refutation algorithm to date works for formulas with at least $\Omega(n^{1.5})$ clauses [13]. On the other hand, Feige, Kim and Ofek [12] considered efficient *nondeterministic* refutation algorithms; namely, short *witnesses* for unsatisfiability of 3CNFs that can be checked for correctness in polynomial-time. They established the current best (again, with respect to the clause density) efficient, alas *nondeterministic*, refutation procedure: they showed that with probability converging to 1 a random 3CNF with n variables and $\Omega(n^{1.4})$ clauses has a witness of size polynomial in n.

Since the current state of the art random 3CNF refutation algorithm works for $\Omega(n^{1.5})$ clauses, while the best nondeterministic refutation algorithm works already for $O(n^{1.4})$, determining whether a deterministic polynomial-time (or even a quasipolynomial-time) refutation algorithm for random 3CNFs with n variables and $\Omega(n^{1.4})$ clauses exists is to a certain extent the frontier open problem in the area of efficient refutation algorithms.

1.1 Results

In this work we reduce the problem of devising an efficient deterministic refutation algorithm for random 3CNFs with $\Omega(n^{1.4})$ clauses to the interpolation problem in propositional proof complexity. For a refutation system $\mathcal{P}$, the *interpolation problem for* $\mathcal{P}$ is the problem that asks, given a $\mathcal{P}$-refutation of an unsatisfiable formula $A(x, y) \wedge B(x, z)$, for x, y, z mutually disjoint sets of variables, and an assignment α for x, to return 0 or 1, such that if the answer is 0 then $A(\alpha, y)$ is unsatisfiable and if the answer is 1 then $B(\alpha, z)$ is unsatisfiable. If the interpolation problem for a refutation system $\mathcal{P}$ is solvable in time $T(n)$ we say that $\mathcal{P}$ has *interpolation in time* $T(n)$.[1] When $T(n)$ is a polynomial we say that $\mathcal{P}$ has *feasible interpolation*. The notion of feasible interpolation was proposed in [18] and developed further in [27,6,19].

[1] We do not distinguish in this paper between proofs and refutations: proof systems prove tautologies and refutation systems refute unsatisfiable formulas (or, equivalently prove the negation of unsatisfiable formulas).

We present a family of unsatisfiable propositional formulas, ***denoted*** Υ_n and called *the 3XOR principle formulas*, expressing a combinatorial principle, such that for any given refutation system $\mathcal{P}$ that admits short refutations of Υ_n, solving efficiently the interpolation problem for $\mathcal{P}$ provides an efficient *deterministic* refutation algorithm for random 3CNFs with $\Omega(n^{1.4})$ clauses. In other words, we have the following:

Theorem 1. *If there exists a propositional proof system $\mathcal{P}$ that has interpolation in time $T(n)$ and that admits $s(n)$-size refutations of Υ_n, then there is a deterministic refutation algorithm for random 3CNF formulas with n variables and $\Omega(n^{1.4})$ clauses that runs in time $T(s(n))$. In particular, if $\mathcal{P}$ has feasible interpolation and admits polynomial-size refutations of Υ_n then the refutation algorithm runs in polynomial-time.*

The argument is based on the following: we show that the *computationally hard part* of the Feige, Kim and Ofek nondeterministic refutation algorithm (namely, the part we do not know how to efficiently compute deterministically) corresponds to a disjoint **NP**-pair. Informally, the pair $(\mathbf{A}, \mathbf{B})$ of disjoint **NP** sets is the following: **A** is the set of 3CNFs that have a certain combinatorial property, that is, they contain a collection of sufficiently many *inconsistent even k-tuples*, as defined by Feige et al. (see Definition 2); and **B** is the set of 3CNFs with m clauses for which there exists an assignment that satisfies more than $m - \ell$ clauses as 3XORs (for ℓ a certain function of the number of variables n).

Theorem 1 then follows from the known relation between disjoint **NP**-pairs and feasible interpolation [26,24]: in short, if **A** and **B** are two disjoint **NP** sets and $A(x, y)$ and $B(x, z)$ are the two polynomial-size Boolean formulas corresponding to **A** and **B**, respectively (i.e., for all x, there exists a short y such that $A(x, y) = 1$ iff $x \in \mathbf{A}$; and similarly for **B**), then short refutations of $A(x, y) \wedge B(x, z)$ imply a polynomial-size algorithm that separates **A** from **B**. For more on the relation between disjoint **NP**-pairs and propositional proof complexity see, e.g., [24,3].

In general, we observe that every efficient refutation algorithm (deterministic or not) corresponds directly to a disjoint **NP**-pair as follows: every efficient refutation algorithm is based on some property P of CNFs that can be witnessed (or better, found) in polynomial-time. Thus, every efficient refutation algorithm corresponds to a family of formulas $P(x) \rightarrow \neg\mathrm{SAT}(x)$, expressing that if the input CNF has the property P then x is unsatisfiable; thus, $P(x)$ and $\mathrm{SAT}(x)$ are two disjoint **NP** predicates. In the case of the refutation algorithm of Feige, Kim and Ofek, $P(x)$ expresses simply that the 3CNF x has the Feige et al. witness. *However, the disjoint* **NP***-pair* $(\mathbf{A}, \mathbf{B})$ *we work with is not of this type.* Namely, **A** is not the predicate $P(x)$ for the full Feige, Kim and Ofek witnesses, rather a specific combinatorial predicate (mentioned above) that is only one ingredient in the definition of the Feige et al. witness; and **B** is not $\mathrm{SAT}(x)$. This saves us the trouble to formalize and prove in a weak propositional proof system the full Feige et al. argument (such a formalization was done recently in [22]; see Sec. 1.2 for a comparison with [22]).

In the second part of this paper we reduce the problem of determinizing the Feige et al. nondeterministic refutation algorithm to the interpolation problem of a concrete and apparently weak refutation system. Specifically, we demonstrate polynomial-size refutations for Υ_n in a refutation system denoted R(quad) that extends both cutting planes with small coefficients[2] (cf. [9,6,23]) and Res(2) (for any natural k, the system Res(k) is resolution that operates with kDNFs instead of clauses, introduced by Krajíček [20]). We note also that R(quad) is a subsystem of TC^0-Frege.

An R(quad) refutation (see full version [28] for the definition) over the variables $\{x_1, \ldots, x_n\}$ operates with *disjunctions* of quadratic equations, where each quadratic equation is of the form:

$$\sum_{i,j\in[n]} c_{ij}x_ix_j + \sum_{i\in[n]} c_ix_i + c_0 = a,$$

in which all c_i, c_{ij} and a are integers written in unary. The system R(quad) has the following derivation rule, which can be viewed as a generalized resolution rule: from two disjunctions of quadratic equations $\bigvee_i L_i \vee (L = a)$ and $\bigvee_j L_j \vee (L' = b)$ one can derive:

$$\bigvee_i L_i \vee \bigvee_j L_j \vee (L - L' = a - b).$$

We also add axioms that force our variables to be $0, 1$. An R(quad) refutation of an unsatisfiable set S of disjunctions of quadratic equations is a sequence of disjunctions of quadratic equations (called *proof-lines*) that terminates with $1 = 0$, and such that every proof-line is either an axiom, or appears in S, or is derived from previous lines by the derivation rules.

We show the following:

Theorem 2. R(quad) *admits polynomial-size refutations of the 3XOR principle formulas Υ_n.*

This polynomial upper bound on the refutation size of the 3XOR principle is non-trivial because the encoding of the 3XOR formula is complicated in itself and further the refutation system is very restrictive.

By Theorem 1, we get the reduction from determinizing Feige et al. work to the interpolation problem for R(quad). In other words:

Corollary 1. *If* R(quad) *has feasible interpolation then there is a deterministic polynomial-time refutation algorithm for random 3CNFs with n variables and $\Omega(n^{1.4})$ clauses.*

Next we reduce the problem of determinizing the Feige et al. refutation algorithm to the weak automatizability of a weaker system than R(quad), namely R(lin), as explained in what follows.

[2] A refutation in *cutting planes with small coefficients* is a restriction of cutting planes in which all intermediate inequalities are required to have coefficients bounded in *value* by a polynomial in n, where n is the size of the formula to be refuted (see [6]).

The concept of automatizability, introduced by Bonet, Pitassi and Raz [7] (following the work of [21]), is central to proof-search algorithms. The *proof-search problem* for a refutation system $\mathcal{P}$ asks, given an unsatisfiable formula τ, to find a $\mathcal{P}$-refutation of τ. A refutation system $\mathcal{P}$ is *automatizable* if for any unsatisfiable τ the proof-search problem for $\mathcal{P}$ is solvable in time polynomial in the smallest $\mathcal{P}$-refutation of τ (or equivalently, if there exists a polynomial-time algorithm that on input τ and a number m in unary, outputs a $\mathcal{P}$-refutation of τ of size at most m, in case such a refutation exists). Following Atserias and Bonet [3], we say that a refutation system $\mathcal{P}$ is *weakly automatizable* if there exists an automatizable refutation system $\mathcal{P}'$ that polynomially simulates $\mathcal{P}$. Note that if $\mathcal{P}$ is not automatizable, it does *not* necessarily follow that also $\mathcal{P}'$ is not automatizable. Hence, from the perspective of proof-search algorithms, weak automatizability is a more natural notion than automatizability (see [24] on this).

In [25], the system R(lin) was introduced which is similar to R(quad), except that all equations are *linear* instead of quadratic. In other words, R(lin) is resolution over linear equations with small coefficients. We show the following:

Theorem 3. R(quad) *is weakly automatizable iff* R(lin) *is weakly automatizable.*

The proof of this theorem follows a similar argument to Pudlák [24]. Since weak automatizability of a proof system implies that the proof system has feasible interpolation [7,24], we obtain the following:

Corollary 2. *If* R(lin) *is weakly automatizable then there is a deterministic refutation algorithm for random 3CNFs with n variables and $\Omega(n^{1.4})$ clauses.*

1.2 Consequences and relations to previous work

The key point of this work is the relation between constructing an efficient refutation algorithm for the clause density $\Omega(n^{0.4})$ and proving upper bounds in weak enough propositional proof systems for the 3XOR principle (namely, proof systems possessing feasible interpolation); as well as establishing such upper bounds in relatively weak proof systems.

There are two ways to view our results: either as **(i)** proposing an approach to improve the current state of the art in refutation algorithms via proof complexity upper bounds; or conversely as **(ii)** providing a *new kind* of important computational consequences that will follow from feasible interpolation and weak automatizability of weak proof systems. Indeed, the consequence that we provide is of a different kind from the group of important recently discovered algorithmic-game-theoretic consequences shown by Atserias and Maneva [4], Huang and Pitassi [17] and Beckmann, Pudlák and Thapen [5]. In what follows we explain these two views in more details.

(i) Our results show that by proving that R(quad) has feasible interpolation or by demonstrating a short refutation of the 3XOR principle in some refutation system that admits feasible interpolation, one can advance the state of the art in refutation algorithms. We can hope that if feasible interpolation of R(quad) does not hold, perhaps interpolation in quasipolynomial-time holds (either for R(quad) or for any other system admitting short refutations of the 3XOR principle), which would already improve exponentially the running time of the current best deterministic refutation algorithm for 3CNFs with $\Omega(n^{1.4})$ clauses, since the current algorithm works in time $2^{O(n^{0.2}\log n)}$ [12].

As mentioned above, R(quad) is a common extension of Res(2) and cutting planes with small coefficients (though it is apparently not the weakest such common extension because already R(lin) polynomially simulates both Res(2) and cutting planes with small coefficients). Whether Res(2) has feasible interpolation (let alone, interpolation in quasi-polynomial time) is open and there is no conclusive evidence for or against it. Note that by Atserias and Bonet [3], Res(2) has feasible interpolation iff resolution is weakly automatizable. However this does not necessarily constitute strong evidence against the feasible interpolation of Res(2), because the question of whether resolution is *weakly* automatizable is itself open, and there is no strong evidence ruling out a positive answer to this question.[3] Similarly, there is no strong evidence that rules out the possibility that cutting planes is weakly automatizable.

(ii) Even if our suggested approach is not expected to lead to an improvement in refutation algorithms, it is still interesting in the following sense. The fact that R(quad) has short refutations of the 3XOR principle provides new evidence that (weak extensions of) Res(2) and cutting planes with small coefficients may not have feasible interpolation, or at least that it would be highly non-trivial to prove they do have feasible interpolation; the reason for this is that establishing feasible interpolation for such proof systems would entail quite strong algorithmic consequences, namely, a highly non-trivial improvement in refutation algorithms. This algorithmic consequence adds to other recently discovered and important algorithmic-game-theoretic consequences that would follow from feasible interpolation of weak proof systems.

Specifically, in recent years several groups of researchers discovered connections between feasible interpolation and weak automatizability of small depth Frege systems to certain game-theoretic algorithms: Atserias and Maneva [4] showed that solving *mean payoff games* is reducible to the weak automatizability of depth-2 Frege (equivalently, Res(n)) systems and to the feasible interpolation of depth-3 Frege systems (actually, depth-3 Frege where the bottom fan-in of formulas is at most two). Subsequently, Huang and Pitassi [17] showed that if depth-3 Frege system is weakly automatizable, then *simple stochastic games* are solvable in polynomial time. Finally, Beckmann, Pudlák and Thapen [5] showed that weak automatizability of resolution implies a polynomial-time algorithm for the *parity game*.

[3] It is known that, based on reasonable hardness assumptions from parameterized complexity, resolution is not *automatizable* by Alekhnovich and Razborov [2], which is, as the name indicates, a stronger property than *weak automatizability*.

Comparison with Müller and Tzameret [22]. In [22] a polynomial-size TC^0-Frege proof of the correctness of the Feige et al. witnesses was shown. However the goal of [22] was different from the current paper. In [22] the goal was to construct short propositional refutations for random 3CNFs (with sufficiently small clause density). Accordingly, the connection to the interpolation problem was not made in [22]; and further, it is known by [7] that TC^0-Frege does not admit feasible interpolation (under cryptographical assumptions). On the other hand, the current paper aims to demonstrate that certain short refutations will have *algorithmic* consequences (for refutation algorithms). Indeed, since we are not interested here to prove the correctness of the full Feige et al. witnesses, we are isolating the computationally hard part of the witnesses from the easy (polytime computable) parts, and formalize the former part (i.e., the 3XOR principle) as a propositional formula in a way that is suitable for the reduction to the interpolation problem.

One advantage of this work over [22] is that Theorem 2 gives a more concrete logical characterization of parts of the Feige et al. witnesses (because the proofs in [22] were conducted indirectly, via a general translation from first-order proofs in bounded arithmetic), and this characterization is possibly *tighter* (because R(quad) is apparently strictly weaker than TC^0-Frege).

Organization of the extended abstract. In the preliminaries we review some necessary background from proof complexity and refutation algorithms. In Sec. 3 we describe the connection between feasible interpolation and refutations algorithms. Due to lack of space, readers who wish to read the full details involved in the short R(quad) refutations of the 3XOR principle, as well as the reduction to weak automatizability of R(lin), are referred to the full version available on-line [28].

2 Preliminaries

We usually assume that a 3CNF has n variables $X = \{x_1, \ldots, x_n\}$ and m clauses.

2.1 Disjoint NP-pairs and feasible interpolation of propositional proofs

A *disjoint **NP**-pair* is simply a pair of languages in **NP** that are disjoint. Let L, N be a disjoint **NP**-pair such that $R(x,y)$ is the corresponding relation for L and $Q(x,z)$ is the corresponding relation for N; namely, there exists polynomials p, q such that $R(x,y)$ and $Q(x,z)$ are polynomial-time relations where $x \in L$ iff $\exists y, |y| \le p(|x|) \wedge R(x,y) = \texttt{true}$ and $x \in N$ iff $\exists z, |z| \le q(|x|) \wedge Q(x,z) = \texttt{true}$.

Since both polynomial-time relations $R(x,y)$ and $Q(x,z)$ can be converted into a family of polynomial-size Boolean circuits, they can be written as a family of polynomial-size (in n) CNF formulas. Thus, let $A_n(\overline{x}, \overline{y})$ be a polynomial-size CNF in the variables $\overline{x} = (x_1, \ldots, x_n)$ and $\overline{y} = (y_1, \ldots, y_\ell)$, that is true iff $R(\overline{x}, \overline{y})$ is true, and let $B_n(\overline{x}, \overline{z})$ be a polynomial-size CNF in the variables $\overline{x}$ and $\overline{z} = (z_1, \ldots, z_m)$, that is true iff $Q(\overline{x}, \overline{z})$ is true (for some ℓ, m that are polynomial in n). For every $n \in \mathbb{N}$, we define the following unsatisfiable CNF formula in three mutually disjoint vectors of variables $\overline{x}, \overline{y}, \overline{z}$:

$$F_n := A_n(\overline{x}, \overline{y}) \wedge B_n(\overline{x}, \overline{z}). \tag{1}$$

Note that because $\overline{y}$ and $\overline{z}$ are disjoint vectors of variables and $A_n(\overline{x}, \overline{y}) \wedge B_n(\overline{x}, \overline{z})$ is unsatisfiable, it must be that given any $\overline{x} \in \{0,1\}^n$, either $A_n(\overline{x}, \overline{y})$ or $B_n(\overline{x}, \overline{z})$ is unsatisfiable (or both).

A *propositional proof system* $\mathcal{P}$ is a polynomial-time relation $V(\pi, \tau)$ such that for every propositional formula τ, τ is a tautology iff there exists a binary string π with $V(\pi, \tau) = \texttt{true}$. A propositional proof system $\mathcal{P}$ *polynomially-simulates* another propositional proof system $\mathcal{Q}$ if there is a polynomial-time computable function f that maps $\mathcal{Q}$-proofs to $\mathcal{P}$-proofs of the same tautologies.

Consider a family of unsatisfiable formulas $F_n := A_n(\overline{x}, \overline{y}) \wedge B_n(\overline{x}, \overline{z})$, $i \in \mathbb{N}$, in mutually disjoint vectors of variables, as in (1). We say that the Boolean function $f(\overline{x})$ is *the interpolant of* F_n if for every n and every assignment $\overline{\alpha}$ to $\overline{x}$:

$$\begin{array}{lcll} f(\overline{\alpha}) = 1 & \Longrightarrow & A_n(\overline{\alpha}, \overline{y}) & \text{is unsatisfiable; and} \\ f(\overline{\alpha}) = 0 & \Longrightarrow & B_n(\overline{\alpha}, \overline{z}) & \text{is unsatisfiable.} \end{array} \tag{2}$$

Note that L (as defined above) is precisely the set of those assignments $\overline{\alpha}$ for which $A(\overline{\alpha}, \overline{y})$ is satisfiable, and N is precisely the set of those assignments $\overline{\alpha}$ for which $B(\overline{\alpha}, \overline{z})$ is satisfiable, and L and N are disjoint by assumption, and so $f(\overline{x})$ *separates* L from N.

Definition 1 (Interpolation property). *A propositional proof system $\mathcal{P}$ is said to have the* interpolation property in time $T(n)$ *if the existence of a size $s(n)$ $\mathcal{P}$-refutation of a family F_n as in (1) above implies the existence of an algorithm computing $f(\overline{x})$ in $T(s(n))$ time. When a proof system $\mathcal{P}$ has the interpolation property in time* poly(n) *we say that $\mathcal{P}$ has the* feasible interpolation property*, or simply that $\mathcal{P}$ has* feasible interpolation*.*

Definition 2 (Inconsistent even k-tuple (Feige et al. [12]). *An* even k-tuple *is a tuple of k many 3-clauses in which every variable appears an even number of times. An* inconsistent even k-tuple *is an even k-tuple in which the total number of negative literals is odd.*

Note that for any even k-tuple, k must be an even number (since by assumption the total number of variable occurrences $3k$ is even). The following is the combinatorial principle, due to Feige et al. [12] that we consider in this work:

The 3XOR Principle 1 *Let K be a 3CNF over the variables X. Let S be t inconsistent even k-tuples from K, such that every clause from K appears in at most d inconsistent even k-tuples in S. Then, given any Boolean assignment to the variables X, the number of clauses in K that are unsatisfied by the assignment as 3XOR is at least $\lceil t/d \rceil$.*

The correctness of the 3XOR principle follows directly from the following proposition (the proof of which follows by counting modulo 2) and the fact that every clause in K appears in at most d even k-tuples in S:

Proposition 1 ([12]). *For any inconsistent even k-tuple (over the variables X) and any Boolean assignment A to X, there must be a clause in the k-tuple that is unsatisfied as 3XOR.*

3 From short proofs to refutation algorithms

In this section we demonstrate that polynomial-size proofs of (encodings of the) 3XOR principle in a proof system that has the feasible interpolation property yield deterministic polynomial-time refutation algorithms for random 3CNF formulas with $\Omega(n^{1.4})$ clauses.

3.1 The witness for unsatisfiability

The Feige, Kim and Ofek nondeterministic refutation algorithm [12] is based on the existence of a polynomial-size witness of unsatisfiability for most 3CNF formulas with sufficiently large clause to variable ratio. The witness has several parts, but as already observed in [12], apart from the t inconsistent even k-tuples (Def. 2), all the other parts of the witness are known to be computable in polynomial-time. In what follows we define the witnesses for unsatisfiability.

Let K be a 3CNF with n variables $x_1, \ldots, x_n$ and m clauses. The *imbalance* of a variable x_i is the absolute value of the difference between the number of its positive occurrences and the number of its negative occurrences. The *imbalance of* K is the sum over the imbalances of all variables, in K, denoted $I(K)$. We define $M(K)$ to be an $n \times n$ rational matrix M as follows: let $i, j \in [n]$, and let d be the number of clauses in K where x_i and x_j appear with different signs and s be the number of clauses where x_i and x_j appear with the same sign. Then $M_{ij} := \frac{1}{2}(d - s)$. In other words, for each clause in K in which x_i and x_j appear with the same sign we add $\frac{1}{2}$ to M_{ij} and for each clause in K in which x_i and x_j appear with different signs we subtract $\frac{1}{2}$ from M_{ij}. Let λ be a rational approximation of the biggest eigenvalue of $M(K)$. We shall assume that the additive error of the approximation is $1/n^c$ for a constant c independent of n; i.e., $|\lambda - \lambda'| \leq 1/n^c$, for λ' the biggest eigenvalue of $M(K)$; see [22].

Definition 3 (FKO witness). *Given a 3CNF K, the* FKO witness for the unsatisfiability of K *is defined to be the following collection:*

1. *the imbalance $I(K)$;*
2. *the matrix $M(K)$ and the (polynomially good) rational approximation λ of its largest eigenvalue;*
3. *a collection S consisting of $t < n^2$ inconsistent even k-tuples such that every clause in K appears in at most d many even k-tuples, for some positive natural k;*
4. *the inequality $t > \frac{d \cdot (I(K) + \lambda n)}{2} + o(1)$ holds.*

(The $o(1)$ above stands for a specific rational number b/n^c, for c and b constants independent of n).

Theorem 4 ([12]). *There are constants c_0, c_1 such that for a random 3CNF K with n variables and $\Omega(n^{1.4})$ clauses, with probability converging to 1 as n tends to infinity there exist natural numbers k, t, d such that $t = \Omega(n^{1.4})$ and*

$$k \le c_0 \cdot n^{0.2} \quad \text{and} \quad t < n^2 \quad \text{and} \quad d \le c_1 \cdot n^{0.2}, \tag{3}$$

and K has a witness for unsatisfiability as in Definition 3.

Inspecting the argument in [12], it is not hard to see that it is sufficient to replace part 3 in the witness with a witness for the following:

> *3'. No assignment can satisfy more than $m - \lceil t/d \rceil - 1$ clauses in K as 3XORs.*

Therefore, since $I(K)$, $M(K)$ and λ are all polynomial-time computable (see [12] for this), in order to determinize the nondeterministic refutation algorithm of [12] it is sufficient to provide an algorithm that almost surely determines (correctly) that part 3' above holds (when also t and d are such that part 4 in the witness holds). In other words, in order to construct an efficient refutation algorithm for random 3CNFs (with $\Omega(n^{1.4})$ clauses) it is sufficient to have a deterministic algorithm A that on every input 3CNF (and for t and d such that part 4 in the witness holds) answers either "`condition 3' is correct`" or "`don't know`", such that A is never wrong (i.e., if it says "`condition 3' is correct`" then condition 3' holds) and with probability $1 - o(1)$ over the input 3CNFs A answers "`condition 3' is correct`". Note that we do not need to actually find the Feige et al. witness nor do we need to decide if it exists or not (it is possible that condition 3' holds but condition 3 does not, meaning that there is *no* Feige et al. witness). The relation between unsatisfiability and bounding the number of clauses that can be satisfied as 3XOR in a 3CNF was introduced by Feige in [10] (and used in [13] as well as in [12]).

3.2 The disjoint NP-pair corresponding to the 3XOR principle

We define the corresponding *3XOR principle disjoint* ***NP****-pair* as the pair of languages (L, N), where k, t, d are natural numbers given in *unary*:

$$\begin{aligned} L := \{\langle X, k, t, d\rangle \,\big|\, & X \text{ is a 3CNF with } n \text{ variables and Equation (3) holds} \\ & \text{for } k, t, d \text{ and there exist } t \text{ inconsistent even } k\text{-tuples such that} \\ & \text{each clause of } X \text{ appears in no more than } d \text{ many } k\text{-tuples}\}, \end{aligned}$$

$$\begin{aligned} N := \big\{\langle X, k, t, d\rangle \,\big|\, & X \text{ is a 3CNF with } n \text{ variables and } m \text{ clauses and} \\ & \text{Equation (3) holds for } k, t, d \text{ and there exists an assignment} \\ & \text{that satisfies at least } m - \lceil t/d \rceil \text{ clauses in } X \text{ as 3XOR}\big\}. \end{aligned}$$

It is easy to verify that both L and N are indeed **NP** sets and that, by the 3XOR principle, $L \cap N = \emptyset$.

Using the same notation as in Section 2.1, we denote by $R(x, y)$ and $Q(x, z)$ the polynomial-time relations for L and N, respectively. Further, for every $n \in$

$\mathbb{N}$, there exists an *unsatisfiable* CNF formula in three mutually disjoint sets of variables $\overline{x}, \overline{y}, \overline{z}$:

$$\Upsilon_n := A_n(\overline{x}, \overline{y}) \wedge B_n(\overline{x}, \overline{z}), \tag{4}$$

where $A_n(\overline{x}, \overline{y})$ and $B_n(\overline{x}, \overline{z})$ are the CNF formulas expressing that $R(x, y)$ and $Q(x, z)$ are true for x of length n, respectively.

Theorem 1. *Assume that there exists a propositional proof system that has interpolation in time $T(n)$ and that admits size $s(n)$ refutations of Υ_n. Then, there is a deterministic refutation algorithm for random 3CNF formulas with $\Omega(n^{1.4})$ clauses running in time $T(s(n))$.*

Proof. By the assumption, and by the definition of the feasible interpolation property, there exists a deterministic polynomial-time interpolant algorithm A that on input a 3CNF K and three natural numbers k, t, d given in unary, if $\mathsf{A}(K, k, t, d) = 1$ then $\langle K, k, t, d\rangle \notin L$ and if $\mathsf{A}(K, k, t, d) = 0$ then $\langle K, k, t, d\rangle \notin N$.

The desired refutation algorithm works as follows: it receives the 3CNF K and for each 3-tuple of natural numbers $\langle k, t, d\rangle$ for which Equation (3) holds it runs $\mathsf{A}(K, k, t, d)$. Note there are only $O(n^3)$ such 3-tuples. If for one of these runs $\mathsf{A}(K, k, t, d) = 0$ then we know that $\langle K, k, t, d\rangle \notin N$; in this case we check (in polynomial-time) that the inequality in Part 4 of the FKO witness (Definition 3) holds, and if it does, we answer "`unsatisfiable`". Otherwise, we answer "`don't know`".

The correctness of this algorithm stems from the following two points:

(i) If we answered "`unsatisfiable`", then there exist k, t, d such that $\langle K, k, t, d\rangle \notin N$ and Part 4 in the FKO witness holds, and so Condition 3' (from Section 3.1) is correct, and hence, by the discussion in 3.1, K is unsatisfiable.

(ii) For almost all 3CNFs we will answer "`unsatisfiable`". This is because almost all of them will have an FKO witness (by Theorem 4), which means that $\langle K, k, t, d\rangle \in L$ for some choice of $t < n^2, d, k$ (in the prescribed ranges) and hence the interpolant algorithm A must output 0 in at least one of these cases (because $\mathsf{A}(K, k, t, d) = 1$ means that $\langle K, k, t, d\rangle \notin L$).

Acknowledgments

I wish to thank Jan Krajíček for very useful comments related to this work as well as Albert Atserias and Neil Thapen for useful related discussions. Thanks also to the anonymous reviewers who helped a lot to improve the exposition.

References

1. D. Achlioptas. Random satisfiability. In *Handbook of Satisfiability*, 245–270. 2009.
2. M. Alekhnovich and A. A. Razborov. Resolution is not automatizable unless W[P] is tractable. *SIAM J. Comput.*, 38(4):1347–1363, 2008.
3. A. Atserias and M. L. Bonet. On the automatizability of resolution and related propositional proof systems. *Information and Computation*, 189:182–201, 2004.
4. A. Atserias and E. Maneva. Mean-payoff games and propositional proofs. In *Int. Conf. Aut., Lang. Prog. (ICALP)*, 102–113. Springer Berlin / Heidelberg, 2012.

5. A. Beckmann, P. Pudlák, and N. Thapen. Parity games and propositional proofs. *ACM Transactions on Computational Logic*. To appear.
6. M. L. Bonet, T. Pitassi, and R. Raz. Lower bounds for cutting planes proofs with small coefficients. *The Journal of Symbolic Logic*, 62(3):708–728, 1997.
7. M. L. Bonet, T. Pitassi, and R. Raz. On interpolation and automatization for Frege systems. *SIAM J. Comput.*, 29(6):1939–1967, 2000.
8. A. Coja-Oghlan, A. Goerdt, and A. Lanka. Strong refutation heuristics for random k-SAT. *Combinatorics, Probability & Computing*, 16(1):5–28, 2007.
9. W. Cook, C. R. Coullard, and G. Turan. On the complexity of cutting plane proofs. *Discrete Applied Mathematics*, 18:25–38, 1987.
10. U. Feige. Relations between average case complexity and approximation complexity. In *STOC*, pages 534–543, 2002.
11. U. Feige. Refuting smoothed 3CNF formulas. In *Proceedings of the IEEE 48th Annual Symposium on Foundations of Computer Science*, pages 407–417. IEEE Computer Society, 2007.
12. U. Feige, J. H. Kim, and E. Ofek. Witnesses for non-satisfiability of dense random 3CNF formulas. In *Proceedings of the IEEE 47th Annual Symposium on Foundations of Computer Science*, 2006.
13. U. Feige and E. Ofek. Easily refutable subformulas of large random 3CNF formulas. *Theory of Computing*, 3(1):25–43, 2007.
14. J. Friedman, A. Goerdt, and M. Krivelevich. Recognizing more unsatisfiable random k-SAT instances efficiently. *SIAM J. Comput.*, 35(2):408–430, 2005.
15. A. Goerdt and M. Krivelevich. Efficient recognition of random unsatisfiable k-SAT instances by spectral methods. In *Annual Symposium on Theoretical Aspects of Computer Science*, pages 294–304, 2001.
16. A. Goerdt and A. Lanka. Recognizing more random unsatisfiable 3-SAT instances efficiently. *Electronic Notes in Discrete Mathematics*, 16:21–46, 2003.
17. L. Huang and T. Pitassi. Automatizability and simple stochastic games. In *ICALP (1)*, pages 605–617, 2011.
18. J. Krajíček. Lower bounds to the size of constant-depth propositional proofs. *The Journal of Symbolic Logic*, 59(1):73–86, 1994.
19. J. Krajíček. Interpolation theorems, lower bounds for proof systems, and independence results for bounded arithmetic. *Jour. Symb. Logic*, 62(2):457–486, 1997.
20. J. Krajíček. On the weak pigeonhole principle. *Fund. Math.*, 170(1-2):123–140, 2001.
21. J. Krajíček and P. Pudlák. Some consequences of cryptographical conjectures for S^1_2 and EF. *Inform. and Comput.*, 140(1):82–94, 1998.
22. S. Müller and I. Tzameret. Short propositional refutations for dense random 3CNF formulas. In *Proceedings of the 27th Annual ACM-IEEE Symposium on Logic In Computer Science (LICS)*, 2012.
23. P. Pudlák. Lower bounds for resolution and cutting plane proofs and monotone computations. *The Journal of Symbolic Logic*, 62(3):981–998, Sept. 1997.
24. P. Pudlák. On reducibility and symmetry of disjoint NP pairs. *Theoret. Comput. Sci.*, 295:323–339, 2003.
25. R. Raz and I. Tzameret. Resolution over linear equations and multilinear proofs. *Ann. Pure Appl. Logic*, 155(3):194–224, 2008.
26. A. A. Razborov. On provably disjoint NP-pairs. *ECCC*, 1(6), 1994.
27. A. A. Razborov. Unprovability of lower bounds on circuit size in certain fragments of bounded arithmetic. *Izv. Ross. Akad. Nauk Ser. Mat.*, 59(1):201–224, 1995.
28. I. Tzameret. Sparser random 3-SAT refutation algorithms and the interpolation problem. 2014. http://arxiv.org/abs/1305.0948.