DOI: 10.1145/3635117

BY VENKATESAN GURUSWAMI, PRAVESH K. KOTHARI, AND PETER MANOHAR

New Spectral Algorithms for Refuting Smoothed k-SAT

Despite being a quintessential example of a hard problem, the quest for finding fast algorithms for deciding satisfiability of propositional formulas has occupied computer scientists both in theory and in practice. In this article, we survey recent progress on designing algorithms with strong refutation guarantees for *smoothed* instances of the k-SAT problem. Smoothed instances are formed by slight random perturbations of arbitrary instances, and their study is a way to bridge the gap between worst-case and average-case models of problem instances. Our methods yield new algorithms for smoothed k-SAT instances with guarantees that match those for the significantly simpler and well-studied model of *random* formulas. Additionally, they have led to a novel and unexpected line of attack on some longstanding extremal combinatorial problems in graph theory and coding theory. As an example, we will discuss the resolution of a 2008 conjecture of Feige on the existence of short cycles in hypergraphs.

1. Introduction

The famous SAT problem asks to determine if a given propositional formula is *satisfiable*. That is, can we set the formula's variables to 0 (False) or 1 (True) in a way so that the formula evaluates to 1 (True). In this article, we will focus on the k-SAT problem where the propositional formula is further restricted to be in the k-CNF form, that is, logical AND of a collection of k-*clauses*, each of which is a logical OR of at most k *literals* (variables or their logical negations). For example, $(x_1 \vee \neg x_2 \vee x_3) \wedge (x_2 \vee x_4 \vee \neg x_5)$ is a 3-CNF formula in variables $x_1, x_2, \ldots, x_5$ where $\vee$, $\wedge$, and $\neg$ denote the logical AND, OR and NOT operations, respectively. Given a k-CNF formula, we are interested in either finding a satisfying truth assignment, if it exists, or a "refutation"—a short, easily-checkable proof that the formula is unsatisfiable. Despite its simplicity, k-SAT is phenomenally expressive and models a long list of important discrete optimization problems. A decades-long quest has thus focused on designing algorithms for k-SAT in both theory and practice.

In this article, we will focus on finding refutations—"obviously verifiable" polynomial size (that is, short) contradictions that confirm unsatisfiability of a k-SAT formula. For any formula, we can simply a tabulate each of the 2^n possible truth assignments x together with a clause violated by x. This is an obviously verifiable refutation but clearly too long—it's exponential in size. On the other hand, if we get lucky and our formula happens to contain two 1-clauses $(x_1) \wedge (\neg x_1)$, then it is manifestly unsatisfiable and the two 1-clauses serve as an easily verifiable and short certificate of unsatisfiability. Of course, it's unrealistic for such clauses to magically occur in interesting inputs. But we can often infer additional clauses that must also be satisfied if the input formula is satisfied and hope that such an obviously verifiable short contradiction arises in the inferred clauses. Such *clause learning* (via mechanical *resolution* rules) for deriving new clauses form an integral part of practical SAT solvers.

In his famous 1972 paper,[18] Karp proved that ascertaining satisfiability or finding refutations for 3-SAT formulas is NP-hard. Thus, finding a fast algorithm for 3-SAT that succeeds (even approximately) on all possible input is likely hard. One might naturally expect finding refutations to get easier as the number of clauses increases (more clauses means more possibilities for contradictions to manifest), and so perhaps one might hope that *denser* instances get easier? No such luck! As it turns out, unless a widely believed, stronger variant of the $\mathsf{P} \neq \mathsf{NP}$ conjecture fails, there are no polynomial time algorithms for refuting k-SAT formulas unless they have essentially the maximum possible $\approx n^k$ (out of $\approx n^k$ possible) clauses. In fact, even substantially beating brute-force search and finding sub-exponential (for example, $2^{\sqrt{n}}$) time algorithms is ruled out for formulas with $\approx n^{k-1}$ clauses.[14]

Despite the grim picture presented by these hardness results, the extraordinary modeling power of k-SAT has motivated a concerted research effort for finding fast heuristics for k-SAT. On the practical side, new algorithmic ideas along with advances in software engineering have made modern SAT solvers[13] a powerful and indispensable tool with applications to solving practical instances of optimization problems in planning, model checking, and verification of software systems. By encoding the task of finding counter-examples to mathematical conjectures into SAT formulas, SAT solvers have even helped resolve longstanding mathematical conjectures.[6] On the theoretical side, algorithms research has focused on more careful modeling of input instances to escape worst-case hardness under minimal assumptions. Such ***beyond worst-case*** input models for hard discrete optimization problems such as k-SAT now form a vibrant area[11] of research in algorithm design.

1.1 The Smoothed k-SAT Model

In 2007, Feige proposed[9] his *smoothed k-SAT* model as a way to circumvent the all-pervasive hardness of k-SAT. He was inspired by a groundbreaking work of Spielman and Teng on smoothed analysis of the simplex algorithm. The simplex algorithm for linear programming, introduced by Dantzig in 1947, presented an uncomfortable disconnect between theoretical predictions and practical performance—here was a fast practical algorithm that also provably needed exponential time in the worst-case! In 2001, Spielman and Teng convincingly resolved[23] this tension and showed that the simplex method runs in polynomial time on smoothed inputs—an input obtained by adding a small random perturbation to an arbitrary instance. Such a perturbation, of the sort one might reasonably expect practical instances to naturally possess, is enough to remove all hardness in even carefully crafted instances.

Feige's model involves an analogous smoothening of a worst case k-SAT formula by randomly perturbing each literal (that is, changing an unnegated variable to negated and vice-versa) in each clause with some small probability, say 0.01, independently. For example, given two clauses $(x_1 \vee \neg x_2 \vee x_3) \wedge (x_2 \vee x_4 \vee \neg x_5)$, we imagine tossing 6 independent coins with bias 0.01, one for each literal in each of the two clauses. The smoothed version of the first clause, will have, for example, x_1 negated if the first coin lands on heads, x_2 unnegated if the 2nd coin lands on heads, and so on.

If the input formula ϕ has $\gg Cn$ clauses in n variables for some large enough constant C, then the resulting smoothed formula is unsatisfiable with high probability over the random perturbation. Feige thus asked if the task of finding refutations for smoothed k-SAT formulas gets significantly easier when compared to worst-case formulas. Equivalently, given our discussion above, do $\ll n^{k-1}$ clause-smoothed k-SAT formulas admit efficient refutation algorithms?

Prior works[2,8,22] showed that the answer is indeed yes for the *random k-SAT* model—a *fully smoothed* model where the negation patterns and the *clause structure*, that is, the variables appearing in the clauses (that are worst-case in smoothed k-SAT) are chosen independently at random. However, those algorithms strongly exploit the abundant randomness in the choice of variables appearing in the clauses.

In this article, we will survey recent developments[1,16,17] on a new class of algorithms, based on the eigenvalues of certain specialized *Kikuchi matrices* (introduced earlier[24] for statistical inference and to simplify algorithms for random k-SAT[22] for even k), that yield optimal (modulo certain hardness conjectures) algorithms for smoothed k-SAT. As a result, these new algorithms succeed in refuting smoothed k-SAT formulas with $m \gtrsim n^{k/2} \log n$ clauses, that is, $\ll n^{k-1}$, in polynomial time and significantly beat brute-force search if $m \gtrsim n^{1+\epsilon}$. In fact, our guarantees for smoothed k-SAT match the best-known (and conjectured optimal) results for the significantly simpler and restricted setting of *random*k-SAT formulas, provide quantitative bounds on the number of clauses that every truth assignment must violate, and extend far beyond k-SAT to handle *all* logical constraint satisfaction problems.

1.2 Spectral Methods for Combinatorics

While the theoretical advances in algorithms for k-SAT haven't yet influenced practical SAT-solving, they already have some surprising applications to long open problems in other areas of mathematics. These include resolving Feige's conjecture on small *even covers* (cycles) in hypergraphs,[16,17] making progress on the decades-long quest for optimal bounds for *locally decodable*[4] and *locally correctable*[19] error-correcting codes, and problems[7] in additive number theory that generalize the famous Szemeredi's theorem.

The principle behind such applications is analogous to how SAT solvers helped resolve mathematical conjectures by encoding the search for a proof into a SAT formula. Our theoretical analog strongly exploits the newfound ability to tackle k-SAT formulas with a *worst-case* clause structure. In this article, we will discuss an application of this method to proving Feige's conjecture on *small cycles in hypergraphs*. Surprisingly, proving this conjecture will let us go full circle to show even better refutations for smoothed k-SAT.

Short Cycles in Graphs.

Feige's conjecture[10] is a generalization of a basic result about short *cycles* in *graphs*. Recall that a graph (aka network), is simply a collection of pairs, called *edges* (modeling pairwise associations), on n nodes. A cycle in such a network is a sequence of nodes $v_1 \to v_2 \to \dots \to v_\ell \to v_1$ that starts and ends at the same node such that every consecutive pair has an edge between them. In his famous 1978 book,[5] mathematician Béla Bollobás conjectured that every graph with n nodes where every node, on average, has $d > 2$ edges (this quantity is called the *average degree*), must have a cycle of length at most $\approx \log_{d-1}(n)$. When $d = 2$, the network can be a single giant cycle on all n vertices that clearly has no cycle of length $\leq n-1$. For any $d > 2$, however, the conjecture implies that we cannot even avoid a cycle of length $O(\log n)$—an exponentially smaller bound than n—and thus signals a *phase transition* in the extremal length of the smallest cycle as the average degree d crosses 2.

Bollobas's conjecture is an example of a result in *extremal combinatorics*. Such results uncover a truth that holds for *all* (thus *extremal*) mathematical objects. Here, it says that no matter how we might build a graph on n nodes, so long as it has average degree $d > 2$, we cannot avoid introducing a short cycle. In their elegant 2002 paper,[3] mathematicians Alon, Hoory and Linial confirmed this conjecture.

Short Cycles in Hypergraphs.

Feige's conjecture asks a similar question about short cycles in *hypergraphs*. A k-uniform hypergraph is a collection of subsets of size k on n nodes, called *hyperedges*, that model associations between a larger number of nodes (instead of 2 in graphs). A 2-uniform hypergraph is simply a graph. In order to pose Feige's question, we will identify a key property of cycles in graphs and use it to motivate a generalized notion of cycles in hypergraphs. Observe that every vertex appears in either two or zero edges in any cycle in a graph. In particular, a cycle is an *even subgraph*—a subset of edges on which, every vertex appears an even integer number of times. Even subgraphs naturally generalize to hypergraphs. We will define a hypergraph cycle or *even covers* to be a collection $C_1, C_2, \ldots, C_\ell$ of hyperedges such that every node of the hypergraph is included in an even number of C_i's (that is, an *even subhypergraph*). (See Figure 1a and 1b.)

Figure 1a. A length 4 cycle in a graph on 4 vertices.

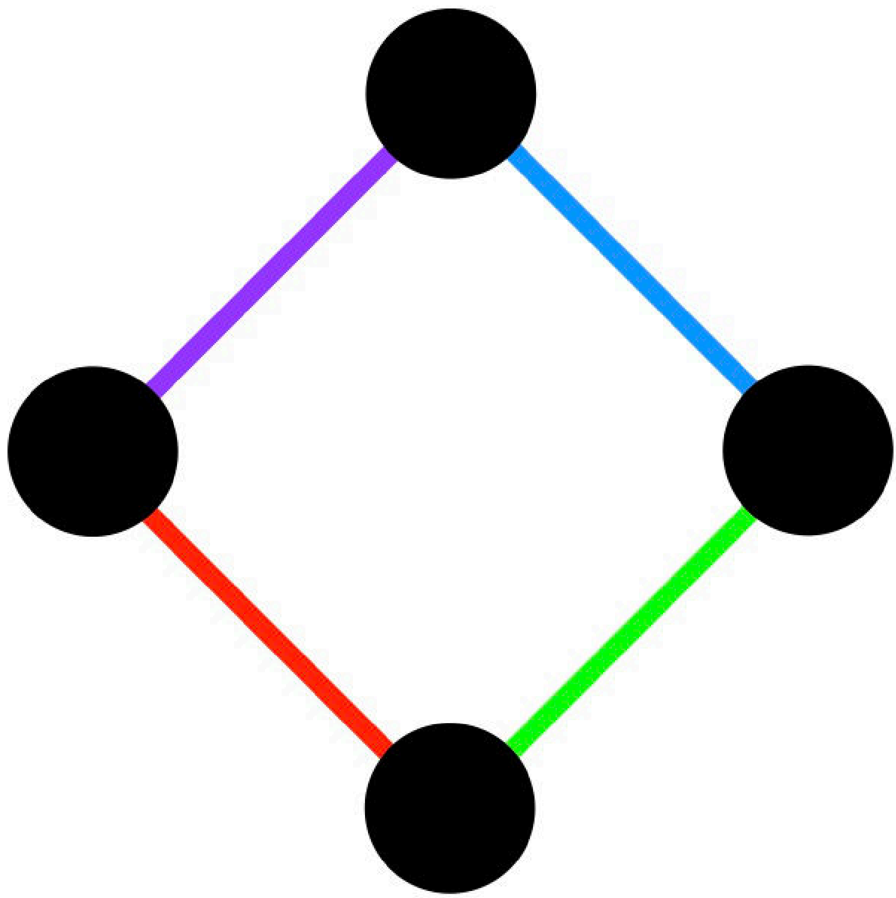
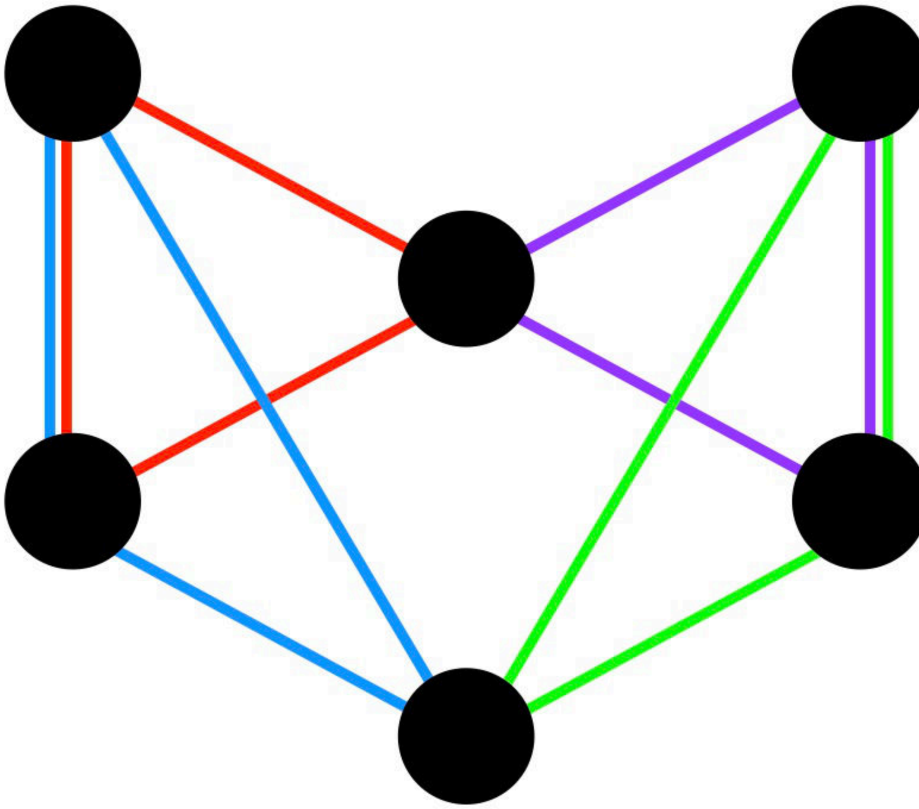


Figure 1b. A length 4 even cover in a 3-uniform hypergraph on 6 vertices.



This definition may look odd (or perhaps a little too even?), at first. A cycle in a graph has an appealing combinatorial structure of a loop. Hypergraph cycles seem to lack a combinatorial meaning. Why did Feige (and why should we) care about it? Feige's motivation for studying short cycles actually stemmed from to finding refutations for k-SAT formulas (see next section). Here, we outline a different source of motivations that comes from deep connections to the theory of error correcting codes because hypergraph cycles naturally relate to solving systems of linear equations.

To see why, let's associate a variable to every node of a graph and let's think of each edge as specifying a linear equation modulo 2. Thus, the edge $1 \sim 2$ between nodes 1 and 2 relates to the equation $x_1 + x_2 = b$ mod 2 where $b \in \{0, 1\}$. A cycle in the graph then naturally corresponds to a subset of 2-sparse (that is, each equation has only two non-zero coefficients) equations that is *linearly dependent*—that is, if you want this subset of equations to be satisfied, then the right hand side b of at least one of the equations is already fixed (that is, cannot be chosen independently) once you fix the choice of the right hand sides for all the others. As a simple example, consider the graph on 3 nodes with edges $1 \sim 2$, $2 \sim 3$ and $3 \sim 1$. Suppose you knew that some x satisfies the two equations corresponding to the first two edges $x_1 + x_2 = 0$ mod 2 and $x_2 + x_3$ is 1 mod 2. Then, adding the left hand sides of the two equations gives $x_1 + 2x_2 + x_3 = 1$ mod 2 which is equivalent to $x_1 + x_3 = 1$ mod 2 since $2x_2 = 0$ mod 2, regardless of the value of x_2. Thus, the right hand side of third equation is determined/dependent and cannot be chosen independently of the first two equations. A cycle in a k-uniform hypergraph similarly corresponds to a linearly dependent subset of k-sparse (that is, each equation has k non-zero coefficients) equations corresponding to each hyperedge.

The length of the smallest cycle thus equals the size of the smallest linearly dependent subset of equations in a given system. Understanding the size of such a set turns out to have a whole gamut of applications, especially in designing *error-correcting codes*. Error correcting codes (or just codes, in short) are a systematic method of adding redundancy to a message so that, when the message transmitted across a noisy channel and incurs errors, one can still *decode* it uniquely thanks to the redundancy. The systematic methods or codes naturally involve adding "parity checks", that is, right hand sides of an appropriately chosen set of linear equations evaluated at the message. In such codes, the length of the smallest linearly dependent subset of equations naturally corresponds to *distance*—a crucial quantity that controls the number of errors that can be corrected. The smallest linear dependencies in k-sparse equations turns out to be equivalent to understanding the best possible distance (and thus, the largest possible rate of errors that can be tolerated) by an important class of codes called *low density parity check* codes introduced by Gallager in 1960s with numerous theoretical and practical applications.

Feige's Conjecture and the Resolution.

As in the case of graphs, we are interested in the extremal trade-off between average degree (that is, average number of hyperedges containing a node) and the length of the smallest hypergraph cycle in a k-uniform hypergraph. Given the connection to linear dependencies above and the basic fact that every collection of $m = n + 1$ equations in n variables are linearly dependent, whenever the average degree $d = mk/n > k$, then the hypergraph must have a (rather long) cycle of length $n + 1$. Making an analogy to graphs, one might expect that if $d \gg k$, the hypergraph must have a $O(\log n)$-length cycle, but this turns out to be false! Mathematicians Assaf Naor and Jacques Verstraete in 2006 showed[21] that one needs (and this is enough!) the average degree $d \gtrsim n^{k/2-1}$ in order for every hypergraph to have a $O(\log n)$-length cycle. For $k = 2$, this matches a coarse version of the bound for graphs but for $k \geq 3$ suggests a new regime between $d \approx 1$ and $d \approx \sqrt{n}$ that has no analog for graphs. What happens when, for example, $d \approx n^{0.25}$? Feige conjectured a precise behavior for this regime:

Conjecture 1 (Feige[10]).
Every k-uniform hypergraph on n nodes with average degree $\approx (n/\ell)^{k/2-1}$ has an $\ell \log n$-length cycle.

Feige's conjecture (up to some $\log n$ factors in d) was motivated by the hypothesis (and is, in fact, equivalent to it) that there is no better construction of hypergraphs avoiding short cycles than simply choosing a random hypergraph. It is thus analogous to the famous 1947 theorem of Shannon (the birthplace of modern coding theory!) that random error correcting codes are "best" in a precise sense.

Despite being a foundational statement about hypergraphs, the conjecture remained largely open with only some partial progress for a special case by Alon and Feige in 2009. Now, by invoking the new algorithms for solving smoothed k-SAT we can essentially completely resolve it — up to one additional $\log n$ factor in the degree bound! This was established first by the authors with additional $\log n$ factors which were later[17] trimmed down to a single $\log n$ factor.

Theorem 2.
Every k-uniform hypergraph on n nodes with average degree $\approx (n/\ell)^{k/2-1} \log n$ has an $\ell \log n$-length cycle.

As we will explain later in this article, the proof of this theorem makes a new connection between the success of our spectral approach for smoothed k-SAT and existence of short cycles in hypergraphs. It forms the first (of a growing list of) application spectral refutations via Kikuchi matrices in combinatorics.

2. A New Spectral Approach

Our approach for finding refutations for smoothed k-SAT formulas relies on continuous time algorithms based on *spectral methods*—methods that use eigenvalues and eigenvectors of matrices built from the input. This is in contrast to the largely discrete algorithmic toolkit (such as resolution refutations and their generalization) in modern practical SAT solvers. In fact, it has been known for more than 20 years that even random k-SAT formulas with $\ll n^{k-1}$ clauses do not admit efficient resolution refutations—a natural formalization of combinatorial refutations.

The spectral approach for refuting k-SAT formulas was conceived[15] by Goerdt and Krivilevich back in 2001. A decade and half of work led to spectral methods with conjectured-optimal guarantees for refuting random k-SAT in 2017. These spectral methods, however, are rather brittle and strongly rely on the randomness in the variables appearing in each clause. For smoothed k-SAT, such methods provably fail since the variables appearing in the clauses are completely arbitrary. In this article, we will present a new class of spectral methods, based on *Kikuchi* matrices, which, when combined with combinatorial pre-processing, provide *robust* methods for refutation that significantly simplify the results for random k-SAT and extend to smoothed k-SAT with no loss in performance.

From k-SAT to Degree k Polynomials.

To bring in spectral methods, we will make a simple but conceptually important translation between refuting a k-SAT formula and finding certificates of upper bounds on the maximum value of a degree k polynomial. For this purpose, it will be more convenient to view truth assignments as $+1$ (True) and -1 (False). Given a k-SAT formula ϕ with m clauses, let $\Phi:\{-1,1\}^n \to \mathbb{N}$ map truth assignments $x \in \{-1,1\}^n$ to the number of clauses satisfies by x. Then, $\Phi(x)$ is clearly the sum of m functions Φ_C, one for each clause C in ϕ where $\Phi_C(x) = 1$ if and only if x satisfies clause C. Since Φ_C depends only on $k\{\pm 1\}$-variables, it is a degree k polynomial. For example, $k=3$ and $C = (x_1 \vee x_2 \vee x_3)$:

$$\Phi_C(x) = \frac{7}{8} + \frac{1}{8}\left(x_1 + x_2 + x_3 - x_1x_2 - x_2x_3 - x_3x_1 + x_1x_2x_3\right) \tag{1}$$

To refute the k-SAT formula ϕ, we will find an easily-checkable certificate of the fact $\Phi(x) = \sum_C \Phi_C(x) < m$ for all x. In fact, we will certify a stronger bound of $\Phi(x) \leq 0.99m$, that is, every x must violate not just 1 but in fact 1% of the clauses.

Observe that Φ_C is a sum of 8 (in general, 2^k) monomials, each of degree ≤ 3 (k, more generally). A standard idea, going back to early 2000s, is to certify bounds on 8 different polynomials, obtained by taking one out of 8 terms corresponding to each C. We can then obtain a bound on $\Phi(x)$ by adding all the 8 quantities. For each such polynomial obtained from a smoothed 3-SAT formula, with a little more work that we omit here, we can also assume that the coefficients of each monomial is an independent, uniform $\{\pm 1\}$. We thus focus on strong refutation of *semirandom homogeneous polynomials* $\Psi(x)$ of the form:

$$\Psi(x) = \sum_{C \in H} b_C \prod_{i \in C} x_i, \tag{2}$$

where H, the *instance hypergraph*, is simply the collection of m different sets $C \subseteq [n]$ of size k (corresponding to each original clause) and $b_C \in \{\pm 1\}$ are chosen uniformly and independently for each $C \in H$. Here, strong refutation involves certifying that $\Psi(x) \leq \epsilon m$ for a sufficiently tiny $\epsilon > 0$.

2.1 From Quadratic Polynomials to Matrices

Let us show how spectral methods show up by starting with the simplest setting of $k=2$. Then, each $C \in H$ is a set of size 2, or simply a pair $\{i,j\} \subseteq [n]$, and $\Psi(x)$ from (2) is a degree 2 polynomial in x. The idea is to view such a degree 2 polynomial as a *quadratic form*.

For an $n \times n$ matrix A, its quadratic form on a vector v equals $v^\top Av = \langle v, Av\rangle = \sum_{i,j \leq n} v_i v_j A(i,j)$. This expression is a homogeneous quadratic polynomial in v. Indeed, every quadratic polynomial is a quadratic form of an associated matrix and vice-versa. For our Ψ, let the $n \times n$ matrix A be defined by:

$$A(i,j) = \begin{cases} b_{\{i,j\}} & \text{if } C = \{i,j\} \in H \\ 0 & \text{otherwise} \end{cases}.$$

Then, for any $x \in \{-1,1\}^n$, $x^\top Ax = \sum_{i,j} x_i x_j b_{i,j} = 2\Psi(x)$.

A basic result in linear algebra allows upper-bounding any quadratic form of A as:

$$v^\top Av \leq \|v\|_2^2 \|A\|_2,$$

where $\|v\|_2 = \sqrt{\sum_i v_i^2}$ is the length of the vector v and $\|A\|_2$ is the "spectral norm" or the largest *singular value* of A. Since x has $\{\pm 1\}$-coordinates and thus length $\sqrt{n}$, we obtain that $\Psi(x) \leq n\|A\|_2$.

This bound on $\Psi(x)$ is easily verifiable. Given Ψ (obtained easily from the k-SAT formula ϕ), we form the matrix A and use a linear time algorithm (called *power iteration*) to obtain good estimates on the spectral norm $\|A\|_2$.

To certify $\Psi(x) \leq \epsilon m$, we need to check that $\|A\|_2 \leq \epsilon m/n$. A is an example of a *random matrix* since its entries $b_{\{i,j\}}$ are uniformly and independently distributed in $\{\pm 1\}$. There is a well-developed theory for understanding the typical value of $\|A\|_2$ for such random matrices that allows us to conclude that $\|A\|_2 \lesssim \sqrt{\Delta_{max} \log n}$ where Δ_{max} is the maximum number of non-zero entries in any row of the matrix A. If the pairs $C = \{i,j\}$ are "equidistributed" that is, any variable i participates in roughly the same number of pairs, then we would expect $\Delta_{max} \approx \Delta_{avg} \approx m/n$ where Δ_{avg} is the average number of non-zero entries in a row of A. Thus, $\|A\|_2 \lesssim \sqrt{m \log n/n}$ which is $\leq \epsilon m/n$ if $m \gtrsim n \log n$.

What if the set of pairs H is not *regular* and some i is "over-represented" in the set of pairs C? While we omit formal details, it turns out that one can use an elegant reweighting trick (discovered in the work of[17]) on the matrix A that effectively allows us to assume regularity if $\Delta_{avg} \gg 1$.

2.2 Generalizing to Quartic Polynomials

The case of odd k turns out to be technically challenging so let us skip $k = 3$ and generalize the above approach when $\Psi(x)$ is of degree $k = 4$ (that is, the case of 4-SAT). So, $\Psi(x)$ is not quadratic in x. We will now view it as a quadratic form in $\binom{n}{2}$ variables each corresponding to quadratic monomials $x_i x_j$ in the original assignment x. Let us write $x^{\circ 2}$ for the vector in $\mathbb{R}^{\binom{n}{2}}$ indexed by pairs $\{i,j\}$ with entry at $\{i_1, i_2\}$ given by $x_{i_1} x_{i_2}$. Define the $\binom{n}{2} \times \binom{n}{2}$ matrix A:

$$A(\{i_1,i_2\},\{j_1,j_2\}) := \begin{cases} b_{i_1,i_2,j_1,j_2} & \text{if } C = \{i_1,i_2,j_1,j_2\} \in H \\ 0 & \text{otherwise.} \end{cases}$$

Then, as before, we can observe that:

$$\left(x^{\circ 2}\right)^\top A\left(x^{\circ 2}\right) = \sum_{\{i,j\},\{k,\ell\}} x_i x_j x_k x_\ell A(\{i,j\},\{k,\ell\}) = 6\Psi(x)$$

The factor 6 comes from the fact that there $\binom{4}{2} = 6$ different ways that a set C of size 4 can be broken into pairs of pairs each of which arises as a term in the quadratic form above.

We can now write

$$\Psi(x) \leq \|x^{\circ 2}\|_2^2 \|A\|_2 = \binom{n}{2} \|A\|_2.$$

And a similar appeal to results in random matrix theory tells us that with high probability $\|A\|_2 \lesssim \sqrt{\Delta_{max} \log \binom{n}{2}} \lesssim \sqrt{\Delta_{max} \log n}$ where Δ_{max} is the maximum number of non-zero entries in any row of A. Equivalently, Δ_{max} is the maximum number of 4-clauses that a pair $\{i,j\}$ participates in. When all pairs behave roughly similarly, we will have $\Delta_{max} \approx \Delta_{avg} \lesssim \frac{m}{\binom{n}{2}}$, in which case

$$\Psi(x) \leq \binom{n}{2} \sqrt{\frac{m}{\binom{n}{2}} \log n} \leq \epsilon m$$

with high probability if $m \gtrsim n^2 \log n$.

Early proofs of such facts used different tools from random matrix theory and worked for random 4-SAT by utilizing that H is a random collection of 4-sets in that case. Our approach here explicitly reveals that only an equi-distribution (that is, $\Delta_{max} \approx \Delta_{avg}$) property of H is required for the success of this approach. This allows us, via a similar reweighting trick (that succeeds if $\Delta_{avg} \gg 1$) discussed above, to obtain a result that works for arbitrary (worst-case) hypergraphs.

Let's finish this part by noting our quantitative bounds. For $k = 2$, our refutation succeeded when $m \gtrsim n \log n$. For $k = 4$, we instead needed $m \gtrsim n^2 \log n$. Indeed, for arbitrary even k, a similar argument yields a bound of $m \gtrsim n^{k/2} \log n$—a significant improvement over the $\Omega(n^k)$ clauses required for refuting an unsatisfiable k-SAT formula in the worst-case, showing us the power of the spectral approach.

2.3 Beyond Basic Spectral Refutations

A smoothed k-SAT formula is unsatisfiable with high probability whenever it has $m \gtrsim n$ clauses. But our spectral refutations above require $m \gtrsim n^2 \log n$—a bound higher by a factor $\approx n$ (and $n^{k/2-1}$ for arbitrary even k). This is because our approach fails whenever the average number of entries in a row of A, that is, $\Delta_{avg} = m/\binom{n}{2}$, is $\ll 1$. The question of whether there are non-trivial refutations for k-SAT formulas when $m \ll n^{k/2}$ (now called the *spectral threshold*) remained open for more than a decade and half. In the same time, researchers found evidence, in the form of restricted lower bounds,[20] that there may be no polynomial time refutation algorithm for $m \ll n^{k/2}$. This, nevertheless, left open the possibility of significantly beating brute-force search below this threshold. This possibility was realized for *random*k-SAT in 2017. We will now discuss a significantly simpler (described essentially in full below!) spectral approach that succeeds even for smoothed k-SAT.

We will continue to work with $k = 4$. As before, we will write $\Psi(x)$ as a quadratic form but instead of the natural matrices we discussed above, we will use ***Kikuchi*** matrices that we next introduce. First, though, a piece of notation: for sets $S, T \subseteq [n]$, we let $S \oplus T = (S \cup T) \setminus (S \cap T)$ denote the symmetric difference of S and T.

DEFINITION 3 (KIKUCHI MATRICES).
For any $r \in \mathbb{N}$, the level r-Kikuchi matrix for Ψ is a $\binom{n}{r} \times \binom{n}{r}$ matrix with rows and columns indexed by sets $S, T \subseteq [n]$ of size r and entries given by:

$$A(S,T) = \begin{cases} b_C & \text{if } S \oplus T = C \in H \\ 0 & \text{otherwise.} \end{cases}$$

Observe that for $r = 2$, the above Kikuchi matrices specialize to the $\binom{n}{2} \times \binom{n}{2}$ matrix we saw in the previous subsection. Let's see why $\Psi(x)$ is a quadratic form of A.

For any assignment $x \in \{\pm 1\}^n$, denote by $x^{\circ r}$ the $\binom{n}{r}$-dimensional vector with coordinates indexed by sets $S \subseteq [n]$ of size r and S-th entry given by $x_S = \prod_{i \in S} x_i$. Then, for $D = \binom{4}{2}\binom{n-4}{r-2}$ we have:

$$\begin{aligned} (x^{\circ r})^\top A(x^{\circ r}) &= \sum_{S,T} x_S x_T A(S,T) \\ &= \sum_{C \in H} b_C \sum_{S,T : S \oplus T = C} x_C = D\Psi(x). \end{aligned}$$

Here, we used that since $S \oplus T = C$, for any $x \in \{\pm 1\}^n$, $x_S \cdot x_T = x_{S \cup T \setminus S \cap T} x_{S \cap T}^2 = x_C$ as $x_i^2 = 1$ for every i. The last equality holds true because the number of pairs (S,T) such that S, T are r-size sets and $S \oplus T = C$ is exactly D for any set C of size 4. Observe how our notational trick of switching to $\{\pm 1\}$-valued truth assignments paid off here.

Given this simple observation, we can now again construct the spectral upper bound $\Psi(x) \leq \|x^{\circ r}\|_2^2 \|A\|_2 = \binom{n}{r} \|A\|_2$. Furthermore, it turns out that powerful tools of random matrix theory still allow us to conclude as before that

$$\|A\|_2 \lesssim \sqrt{\Delta_{max} \log \binom{n}{r}} = \sqrt{\Delta_{max} r \log n}. \tag{3}$$

The Kikuchi superpower: Density Increment.

Why might this upper bound be better? The meat is in the *density increment*. As r grows, the number of rows in A grow. But the number of non-zero entries in A grow even faster, giving us a net increase in Δ_{avg}. Indeed, let $m = n^2 \log n/\ell$ for some parameter $\ell \in \mathbb{N}$. Then, since each $C \in H$ contributes D non-zero entries, $\Delta_{avg} = mD/\binom{n}{r} \approx \left(n^2 \log n/\ell\right)\left(r^2/n^2\right) \approx r^2 \log n/\ell$. In particular, even when $\ell \gg \log n$ (and thus we have $m = n^2 \log n/\ell \ll n^2$ clauses) choosing r large enough still allows us to obtain a $\Delta_{avg} \gg 1$.

Surprisingly, the rest of the proof idea is more or less the same as before! Let us assume, as we did at first in both the previous subsections, that all rows of A have roughly equal number of non-zero entries. Such a condition holds true if H is a random collection of sets of size 4. Then, $\Delta_{max} \lesssim \Delta_{avg} \approx r^2 \log n/\ell$. Plugging this back in (3) gives

$$\Psi(x) \leq \frac{\binom{n}{r}}{D}\|A\|_2 \lesssim \frac{m}{\Delta_{avg}}\sqrt{\Delta_{avg} r \log n} \leq \epsilon m$$

if $\Delta_{avg} \gtrsim \sqrt{\Delta_{avg} r \log n}/\epsilon$ or $\Delta_{avg} \gtrsim \frac{r \log n}{\epsilon^2} \log n$. Since $\Delta_{avg} = r^2 \log n/\ell$, this condition holds if $r \geq \ell/\epsilon^2$.

Furthermore, as in the previous two subsections, we can use a variant of our reweighting trick to generalize this argument to arbitrary H without any further assumptions. To verify this bound algorithmically (that is, to "check" our refutation), we need to construct the matrix A and compute its spectral norm. This requires a run time proportional to the dimension of the matrix which scales as $\approx n^r$. So, all in all, we obtain a roughly n^{ℓ/ϵ^2} time algorithm to certify that $\Psi(x) \leq \epsilon m$ whenever $m \geq n^2/\ell$ for any $\ell \in \mathbb{N}$. When $m \approx n^{1+\delta}$ for some small $\delta > 0$, that is, even slightly superlinear, the runtime of our algorithm is strictly sub-exponential (specifically $\approx 2^{n^{1-\delta}}$) and thus asymptotically beats brute-force search.

Handling odd k: We described our approach so far for even k. The case of odd k turns out to be a little more involved. This has been true for spectral algorithms ever since the earliest spectral algorithms for the problem. The polynomial time case (for example, when $m \gtrsim n^{1.5} \log n$ for 3-SAT analogous to $m \gtrsim n^2 \log n$ for 4-SAT) were first found in a work of Abascal, Guruswami, Kothari in 2021. The full trade-off required introducing the correct generalizations of the Kikuchi matrices that we have described above. The analysis of the spectral norms of such matrices requires more effort and some additional combinatorial ideas.

We will not formalize these ideas here but note the following eventual result that we derive as a consequence:

THEOREM 4.
For any $k \in \mathbb{N}$ and $\ell \in \mathbb{N}$, given a semi-random homogeneous degree k polynomial $\Psi(x)$ with $m \geq n(n/\ell)^{k/2-1} \log n$ non-zero coefficients, there is a $2^{O(\ell \log n/\epsilon^2)}$ time spectral algorithm that certifies $\Psi(x) \leq \epsilon m$. Consequently, for any k, we can refute smoothed k-SAT formulas with m clauses also in time $2^{O(\ell \log n/\epsilon^2)}$.

3. Proving Feige's Conjecture

Let us now see how our spectral algorithms for smoothed k-SAT provides a resolution for Feige's conjecture. Our approach can be thought of as a theoretical equivalent of encoding the search for a proof into (un)-satisfiability of a SAT formula and then running a practical SAT solver. Given a hypergraph H, we will build a SAT formula Ψ_{random} that will be *satisfiable* if H does not have a short cycle. We will then prove that Ψ_{random} is in fact unsatisfiable to complete our proof. Of course, instead of using the computer to find such a refutation, *we* will "find" them (that is, argue their existence) analytically by appealing to our spectral algorithms.

Let us now describe this our argument in more detail. We are given an arbitrary k-uniform hypergraph H on n nodes and average degree $d \approx (n/\ell)^{k/2-1} \log n$. Our goal is to show that H must have an $\ell \log n$-length cycle. Starting from H, we will define a family of homogeneous degree k polynomials:

$$\Psi_{\text{sat}} = \sum_{C \in H} b_C x_C.$$

Observe that Ψ_{sat} is clearly ***satisfiable***. Indeed, if $x_i = 1$ for every i then $\Psi_{\text{sat}}(x) = |H|$, the maximum possible value.

Our key claim below will argue, using the analysis of our spectral algorithm from above, that if H has no short cycle then Ψ_{sat} must in fact be unsatisfiable:

LEMMA 5 (KEY CLAIM).
If H has no $\approx (\ell \log n)/\epsilon^2$ length cycle, then, $\max_{x \in \{\pm 1\}^n} \Psi_{\text{sat}}(x) \leq \epsilon |H|$.

We thus immediately hit a contradiction unless H has a $\approx \ell \log n$ length cycle.

Let us now discuss why Lemma 5 must hold for even k. For $r \in \mathbb{N}$, we let A be the Kikuchi matrix for the polynomial Ψ_{sat} that we built in the previous section:

$$A_{\text{sat}}(S, T) = \begin{cases} 1 & \text{if } S \oplus T \in H \\ 0 & \text{otherwise} \end{cases}.$$

Then, we have: $\Psi_{\text{sat}}(x) \leq \binom{n}{r} \|A_{\text{sat}}\|_2$.

We will now argue a rather odd-looking fact. Consider the polynomial Ψ defined below for arbitrary $b_C \in \{\pm 1\}$:

$$\Psi_{\mathbf{b}} = \sum_{C \in H} b_C x_C,$$

We also let Ψ_{random} be the special case when b_Cs are chosen uniformly at random and independently. Notice that Ψ_{random} has the same form as the polynomial Ψ we analyzed in the previous section. Let $A_{\mathbf{b}}$ be the Kikuchi matrix built from $\Psi_{\mathbf{b}}$:

$$A_{\mathbf{b}}(S, T) = \begin{cases} b_C & \text{if } S \oplus T = C \in H \\ 0 & \text{otherwise} \end{cases}.$$

We will argue that if H had no $\log \binom{n}{r} \approx \ell \log n$-length cycle, then $\|A_{\mathbf{b}}\|_2 \approx \|A_{\text{sat}}\|_2$ *no matter what the value of b_C's are*. Now, from the previous section, we know that $\binom{n}{r} \|A_{\mathbf{b}}\|_2 \leq \epsilon |H|$ for random $\mathbf{b}$ thus for every x:

$$\Psi_{\text{sat}}(x) \leq \binom{n}{r} \|A_{\text{sat}}\|_2 \leq \binom{n}{r} \|A_{\mathbf{b}}\|_2 \leq \epsilon |H|.$$

This claim can appear strange. How can it be that $\|A_{\mathbf{b}}\|_2$ does not depend on the b_C's at all? In a sense, our proof reveals how short cycles in H are "necessary" for $\|A_{\mathbf{b}}\|_2$ to be as small as it is in the previous section!

Trace Moments and Spectral Norms.

To relate $\|A_{\text{sat}}\|_2$ and $\|A_{\text{random}}\|_2$ and to bring in the cycles in H, we will utilize a classical connection between $\|B\|_2$ and the so-called *trace moments* of a matrix B that, in turn, are related to a certain combinatorial count of *walks* on B.

For any $N \times N$ symmetric matrix B, let $\|B\|_2 = \sigma_1 \geq \sigma_2 \geq \cdots \sigma_N \geq 0$ be its N singular values placed in descending order. The *trace* $\text{tr}(B)$ is simply the sum of the diagonal elements of B. For any even $2t \in \mathbb{N}$, a classical observation in linear algebra says that the trace of the $2t$-th power of B equals the sum of $2t$-th powers of its singular values:

$$\text{tr}\left(B^{2t}\right) = \sigma_1^{2t} + \sigma_2^{2t} + \cdots + \sigma_N^{2t}.$$

This is helpful because we can now write:

$$\|B\|_2^{2t} = \sigma_1^{2t} \leq \mathrm{tr}\left(B^{2t}\right) \leq \sum_{i=1}^{n} \sigma_i^{2t} \leq N\sigma_1^{2t} = N\|B\|_2^{2t}$$

That is, the $2t$-th power of $\|B\|_2$ equals $\mathrm{tr}(B^{2t})$ up to a factor N. By taking $2t = \log N$ and taking $1/2t$-th powers on both sides and recalling that $N^{1/\log N} \to 1$ as $N \to \infty$ gives:

$$\|B\|_2 \leq \mathrm{tr}\left(B^{2t}\right)^{1/2t} = N^{1/\log N}\|B\|_2 \approx \|B\|_2.$$

Thus, for $2t \approx \log N$, $\mathrm{tr}\left(B^{2t}\right)^{1/2t}$ is a faithful approximation to $\|B\|_2$.

Relating $\|A_{\text{sat}}\|_2$ and $\|A_{\text{random}}\|_2$ via Trace Moments.

Using the above connection, we will now focus on arguing that $\mathrm{tr}\left(A_{\text{sat}}^{2t}\right) = \mathrm{tr}\left(A_{\text{random}}^{2t}\right)$ for $2t = \log N = \log\binom{n}{r} \approx r \log n$. This would give us $\|A_{\text{sat}}\|_2 \approx \|A_{\text{random}}\|_2$.

We now recall another classical observation from basic linear algebra that relates trace moments of a matrix B to a certain combinatorial count of "walks" on B. For A (standing for A_{sat} or $A_{\mathbf{b}}$), we thus have:

$$\mathrm{tr}\left(A^{2t}\right) = \sum_{S_1, S_2, \ldots, S_{2t}} A\left(S_1, S_2\right) A\left(S_2, S_3\right) \cdots A\left(S_{2t}, S_1\right). \tag{4}$$

That is, $\mathrm{tr}(A^{2t})$ is the sum over all sequences of $2t$ row indices (that is, subsets of $[n]$ of size r) of product of the entries of A on consecutive elements of the sequence.

Every entry of A is either 0 or ± 1 and for each non-zero entry $A(S_i, S_{i+1})$, $S_i \oplus S_{i+1} = C$ for some $C \in H$. Thus, any $2t$-sequence $\left(S_1, S_2, \ldots, S_{2t}\right)$ that contributes a non-zero (and thus exactly $\prod_{i=1}^{2t} b_{C_i}$ in $\mathrm{tr}(A_{\mathbf{b}}^{2t})$) value to the sum above must correspond to a $2t$-tuple $\left(C_1, C_2, \ldots, C_{2t}\right)$ of hyperedges from H, one for each entry $A(S_i, S_{i+1})$.

More is true about such a $\left(C_1, C_2, \ldots, C_{2t}\right)$, as we next demonstrate in the crucial observation below. Let $\mathbf{1}_{C_i} \in \{0,1\}^n$ be the 0-1 indicator of the set $C_i \subseteq [n]$. That is, $\mathbf{1}_{C_i}(j) = 1$ if and only if $j \in C_i$.

Observation 6.
Any $\left(C_1, C_2, \ldots, C_{2t}\right)$ corresponding to a non-zero term in (4) satisfies $\sum_{i=1}^{2t} \mathbf{1}_{C_i} = 0 \bmod 2$.

This crucial observation is actually quite simple to prove. We know that $\mathbf{1}_{S_i} + \mathbf{1}_{S_{i+1}} = \mathbf{1}_{C_i} \bmod 2$ for every i—since $S_i \oplus S_{i+1} = C_i$. If we add up all the left hand sides we get a sum over $\mathbf{1}_{S_i}$'s where every $\mathbf{1}_{S_i}$ appears exactly twice (since S_i occurs in exactly two entries $A(S_{i-1}, S_i)$ and $A(S_i, S_{i+1})$). Thus the left hand side (and thus also the right hand side) must add up to the 0 vector modulo 2.

Next, notice that the j-th entry $\sum_{i=1}^{2t} \mathbf{1}_{C_i}(j) = 0 \bmod 2$ if and only if j occurs in an even number of C_i's. Thus, the above observation says that the (multi)-set $\{C_1, C_2, \ldots, C_{2t}\}$ is a cycle or an even cover. This appears exciting since we have a direct relationship between $\mathrm{tr}(A_{\text{sat}}^{2t})$ and cycles in H!

There is a crucial snag though—the same C could recur multiple times in $\left(C_1, C_2, \ldots, C_{2t}\right)$. Indeed, if $C_i = C$ for every i or more generally, every C_i appeared in pairs, then, of course every element $j \in [n]$ will occur in an even number of C_i's, for the trivial reason that the C_i's themselves occur in pairs. Let's call such $2t$-tuples *trivial cycles*—that is, the C_i's occur in pairs and thus do not relate to cycles in H.

Now for our endgame. For every trivial cycle, since C_i's appear in pairs, the quantity $\prod_{i=1}^{2t} b_{C_i}$ has even number of copies of b_{C_i} for every b_{C_i}. Since $b_{C_i}^2 = 1$, this quantity must equal 1 *regardless of the C_i's*! Thus, no matter what the b_{C_i}'s, all non-zero terms contribute exactly 1. In particular, $\mathrm{tr}(A_{\text{sat}}^{2t})$ (where all b_{C_i}'s equal 1) must equal $\mathrm{tr}(A_{\mathbf{b}}^{2t})$ regardless of b_{C_i}s.

This finishes our argument, but it's perhaps helpful to summarize it: we related $\|A\|_2$ (for both $A = A_{\text{sat}}$ and $A = A_{\mathbf{b}}$) to $\mathrm{tr}(A^{2t})$. We then related $\mathrm{tr}(A^{2t})$ to a sum over $2t$-tuples $(S_1, S_2, \ldots, S_{2t})$. The non-zero terms in this sum correspond to $\prod_{i=1}^{2t} b_{C_i}$ for $(C_1, C_2, \ldots, C_{2t})$ for $C_i \in H$ such that $\sum \mathbf{1}_{C_i} = 0 \bmod 2$—this step crucially uses the structure of the Kikuchi matrices. If H has no $2t = \log \binom{n}{r}$ length cycles, then in every nonzero term in the sum the C_i's must occur in pairs, in which case we observe that $\prod_{i=1}^{2t} b_{C_i} = 1$ and is thus independent of what the b_{C_i}'s themselves are.

4. Even Smaller Refutations

In the final act of this article, we will come full circle to show how the purely combinatorial Feige's conjecture yields a surprising corollary for refutations for k-SAT. We discussed a spectral algorithm that finds refutations for smoothed k-SAT whenever the number of clauses $m \gtrsim n^{k/2} \log n$. Improving on this spectral threshold even for the substantially specialized setting of random 3-SAT has been open (with accumulating evidence that this maybe impossible) ever since the 2004 work[8] that obtained first such result.

In a surprising twist from 2006, Feige, Kim and Ofek proved[12] that for *random* 3-SAT formulas with $m \gtrsim n^{1.4}$ clauses (significantly short of the spectral threshold of $\approx n^{1.5}$) admit short, polynomial size refutations with high probability. That is, there ***exists*** a polynomial size certificate, based on a clever combination of spectral and combinatorial ideas, which, if given, can easily help convince us of its unsatisfiability. But despite around two decades of efforts, we do not know polynomial time algorithms to find such a certificate. The FKO result forces us to grapple with the possibility that there may be a marked difference between ***existence*** of short certificates for NP-hard problems and efficient algorithms to find them. No such gap is known (or expected!) for worst-case k-SAT, making this a truly average-case phenomenon. And, no such gap is known for any other discrete optimization problem, even in the average case. Indeed, ever since its discovery, FKO has been a one-of-a-kind result with an aura of mystery around it.

Are the mysterious FKO certificates a quirk of the random 3-SAT model? Or should we expect analogs in more general instances? We will now sketch how our results from the previous two sections allow us a surprising corollary:

> COROLLARY 7.
> With high probability, smoothed 3-SAT formulas with $m \gtrsim n^{1.4} \log n$ clauses admit an easily-checkable polynomial size refutation. More generally, a similar result holds for smoothed k-SAT formulas with $n^{k/2-\delta_k}$ clauses where $\delta_k > 0$ depends only on k.

That is, the FKO results extends without any quantitative change to smoothed 3-SAT formulas. The existence of short cycles in hypergraphs plays a major role in obtaining this corollary. Indeed, this was also a principle motivation for Feige's conjecture back in 2008. At the time, since Conjecture 1 was not known, FKO's proof used a sophisticated application of the second moment method from probability theory with rather complicated calculations. Given Theorem 2, our new certificate and its analysis will be simple.

The idea for construction such refutations is quite simple. Our spectral refutation worked by splitting the polynomial Φ into 8 different polynomials of degree ≤ 3 and then refuting each polynomial via our spectral algorithm. As before, we will do the splitting and use the spectral algorithm for all terms of degree ≤ 2, that is, for all *except for the homogeneous degree 3 polynomial* corresponding to the last term in (1) for which, we will use a "combinatorial" method. Importantly, for the degree ≤ 2 terms, our polyomial time spectral algorithm from Section 2.1 only needs $m \gtrsim n/\epsilon^2 \log n$ (instead of $\approx n^{1.5}$) to certify a bound $\leq \epsilon$.

The basic observation behind the combinatorial method is rather simple. Let H be the 3-uniform hypergraph of monomials appearing in Ψ. Let $\{C_1, C_2, \ldots, C_t\}$ be a cycle in H and let $\Psi_{\text{cycle}} = \sum_{i=1}^{t} b_{C_i} x_{C_i}$ be the "fragment" of Ψ that only keeps the monomials corresponding to the cycle. Then, if $\prod_{i=1}^{t} b_{C_i} = -1$

(which happens with probability $1/2$ over the choice of b_{C_i}'s), then, we claim that $\Psi_{\text{cycle}}(x) \le t - 1$ for every x. That is, every x must in fact be at least 1 short of the maximum value of t on such a fragment. Suppose not and say for some x, $\Psi_{\text{cycle}}(x) = t$. Then, $x_{C_i} = b_{C_i}$ for every $1 \le i \le t$. Thus, $\prod_{i=1}^{t} x_{C_i} = \prod_{i=1}^{t} b_{C_i} = -1$. But on the other hand, since $\{C_1, C_2, \ldots, C_t\}$ is a cycle, every j occurs in an even number of C_i's and thus, $\prod_{i=1}^{t} x_{C_i} = \prod_{j=1}^{n} x_j^{(even)} = 1$. This is a contradiction!

Here's how this basic observation helps. Suppose H has $m \approx n^{1.5} \log n/\sqrt{\ell}$ hyperedges. Then, we know from Theorem 2 that H contains a $\approx \ell \log n$ cycle. We can then remove the hyperedges in this cycle and repeatedly find $\ell \log n$ length cycles in the residual hypergraph. This process gives us a "cycle partition" of 99% of hyperedges of H into cycles of length $\approx \ell \log n$. From our argument above, for about $1/2$ of such cycles, the product of the corresponding b_{C_i}'s will turn out to be -1. Let's call such cycles ***violated***. Thus, can write:

$$\Psi = \sum_{\text{violated cycles in partition}} \Psi_{\text{cycle}}(x) + \Psi_{\text{remaining}}$$

For each violated cycle, every x must "lose" at least 1 on the maximum possible value of the polynomial. So, we know that at any x, Ψ must be short of its maximum value by at least the number of violated cycles in the partition. So, $\Psi(x) \le m - O\left(\frac{m}{\ell \log n}\right) = \left(1 - O\left(\frac{1}{\ell \log n}\right)\right)m$. This is a significantly weaker bound than that of our spectral algorithm (ϵm), but is still non-trivial. It is also efficiently verifiable given a list of violated cycles (of size at most $O(n^{1.5})$, so polynomial size).

The corollary follows by combining this combinatorial certificate with the spectral bound on the degree ≤ 2 parts of Φ. The precise parameters are obtained by optimizing ℓ above but we will omit it here.

5. Conclusion

In this article, we surveyed a new class of spectral algorithms based on *Kikuchi* matrices to find refutations, that is, easily verifiable proofs of unsatisfiability, for smoothed k-SAT formulas. The guarantees we obtained were as strong as the best-known (and conjectured optimal) ones for the substantially simpler random k-SAT formulas and substantially surpass the best-possible (assuming standard hardness conjectures) running times for worst-case k-SAT formulas at every clause density. The approach generalizes to yield similar results for all logical constraint satisfaction problems. We also saw the resolution of the 2008 conjecture of Feige on short cycles in hypergraphs as an example application. And as a consequence, we saw how to extend the one-of-a-kind Feige-Kim-Ofek result from random k-SAT formulas to all smoothed k-SAT formulas. Taken together, the results show that, per the current state-of-the-art, smoothed k-SAT is no harder than the substantially simpler random k-SAT formulas for both refutation algorithms and existence of short certificates.

The *Kikuchi matrix method*, the method of proving results by finding spectral refutations for a related propositional formula, coming out this line of work appears to be a promising new attack on problems in combinatorics and coding theory. It is a pleasing theoretical analog of the powerful approach for resolving mathematical problems via practical SAT solvers—a decidedly "computer science" approach to solve problems in mathematics. A few more applications, including making progress on some decades-old problems in the theory of local error correcting codes,[4,19] are now already around and we anticipate more such results in the near future. (See Figure 2a and 2b.)

Figure 2a. runtime vs #clauses for worst-case k-SAT (best possible modulo standard conjectures).

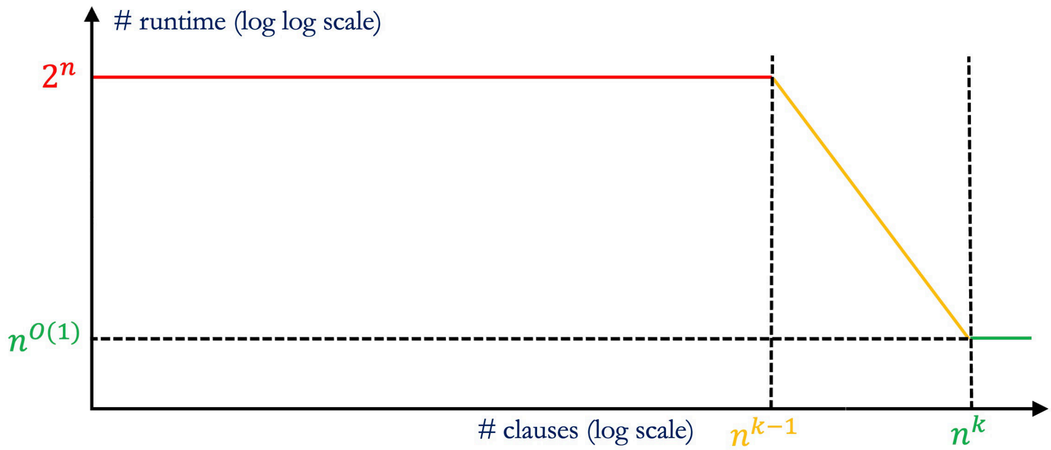
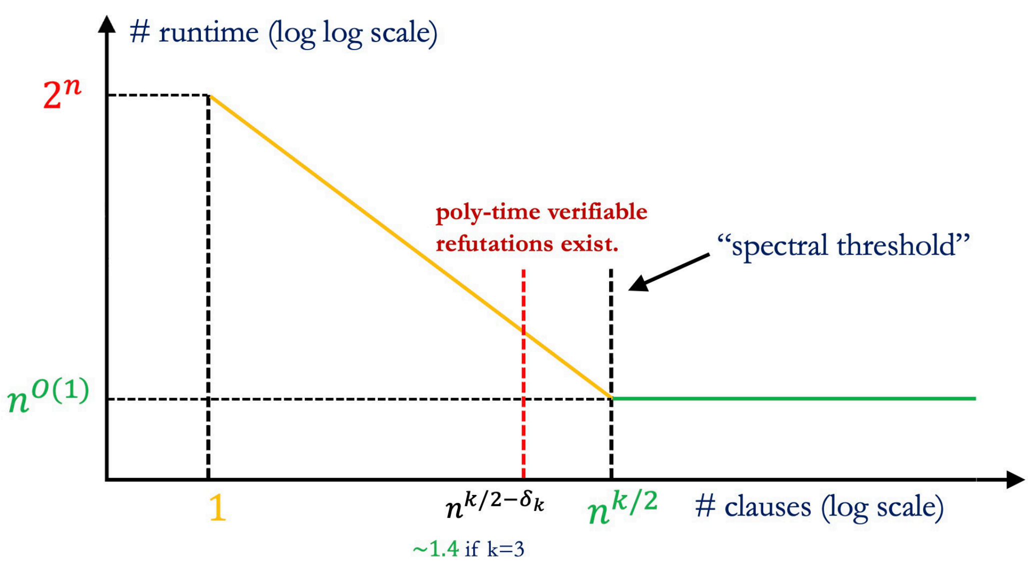


Figure 2b. Runtime vs #clauses for smoothed k-SAT via Kikuchi-based spectral algorithm.



References

1. Abascal, J., Guruswami, V., and Kothari, P.K. Strongly refuting all semi-random boolean csps. In *Proceedings of the 2021 ACM-SIAM Symp. On Discrete Algorithms, SODA 2021, Virtual Conf., January 10 - 13, 2021*. SIAM, 2021, 454–472.
2. Allen, S.R., O'Donnell, R., and Witmer, D. How to refute a random CSP. *Corr, Abs/1505.04383*, 2015.
3. Alon, N., Hoory, S., and Linial, N. The moore bound for irregular graphs. *Graphs and Combinatorics 18* (2002), 53–57.
4. Alrabiah, O., Guruswami, V., Kothari, P.K., and Manohar, P. A near-cubic lower bound for 3-query locally decodable codes from semirandom CSP refutation. In *Proceedings of the 55th Annual ACM Symp. on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*. ACM, 2023, 1438–1448.
5. Bollobás, B. Extremal graph theory. Academic Press, 1978.
6. Brakensiek, J., Heule, M., Mackey, J., and Narváez, D. The resolution of keller's conjecture. In *Automated Reasoning*. Springer International Publishing, Cham, 2020, 48–65.

7. Briët, J. and Castro-Silva, D. On the threshold for Szemerédi's theorem with random differences, 2023.
8. Coja-Oghlan, A., Goerdt, A., and Lanka, A. Strong refutation heuristics for random k-sat. In *8th Intern. Workshop on Randomization and Computation, RANDOM 2004, Cambridge, MA, USA, August 22-24, 2004, Proceedings, volume 3122 of Lecture Notes in Computer Science*. Springer, 2004, 310–321.
9. Feige, U. Refuting smoothed 3cnf formulas. In *48th Annual IEEE Symp. on Foundations of Computer Science (FOCS 2007), October 20-23, 2007, Providence, RI, USA, Proceedings*. IEEE Computer Society, 2007, 407–417.
10. Feige, U. *Small Linear Dependencies for Binary Vectors of Low Weight*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, 283–307.
11. Feige, U. Introduction to semirandom models. In *Beyond the Worst-Case Analysis of Algorithms*. Cambridge University Press, 2020, 189–211.
12. Feige, U., Kim, J.H., and Ofek, E. Witnesses for non-satisfiability of dense random 3cnf formulas. In *47th Annual IEEE Symp. on Foundations of Computer Science (FOCS 2006), 21-24 October 2006, Berkeley, California, USA, Proceedings*. IEEE Computer Society, 2006, 497–508.
13. Fichte, J.K., Berre, D.L., Hecher, M., and Szeider, S. The silent (r)evolution of sat. *Commun. ACM 66*, 6 (may 2023), 64–72.
14. Fotakis, D., Lampis, M., and Paschos, V.T. Sub-exponential approximation schemes for csps: From dense to almost sparse. In *33rd Symp. on Theoretical Aspects of Computer Science, STACS 2016, February 17-20, 2016, Orléans, France, volume 47 of LIPIcs*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016, 37:1–37:14.
15. Goerdt, A. and Krivelevich, M. Efficient recognition of random unsatisfiable k-sat instances by spectral methods. In *STACS 2001, 18th Annual Symp. on Theoretical Aspects of Computer Science, Dresden, Germany, February 15-17, 2001, Proceedings, volume 2010 of Lecture Notes in Computer Science*. Springer, 2001, 294–304.
16. Guruswami, V., Kothari, P.K., and Manohar, P. Algorithms and certificates for boolean CSP refutation: smoothed is no harder than random. In *STOC '22: 54th Annual ACM SIGACT Symp. on Theory of Computing, Rome, Italy, June 20 - 24, 2022*. ACM, 2022, 678–689.
17. Hsieh, J., Kothari, P.K., and Mohanty, S. A simple and sharper proof of the hypergraph moore bound. In *Proceedings of the 2023 ACM-SIAM Symp. On Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*. SIAM, 2023, 2324–2344.
18. Karp, R.M. Reducibility among combinatorial problems. In *Proceedings of a Symp. On the Complexity of Computer Computations, Held March 20-22, 1972, at the IBM Thomas J. W* Plenum Press, New York, 1972, 85–103.
19. Kothari, P.K. and Manohar, P. An exponential lower bound for linear 3-query locally correctable codes. *Corr, Abs/2311.00558*, 2023.
20. Kothari, P.K., Mori, R., O'Donnell, R., and Witmer, D. Sum of squares lower bounds for refuting any CSP. In *Proceedings of the 49th Annual ACM SIGACT Symp. on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*. ACM, 2017, 132–145.
21. Naor, A. and Verstraete, J. Parity check matrices and product representations of squares. *Combinatorica 28* (03 2008), 163–185.
22. Raghavendra, P., Rao, S., and Schramm, T. Strongly refuting random csps below the spectral threshold. *Proceedings of the 49th Annual ACM SIGACT Symp. on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, 2017, 121–131.
23. Spielman, D.A. and Teng, S. Smoothed analysis of algorithms: why the simplex algorithm usually takes polynomial time. In *Proceedings on 33rd Annual ACM Symp. on Theory of Computing, July 6-8, 2001, Heraklion, Crete, Greece*. ACM, 2001, 296–305.
24. Wein, A.S., Alaoui, A.E., and Moore, C. The kikuchi hierarchy and tensor PCA. In *60th IEEE Annual Symp. on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*. IEEE Computer Society, 2019, 1446–1468.

Venkatesan Guruswami (venkatg@berkeley.edu), University of California, Berkeley, CA, USA.
Pravesh K. Kothari (praveshk@cs.cmu.edu), Carnegie Mellon University, Pittsburgh, PA, USA.
Peter Manohar (pmanohar@cs.cmu.edu), Carnegie Mellon University, Pittsburgh, PA, USA.