

THE PRODSAT PHASE OF RANDOM QUANTUM SATISFIABILITY

JOON LEE, NICOLAS MACRIS, JEAN BERNOULLI RAVELOMANANA, PERRINE VANTALON

ABSTRACT. The k -QSAT problem is a quantum analog of the famous k -SAT constraint satisfaction problem. We must determine the zero energy ground states of a Hamiltonian of N qubits consisting of a sum of M random k -local rank-one projectors. It is known that product states of zero energy exist with high probability if and only if the underlying factor graph has a clause-covering dimer configuration. This means that the threshold of the PRODSAT phase is a purely geometric quantity equal to the dimer covering threshold. We revisit and fully prove this result through a combination of complex analysis and algebraic methods based on Buchberger’s algorithm for complex polynomial equations with random coefficients. We also discuss numerical experiments investigating the presence of entanglement in the PRODSAT phase in the sense that product states do not span the whole zero energy ground state space.

1. INTRODUCTION

The quantum version of the k -satisfiability problem (k -QSAT) was introduced by Bravyi [1] as a natural quantum analog of the classical k -SAT constraint satisfaction problem which has been a focal point in classical computer science ever since it was recognized to be NP-complete by Cook and Levin [2], [3]. In classical k -SAT one must determine the satisfiability of a logical formula in conjunctive normal form. Boolean variables $(x_1, \dots, x_N) \in \{0, 1\}^N$ must simultaneously satisfy M constraints in the form of disjunctions of k literals, $\chi_{m_1} \vee \dots \vee \chi_{m_k}$ where $(m_1, m_2, \dots, m_k) \in \{1, \dots, N\}^k$ and $\chi_{m_i} \in \{x_{m_i}, \neg x_{m_i}\}$. We label the constraints as $m \in \{1, \dots, M\}$ and by a slight abuse of notation identify $m \equiv (m_1, \dots, m_k)$. A k -SAT formula is a conjunction of disjunctions $F = \bigwedge_{m=1}^M (\bigvee_{i=1}^k \chi_{m_i})$. The satisfiability of F may be formulated as the study of the set of zero energy assignments for the classical Hamiltonian function

$$(1) \quad h_F(x_1, \dots, x_N) = \sum_{m=1}^M (1 - \mathbb{1}(\chi_{m_1} \vee \dots \vee \chi_{m_k}))$$

In the quantum analog k -QSAT, the Boolean variables are replaced by N quantum bits (or qubits) labelled $\{1, \dots, N\}$ in a collective pure state vector $|\Psi\rangle$ belonging to the Hilbert space $\mathbb{C}_1^2 \otimes \dots \otimes \mathbb{C}_N^2$.¹ The state of the qubits must simultaneously satisfy a set of M quantum constraints labelled by $m \in \{1, \dots, M\}$. Each constraint ensures that $|\Psi\rangle$ is in the kernel of the projector

$$(2) \quad \Pi_m = |\Phi^m\rangle\langle\Phi^m| \otimes I_{N-k}$$

where $|\Phi^m\rangle \in \mathbb{C}_{m_1}^2 \otimes \dots \otimes \mathbb{C}_{m_k}^2$ is a pure state vector of the Hilbert space of the k qubits (m_1, \dots, m_k) involved in the constraint m . The matrix I_{N-k} is the $2^{N-k} \times 2^{N-k}$ identity matrix acting trivially on the remaining $N - k$ qubits not involved in the constraint m . A

joonhyung.lee, nicolas.macris, jean.ravelomanana, perrine.vantalon@epfl.ch.

¹We note that in this quantum model the degrees of freedom are *distinguishable* and hence labelled.

k -QSAT “formula” F is defined by the collection of projectors $\{H_m, m \equiv (m_1, \dots, m_k)\}$. In k -QSAT, the Hamiltonian is given by the $2^N \times 2^N$ matrix

$$(3) \quad H_F = \sum_{m=1}^M H_m.$$

We say that F is satisfiable if the Hamiltonian has zero energy eigenstates, in other words if $\ker(H_F)$ (also called kernel or ground state space) contains a non-trivial vector.

This Hamiltonian represents a natural quantum generalization of the classical cost function (1). First note that each classical disjunction excludes one among 2^k assignments of its k Boolean variables. Analogously each projector (2) excludes one direction in the 2^k dimensional Hilbert space of k qubits. Furthermore, it is not difficult to see that when $|\Phi^m\rangle$'s are tensor products of computational basis vectors of \mathbb{C}^2 , the matrix Hamiltonian (3) reduces to a diagonal matrix (in the computational basis) with diagonal given by values of (1) for all possible 2^N assignments. Of course this is not so anymore when $|\Phi^m\rangle$ is an arbitrary state of $\mathbb{C}_{m_1}^2 \otimes \dots \otimes \mathbb{C}_{m_k}^2$, be it a tensor product or entangled state. Finally it is well known that k -SAT is NP-complete for $k \geq 3$. The analogous statement for k -QSAT is that it is QMA-complete for $k \geq 3$ (see [1], [4] for a discussion).

In this paper we look at the *average case* analysis of k -QSAT. In this formulation the Hamiltonian is taken at random from a set of instances and the problem is to determine the typical behavior of the kernel space. This is perfectly analogous to the random k -SAT problem which studies the typical behavior of the space of zero cost assignments of a random formula. To formulate the problem more precisely we must define an ensemble of random Hamiltonians (or formulas). This is best done in the language of random factor graphs and is beneficial because it turns out that important typical properties are determined by typical geometric properties of these factor graphs. A factor graph is a bipartite graph with labelled “variable nodes” $\{1, \dots, N\}$ (associated to qubits or Boolean variables) and labelled “constraint nodes” $\{1, \dots, M\}$ (associated to projectors or disjunctions). For each constraint node m we pick a k -tuple of variable nodes (m_1, \dots, m_k) uniformly at random among all $\binom{N}{k}$ possible ones, and draw edges $(m, m_1), \dots, (m, m_k)$. This ensemble of random graphs is denoted $G_{N,M}^k$. There is a further level of randomness. In the classical problem once a graph from $G_{N,M}^k$ is chosen, the variables in each disjunction are negated/non-negated with probability one-half (this information is usually encoded in the graph as a dashed/undashed edge). In the quantum problem, once a graph is chosen from $G_{N,M}^k$, one samples the state $|\Phi^m\rangle$ uniformly at random in the (complex) Hilbert space of k qubits (this time this information is not encoded in the graph structure). In practice $|\Phi^m\rangle$ is sampled by generating a 2^k -dimensional complex Gaussian vector with i.i.d $\mathcal{CN}(0, 1)$ components and then is normalized to make it a unit norm.²

The parameter $\alpha = M/N$ is called the constraint density of the ensemble. There is a large literature on the phase diagram of random k -SAT in the thermodynamic limit $N, M \rightarrow +\infty$ with $M/N \rightarrow \alpha$ fixed. Structural and algorithmic phase transitions, as well as their interplay, are largely determined, although many questions remain unanswered (we refer to [5]–[7] for more information). For random k -QSAT the current state of knowledge is more rudimentary and is summarized in figure 1.

² $\mathcal{CN}(0, 1)$ means that real and imaginary parts are independent and distributed as $\mathcal{N}(0, 1/2)$.

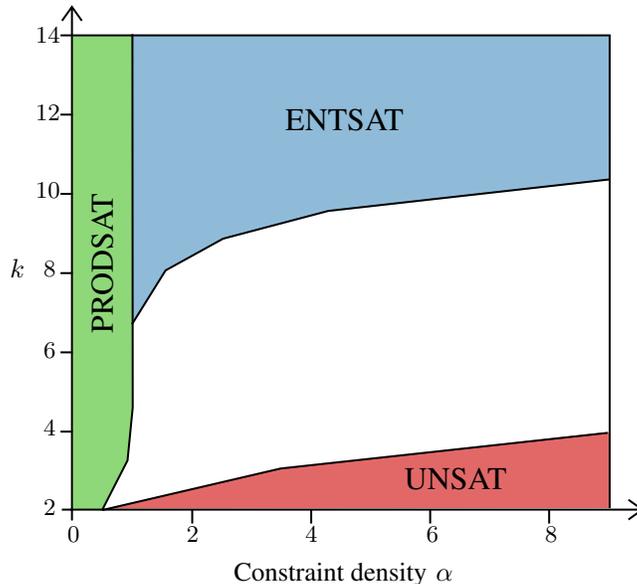


FIGURE 1. Phase diagram of QSAT from [8]. In the red region the problem is UNSAT and it is SAT in the green and blue regions. Furthermore it is PRODSAT in the green region and ENTSAT in the blue region (no satisfying product states but satisfying entangled states). The only line which is known to be exact is the one delimiting the green region: we have $\alpha_{dc}(k = 3) \approx 0.92$ and for $k \geq 4$ this threshold monotonically approaches 1 from below. The two lines delimiting the red and blue regions give respectively upper and lower bounds on the SAT-UNSAT threshold. The proof techniques of these bounds are adapted from corresponding proofs in the classical problem.

The quantum phase diagram is richer than its classical counterpart already at the level of structural phase transitions, and almost nothing is known about algorithmic phase transitions. One may distinguish between PRODSAT, ENTSAT, and UNSAT phases. The UNSAT phase is simply the one where no zero energy eigenstates exist with high probability (w.h.p.). The SAT phase on the contrary is the one where zero energy eigenstates exist w.h.p. It can be decomposed into a PRODSAT phase for which there exist zero energy eigenstates which are fully factorized into a tensor product of single qubit states, and an ENTSAT phase where the zero energy eigenstates cannot be fully factorized into single qubit states (of course one could also envision more refined decompositions of the ENTSAT phase corresponding to partial “non-single-qubit” factorizations). It is rigorously known that for $k = 2$ the ENTSAT phase does not exist and that a sharp PRODSAT-UNSAT phase transition takes place with threshold $\alpha_c = 1/2$ [9]. This is really a geometric transition, closely tied to the sudden proliferation of closed loops in the random factor graphs at this critical density. For $k \geq 3$ we only have loose upper and lower bounds for the various thresholds. In particular the existence of an ENTSAT phase is only proven for $k \geq 7$ and what happens for $3 \leq k \leq 6$ is unclear.

In a series of very interesting papers [9], [10], it is shown that PRODSAT states exist if and only if the factor graph has a constraint-covering dimer configuration. A constraint-covering dimer configuration is a set of edges where all constraints are covered and no two edges meet at a constraint or at a variable node (but some variable nodes may not be covered). This is again a purely geometrical characterization of the PRODSAT phase. Using previous work

on random graph theory [11], [12], this characterization allows to identify the maximum constraint density for which PRODSAT states exist w.h.p., as $\alpha_{\text{PRODSAT}} = \alpha_{\text{dc}}(k)$, where $\alpha_{\text{dc}}(k)$ is the threshold corresponding to the existence of dimer coverings (in thermodynamic limit). However there remains the algorithmic question: can one efficiently find PRODSAT states and with what complexity? This question has been partly answered by looking at a purely geometric leaf-removal based algorithm which determines PRODSAT states in linear time $O(N)$ for $\alpha < \alpha_{\text{lr}}(k) < \alpha_{\text{dc}}(k)$. It has remained unanswered for $\alpha_{\text{lr}}(k) < \alpha < \alpha_{\text{dc}}(k)$.

In this paper we concentrate on the PRODSAT phase and make the above picture fully rigorous.

Theorem 1.1 (Main Theorem). *Take a factor graph from the ensemble $G_{N,M}^k$. Given this graph take a set of M projectors $\{\Pi_m, m = (m_1, \dots, m_k)\}$ uniformly at random. This defines a random formula or equivalently a Hamiltonian H_F in (3). Let \mathbb{P}_F be the probability with respect to this ensemble of random Hamiltonians. We have*

$$\lim_{N,M \rightarrow +\infty} \mathbb{P}_F(\exists |\Psi\rangle = |\varphi_1\rangle \otimes \dots \otimes |\varphi_N\rangle \in (\mathbb{C}^2)^{\otimes N} : H_F|\Psi\rangle = 0) = \begin{cases} 1, & \alpha < \alpha_{\text{dc}}(k) \\ 0, & \alpha > \alpha_{\text{dc}}(k) \end{cases}$$

where the limit is such that $M/N \rightarrow \alpha$ fixed.

Remark 1.2. *It is known [11], [12] that the dimer covering threshold of the factor graph ensemble satisfies $\lim_{N,M \rightarrow +\infty} \mathbb{P}_{G_{N,M}^k}(\exists \text{ a dimer covering}) = 1$ for $\alpha < \alpha_{\text{dc}}(k)$ and 0 for $\alpha > \alpha_{\text{dc}}(k)$.*

The proof draws on ideas already present in [9]. In section 2 we first reformulate the problem on the 2-core of the factor graph and explain the main strategy of proof. For the direct statement we combine two arguments: one is purely analytical (section 3) based on the implicit function theorem for functions of multiple complex variables, and the second is purely algebraic (section 4) based on Buchbergers's algorithm for solving algebraic polynomial equations. In the process we remark that the Buchberger algorithm can be used to provide w.h.p PRODSAT solutions for $\alpha_{\text{lr}}(k) < \alpha < \alpha_{\text{dc}}(k)$. As we will see randomness plays an important role in the analysis. The generic complexity of the algorithm is doubly exponential and it is an open problem to determine what is the real algorithmic complexity of finding these solutions in this range of densities. The proof of the converse statement for $\alpha > \alpha_{\text{dc}}(k)$ is presented in section 5.

The nature of entanglement in this problem (beyond its mere existence for $k \geq 7$) is a largely open question. We make a few numerical observations for formulas with $N = M$ finite and such that dimer coverings exist. These formulas have a finite number of PRODSAT states w.h.p., however our observations suggest that for a fraction of the formulas these states do not span the whole kernel space of the Hamiltonian. Therefore there exists a subspace of the kernel space which only contains entangled states. These observations are presented in section 6.

For the remaining of this paper we say that an event \mathcal{E} happens w.h.p. if $\lim_{N,M \rightarrow +\infty} \mathbb{P}(\mathcal{E}) = 1$ where the limit is such that $M/N \rightarrow \alpha$ and \mathbb{P} is with respect to an ensemble that depends on the context. For example in Theorem 1.1 the ensemble corresponds to the random Hamiltonians (random factor graphs and projectors) whereas in remark 1.2 it simply corresponds to random factor graphs.

2. STRATEGY OF ANALYSIS AND MAIN RESULTS

We proceed with a two-stage process. First, given a graph from $G_{N,M}^k$ we simplify the constraint satisfaction problem to a problem where the numbers of constraints (or projectors) and qubits are equal. Second, this residual constraint satisfaction problem is reformulated as the study of solutions of a set of polynomial equations in complex variables.

2.1. First step: $\alpha < \alpha_{\text{lr}}(k)$. This step is accomplished using the leaf removal process, a Markov process in the space of factor graphs. Given an initial graph $G \in G_{N,M}^k$ one iteratively removes degree-one variable nodes together with its attached constraint node, until the residual graph has minimal variable-node degree at least two (the process then stops). This residual graph is called the core (equivalently, 2-core or hypercore). The theoretical analysis of this Markov process is well known and reviewed in Appendix A. Lemma A.1 defines a threshold $\alpha_{\text{lr}}(k)$ such that for $\alpha < \alpha_{\text{lr}}(k)$ the core is empty w.h.p. and for $\alpha > \alpha_{\text{lr}}(k)$ the core is not empty w.h.p.. Let us denote by G' the residual graph and assume it contains $M' \leq M$ constraint nodes and $N' \leq N$ variable nodes.

Given any *product* state $|\Psi'\rangle \in (\mathbb{C}^2)^{\otimes N'}$ for the qubits of the core (assuming it is non-empty), we construct a state

$$(4) \quad |\Psi\rangle = |\Psi'\rangle \otimes \prod_{i \in G \setminus G'} |\varphi_i\rangle$$

where $|\varphi_i\rangle$ are single qubit states iteratively constructed thanks to Bravyi's transfer matrix by reversing the leaf removal steps. If m is a previously deleted constraint node along with the deleted variable node m_i where $(m_1, \dots, m_{i-1}, m_{i+1}, \dots, m_k)$ are the remaining neighboring variable nodes, the transfer matrix is a linear map $T : (\mathbb{C}^2)^{\otimes k-1} \rightarrow \mathbb{C}^2$ such that

$$(5) \quad |\chi_{m_i}\rangle := T(|\chi_{m_1}\rangle \otimes \dots \otimes |\chi_{m_{i-1}}\rangle \otimes |\chi_{m_{i+1}}\rangle \otimes \dots \otimes |\chi_{m_k}\rangle)$$

and the constraint (2) is satisfied,

$$(6) \quad \langle \Phi^m | \chi_{m_1}\rangle \otimes \dots \otimes |\chi_{m_k}\rangle = 0$$

for any single qubit states $|\chi_{m_1}\rangle, \dots, |\chi_{m_{i-1}}\rangle, |\chi_{m_{i+1}}\rangle, \dots, |\chi_{m_k}\rangle$. That such a linear map T exists and can be constructed explicitly is reviewed in Appendix A. Because we take a product state for $|\Psi'\rangle$ we can apply this transfer matrix to the qubits involved in the last constraint removed and get the qubit state of the last variable node removed. We can then iterate this process following the reversed steps of leaf removal until all qubits are assigned a state thereby obtaining $|\Psi\rangle$.

When the core G' is empty, this construction yields a PRODSAT zero energy state simply by starting with an arbitrary tensor product of the $k-1$ qubits connected to the last removed leaf with its attached constraint. Thus we have the intermediate result

Lemma 2.1. *For $\alpha < \alpha_{\text{lr}}(k)$ there exist PRODSAT zero energy states w.h.p.. Furthermore their construction has time-complexity bounded by $O(N)$.*

2.2. Second step: $\alpha_{\text{lr}}(k) < \alpha < \alpha_{\text{dc}}(k)$. From now on we assume the core G' is non-empty. In order to prove Theorem 1.1 we must solve a constraint satisfaction problem on this residual graph. More precisely we must show that there exists $|\Psi'\rangle$ of *product* form such that for all $m \in G'$

$$(7) \quad |\Phi^m\rangle \langle \Phi^m| \otimes I_{N'-k} |\Psi'\rangle = 0$$

Let us relabel the variable and constraint nodes of G' as $\{1, \dots, N'\} = [N']$ and $\{1, \dots, M'\} = [M']$. Without loss of generality we can use the parametrizations

$$(8) \quad |\Psi'\rangle = \frac{|0\rangle + z_1|1\rangle}{1 + |z_1|^2} \otimes \dots \otimes \frac{|0\rangle + z_{N'}|1\rangle}{1 + |z_{N'}|^2}$$

and ³

$$(9) \quad |\Phi^m\rangle = \sum_{(i_1, \dots, i_k) \in \{0,1\}^k} \phi_{i_1 \dots i_k}^m |i_1\rangle_{m_1} \otimes \dots \otimes |i_k\rangle_{m_k}$$

where $z_1, \dots, z_{N'}$ and $\phi_{i_1 \dots i_k}^m$ are complex numbers and $\sum_{(i_1, \dots, i_k) \in \{0,1\}^k} |\phi_{i_1 \dots i_k}^m|^2 = 1$. It is not difficult to see that the constraints (7) then become

$$(10) \quad \sum_{(i_1, \dots, i_k) \in \{0,1\}^k} \phi_{i_1 \dots i_k}^m z_{m_1}^{i_1} \dots z_{m_k}^{i_k} = 0, \quad m \in [M']$$

Note that the normalization constraint for the coefficients $\phi_{i_1 \dots i_k}^m$ can be dropped as we can always multiply each equation by an arbitrary real positive number. Hence the problem is reduced to solving a set of polynomial equations in the ring $\mathbb{C}[z_1, \dots, z_{N'}]$.

An important idea introduced in [10] is that the existence of solutions for the system (10) is controlled by the presence of constraint-covering dimer configurations of the factor graph (as defined in the introduction).

Proposition 2.2. *Let G' have a constraint-covering dimer configuration. Then the system (10) has a solution for almost all choices of complex coefficients $\{\phi_{i_1 \dots i_k}^m, m \in [M'], (i_1 \dots i_k) \in \{0, 1\}^k\}$.*

In [10] the starting point is the observation that Proposition 2.2 is easy to check if the projector $|\Phi^m\rangle\langle\Phi^m| = |\varphi^{m_1}\rangle\langle\varphi^{m_1}| \otimes \dots \otimes |\varphi^{m_k}\rangle\langle\varphi^{m_k}|$ has product form, i.e., $\phi_{i_1 \dots i_k}^m = \varphi_{i_1}^{m_1} \dots \varphi_{i_k}^{m_k}$ because it suffices then to solve $\varphi_0^{m_*} + z_{m_*} \varphi_1^{m_*} = 0$, where m_* is the unique variable node belonging to the dimer that covers clause m . This equation has a solution for almost all $\varphi_0^{m_*}, \varphi_1^{m_*} \in \mathbb{C}$. Then this result is extended to non-product states through a perturbative argument combined with abstract algebraic geometry theorems. The proof is non-constructive.

Here we will proceed differently with a constructive proof. We first note that when $\{\phi_{0 \dots 0}^m = 0, m \in [M']\}$ the polynomial equations have the trivial solution $z_1 = \dots = z_{N'} = 0$. In section 3 we show through complex analysis arguments that one can construct (with probability one) a unique solution for $\{\phi_{0 \dots 0}^m \neq 0, m \in [M']\} \in U$ a small enough open neighborhood (depending on the instance) of the origin in $\mathbb{C}^{M'}$. It is also possible to give an explicit series expansion formula for the solution. Then, in section 4 we extend the existence of this solution (again with probability one) through an analysis of Buchberger's algorithm for solving polynomial equations in the ring $\mathbb{C}[z_1, \dots, z_{N'}]$.

In section 5 we show a converse statement.

Proposition 2.3. *Let G' have no clause-covering dimer configuration. Then for almost all choices of complex coefficients $\{\phi_{i_1 \dots i_k}^m, m \in [M'], (i_1 \dots i_k) \in \{0, 1\}^k\}$ the system of equations (10) has no solution.*

Putting together Propositions 2.2 and 2.3 we obtain Theorem 1.1.

³Here $|i_1\rangle_{m_1} = |0\rangle_{m_1}, |1\rangle_{m_1}$ are the computational basis states of the space $\mathbb{C}_{m_1}^2$ of qubit m_1 , and similarly for the other qubits m_2, \dots, m_k involved in m .

Proof of Theorem 1.1. Take an instance of a factor graph $G \in G_{N,M}^k$. Note that the instance has a dimer covering if and only if the residual core G' also has a dimer covering. This is in fact so at each intermediate step of leaf removal, because constraint nodes that are not removed remain degree k , and thus their dimer is left untouched. As a result for $\alpha < \alpha_{\text{dc}}$ w.h.p. the core G' has a dimer covering, and Proposition 2.2 implies that w.h.p. there exist tensor product states \mathfrak{s} with zero energy. Conversely for $\alpha > \alpha_{\text{dc}}$ the graph G' w.h.p. has no dimer covering and Proposition 2.3 implies that w.h.p. there are no such product states of zero energy. \square

3. ANALYTICAL PERTURBATIVE ARGUMENT

In this section we prove that when all the constant terms $\{\phi_{0,\dots,0}^m, m \in [M']\}$ are small enough, the system of equations (10) (restricted to the core G') has a solution. Recall that when $\phi_{0,\dots,0}^m = 0$ for all $m \in [M']$ this is obvious as $\mathbf{z} = (z_1, \dots, z_{N'}) = (0, \dots, 0)$ is a trivial solution. We will show that this trivial solution can be extended to a solution when not all $\{\phi_{0,\dots,0}^m, m \in [M']\}$ are 0 but small enough:

Proposition 3.1. *Suppose that G' admits a clause-covering dimer configuration. Then there exists $\epsilon > 0$ such that if $|\phi_{0,\dots,0}^m| < \epsilon$ for all $m \in [M']$ then there exists a solution $(z_1^*, \dots, z_{N'}^*)$ to the system (10) and a corresponding tensor product state $|\Psi'\rangle$ satisfying (7). Note that ϵ depends on G' and $\{\phi_{i_1 \dots i_k}^m, m \in [M'], (i_1, \dots, i_k) \in \{0, 1\}^k \setminus (0, \dots, 0)\}$.*

Before proving Proposition 3.1 we make a *reduction* of the system (10) to a *square* system. For $\mathbf{z} = (z_1, \dots, z_{N'}) \in \mathbb{C}^{N'}$ and $m \in [M']$ let

$$(11) \quad f^m(\mathbf{z}) = \sum_{\substack{i_1 \dots i_k \in \{0,1\}^k \\ \phi_{0,\dots,0}^m = 0}} \phi_{i_1 \dots i_k}^m z_{m_1}^{i_1} \dots z_{m_k}^{i_k},$$

These functions are simply the polynomials in (10) without the constant terms and $(0, \dots, 0)$ is a common zero. The existence of the dimer covering (assumed in Proposition 3.1) guarantees that there is one variable node m_\star in the neighborhood of constraint m that matches it (i.e. (m_\star, m) is a dimer). We *reduce* the system (11) of M' equations and N' variables to a *square* system of M' equations and M' variables by assigning $z = 0$ for all the variable nodes that are not in the dimer covering. The polynomials of the reduced system will be denoted as f_{sqr}^m . Note that f_{sqr}^m might contain fewer than k variables for some m (this happens when m has neighboring variables not covered by the chosen dimer covering). The following relabelling of complex variables turns out to be useful: given the labelling $\{1, \dots, M'\}$ of constraint nodes we relabel the complex variables associated to nodes in the dimer covering by $\{z_{1_\star}, \dots, z_{M'_\star}\}$. Finally we define the multivariate complex map

$$(12) \quad F_{\text{sqr}} : \mathbb{C}^{M'} \rightarrow \mathbb{C}^{M'}$$

$$(13) \quad \mathbf{z} = (z_{1_\star}, \dots, z_{M'_\star}) \mapsto (f_{\text{sqr}}^1(\mathbf{z}), \dots, f_{\text{sqr}}^{M'}(\mathbf{z}))$$

where $\mathbf{z} = (z_{1_\star}, \dots, z_{M'_\star})$ is the set composed only of variables in the dimer covering. The Jacobian matrix of F_{sqr} is the $M' \times M'$ matrix

$$(14) \quad J_{F_{\text{sqr}}} := \left(\frac{\partial f_{\text{sqr}}^m}{\partial z_{j_\star}} \right)_{m \in [M'], j \in [M']}.$$

A crucial remark is the following: with probability one $\phi_{i_1 \dots i_k}^m \neq 0$, thus for any $m \in [M']$ there are monomials in f_{sqr}^m containing the variable z_{m_\star} ; in particular it is guaranteed that a linear monomial of the form $\phi_{0 \dots 010 \dots 0}^m z_{m_\star}$ is always present. Thus, with probability one, all the diagonal elements of the Jacobian $\frac{\partial f_{\text{sqr}}^m}{\partial z_{m_\star}}$ are polynomials with a constant term $\phi_{0 \dots 010 \dots 0}^m$ and in particular,

$$(15) \quad \left. \frac{\partial f_{\text{sqr}}^m}{\partial z_{m_\star}} \right|_{\mathbf{z}=\mathbf{0}} = \phi_{0 \dots 010 \dots 0}^m \neq 0.$$

Lemma 3.2. *G' has a dimer covering if and only if $J_{F_{\text{sqr}}}(\mathbf{0})$ is a full rank matrix with $\det J_{F_{\text{sqr}}}(\mathbf{0}) \neq 0$.*

Proof. Consider the Jacobian matrix at $\mathbf{z} = \mathbf{0}$. A row $m \in [M']$ always contains *at most* k non-vanishing elements among $\phi_{10 \dots 0}^m, \phi_{010 \dots 0}^m, \dots, \phi_{0 \dots 01}^m$ corresponding to the neighboring nodes belonging to some dimer covering.

We first prove the direct statement. Suppose G' has a dimer covering. Then as remarked above, with probability one each row contains *at least* one non-vanishing element and especially one on the main diagonal. We run Gaussian elimination on $J_{F_{\text{sqr}}}(\mathbf{0})$ with the elements on the main diagonal as the pivot to obtain the matrix in row echelon form. At each step of the algorithm we linearly combine rows, and the new terms we get on the diagonal can only be polynomial functions of the ϕ 's. Since these polynomials are holomorphic multivariate functions (of the ϕ 's) they have zero locus of measure zero [13]. Eventually we get an upper triangular matrix with non-zero terms on the main diagonal. This proves that $J_{F_{\text{sqr}}}(\mathbf{0})$ is full row-rank, and since it is a square matrix $\det J_{F_{\text{sqr}}}(\mathbf{0}) \neq 0$.

For the converse statement we must show that if $J_{F_{\text{sqr}}}(\mathbf{0})$ is full row-rank, then we can associate to each row $m \in [M']$ a column $j_m \in [M']$ such that the matrix element (m, j_m) is non-zero and $j_m \neq j_{m'}$ for $m \neq m'$. The injective mapping $m \mapsto j_m$ provides the dimer covering. Assume that no such injective mapping exists. Then, as we go down the rows, at a certain point, say for the \bar{m} -th row, we cannot come up with $j_{\bar{m}}$ such that $j_{\bar{m}} \neq j_m$ for all $m < \bar{m}$. This means all non-zero elements of the \bar{m} -th row belong among the previously chosen columns $\{j_m | m < \bar{m}\}$. Therefore the \bar{m} -th row is a vector in the span of the previous rows $\{m < \bar{m}\}$. This contradicts the full row-rank assumption. \square

In the rest of this section we use results from multivariate complex analysis reviewed in Appendix B.

Lemma 3.3. *Suppose $\det J_{F_{\text{sqr}}}(\mathbf{0}) \neq 0$. Then there exist $\epsilon > 0$ such that $\mathbf{0}$ is the only zero of the map F_{sqr} in the open ball $B(\mathbf{0}, \epsilon)$. In other words $\mathbf{0}$ is an isolated zero of the map F_{sqr} .*

Proof. By construction $F_{\text{sqr}}(\mathbf{0}) = \mathbf{0}$. Since each polynomial f_{sqr}^m is an holomorphic multivariate function we can use Theorem B.6 to directly deduce the existence of $B(\mathbf{0}, \epsilon) \subset \mathbb{C}^{M'}$ such that F_{sqr} is biholomorphic in $B(\mathbf{0}, \epsilon)$. In particular $F_{\text{sqr}}|_{B(\mathbf{0}, \epsilon)}$ is a bijection from $B(\mathbf{0}, \epsilon)$ to $F_{\text{sqr}}(B(\mathbf{0}, \epsilon))$ so $\mathbf{0}$ is the only solution of $F_{\text{sqr}}(\mathbf{z}) = \mathbf{0}$ in $B(\mathbf{0}, \epsilon)$. This means $\mathbf{0}$ is an isolated zero. \square

We now turn to the proof of the main result of this section:

Proof of Proposition 3.1. Since the graph has a dimer covering, Lemma 3.2 implies that $\det J_{F_{\text{sqr}}}(\mathbf{0})$ is nonzero. So by Lemma 3.3, $\mathbf{0}$ is an isolated zero in the open ball $B(\mathbf{0}, \epsilon)$.

Choose $0 < \epsilon' < \epsilon$ so that $\mathbf{0}$ is the only zero of F_{sqr} in the closure of $B(\mathbf{0}, \epsilon')$. Proposition B.3 then states that we can find $\varphi > 0$ small enough such that the system of equations

$$(16) \quad f_{\text{sqr}}^m(\mathbf{z}) + \phi_{0\dots 0}^m = 0, \quad m \in [M']$$

has simple zeros for almost all values of the constant terms in the set $\{|\phi_{0\dots 0}^m| < \varphi, m \in [M']\}$. Moreover because $\mathbf{0}$ itself is a simple zero of the F_{sqr} (i.e., it is isolated and $\det J_{F_{\text{sqr}}}(\mathbf{0}) \neq 0$) we deduce from Proposition B.5 that the solution of equations (16) is unique for small enough constant terms. This implies the existence of a solution $(z_1^*, \dots, z_{N'}^*)$ for the full system (10) for small enough constant terms. The solution we have constructed here consists of $z_j^* = 0$ if j does not belong to the dimer covering (recall the reduction step above) and z_j^* the unique solution of (16) if j belongs to the dimer covering. We note that while this solution for (16) is unique (for small enough constant terms) it is not unique for (10). Indeed we could have done a similar construction by setting the z_j variables of nodes j not in the dimer covering to non-zero values. \square

4. ALGEBRAIC NON-PERTURBATIVE ARGUMENT

In the previous section, we proved Lemma 3.1 stating that if there exists a dimer covering of the interaction graph, then instances with small constant terms have a PRODSAT solution w.h.p.. To extend this result to all possible instances, we will use Buchberger algorithm and Gröbner basis. These are powerful tools to solve systems of complex multivariate polynomial equations and hence also give a method to directly find the zeros of the system (10) of constraints. For the description of Buchberger algorithm, we refer to Appendix C and [14].

Definition 4.1. *A polynomial is called generic if it is a polynomial of the form Eq. 10 such that the coefficients of each monomial are taken uniformly at random on the unit sphere $(\mathbb{C}^2)^{\otimes k}$.*

In Appendix C, we review the following corollary of Hilbert's Nullstellensatz.

Corollary 4.2. *A set of polynomials in an algebraically closed field has no common zeros if and only if the reduced Gröbner basis is $\{1\}$.*

Proposition 4.3. *Let $\mathcal{F} = (f_1, f_2, \dots, f_m)$ be a set of generic polynomial equations in $K[X_0, \dots, X_n]$ that has a common solution, then for any given $a \in K$, $\mathcal{F}_a := (f_1 + a, f_2, \dots, f_m)$ also have a common solution with probability 1 with respect to the distribution of the constituent coefficients of \mathcal{F} .*

Proof. If $\mathcal{F} = (f_1, \dots, f_m)$ have a common zero then by Corollary 4.2, there exists a Gröbner basis not reduced to 1 for \mathcal{F} . We want to show that $\mathcal{F}_a = (f_1 + a, f_2, \dots, f_m)$ will also have a Gröbner basis that is not reduced to 1, and therefore admitting a common solution. This will be achieved by convincing ourselves that the Buchberger's algorithm applied on \mathcal{F} and \mathcal{F}_a produce the same steps in the sense that the monomials involved in each step for \mathcal{F} and for \mathcal{F}_a are identical with probability 1. Let us analyze each step of Buchberger's algorithm.

Computation of the S-polynomial. When we compute $S_{i,j}$ in \mathcal{F} and $S_{i,j}^a$ in \mathcal{F}_a (see step 6 in Algorithm 2), the two lists of monomials in $S_{i,j}$ and $S_{i,j}^a$ will be the same with probability 1. Indeed, it could happen that the coefficients of the monomials in $S_{i,j}^a$ vanish but this puts algebraic constraint on the constituent coefficients of \mathcal{F} . With respect to the distribution of the constituent coefficients, the constraint is satisfied with probability 0.

Reduction through multivariate division algorithm 2. We should also check that in the steps 6 to 8 of Algorithm 2, the monomials produced starting from \mathcal{F} and those produced starting from \mathcal{F}_a will be the same with probability 1. The only way that at some steps the monomials differ is that the coefficients of the monomials produced by \mathcal{F}_a vanish. As previously, this puts an algebraic constraint on the constituent coefficients of \mathcal{F} that would be satisfied with vanishing probability.

We note that the number of steps in the algorithm is finite and therefore the number of algebraic constraints that stem from the application of Buchberger's algorithm to \mathcal{F}_a is finite. Thus such algebraic constraints make up a set of measure 0 of coefficients for \mathcal{F} . \square

To illustrate this proof, we detail the steps of Buchberger's algorithm using an example of 2-QSAT on 3 variables.

$$\begin{aligned} f_1 &= a_0 + a_1 z_1 + a_2 z_2 + a_{12} z_1 z_2 \\ f_2 &= b_0 + b_2 z_2 + b_3 z_3 + b_{23} z_2 z_3 \\ f_3 &= c_0 + c_1 z_1 + b_3 z_3 + c_{13} z_1 z_3. \end{aligned}$$

Example 4.4 (Computation of the S -polynomial).

$$\begin{aligned} LCM(f_1, f_2) &= a_{12} b_{23} z_1 z_2 z_3 \\ S(f_1, f_2) &= b_{23} z_3 \cdot f_1 - a_{12} z_1 \cdot f_2 \\ &= -a_{12} b_0 z_1 + a_0 b_{23} z_3 - a_{12} b_2 z_1 z_2 + (a_1 b_{23} - a_{12} b_3) z_1 z_3 + a_2 b_{23} z_2 z_3 \\ LCM(f_1 + a, f_2) &= a_{12} b_{23} z_1 z_2 z_3 \\ S(f_1 + a, f_2) &= b_{23} z_3 \cdot (f_1 + a) - a_{12} z_1 \cdot f_2 \\ &= -a_{12} b_0 z_1 + (a_0 + a) b_{23} z_3 - a_{12} b_2 z_1 z_2 + (a_1 b_{23} - a_{12} b_3) z_1 z_3 + a_2 b_{23} z_2 z_3 \end{aligned}$$

In this example, we must avoid the event $a + a_0 = 0$ which would delete the monomial z_3 . This event has probability 0.

Example 4.5 (Reduction through multivariate division algorithm). $S(f_1, f_2)$ (resp. $S(f_1 + a, f_2)$) is successively reducible by f_2, f_3 and f_1 (resp. f_2, f_3 and $f_1 + a$). Set $A = (a_1 b_{23} - a_{12} b_3)/c_{13}$. We have

$$\begin{aligned} p_1 &= S(f_1, f_2) - a_2 \cdot f_2 \\ &= -a_2 b_0 - a_{12} b_0 z_1 - a_2 b_2 z_2 + (a_0 b_{23} - a_2 b_3) z_3 - a_{12} b_2 z_1 z_2 + (a_1 b_{23} - a_{12} b_3) z_1 z_3 \\ p_2 &= S(f_1, f_2) - a_2 \cdot f_2 - \frac{a_1 b_{23} - a_{12} b_3}{c_{13}} \cdot f_3 = S(f_1, f_2) - a_2 \cdot f_2 - A \cdot f_3 \\ &= -(a_2 b_0 + c_0 A) - (a_{12} b_0 + c_1 A) z_1 - a_2 b_2 z_2 + (a_0 b_{23} - a_2 b_3 - b_3 A) z_3 - a_{12} b_2 z_1 z_2 \\ p_3 &= S(f_1, f_2) - a_2 \cdot f_2 - A \cdot f_3 + b_2 \cdot f_1 \\ &= (a_0 b_2 - a_2 b_0 - c_0 A) + (a_1 b_2 - a_{12} b_0 - c_1 A) z_1 + (a_0 b_{23} - a_2 b_3 - b_3 A) z_3 \end{aligned}$$

$$\begin{aligned}
p_1^\alpha &= S(f_1 + a, f_2) - a_2 \cdot f_2 \\
&= -a_2 b_0 - a_{12} b_0 z_1 - a_2 b_2 z_2 + ((a_0 + a) b_{23} - a_2 b_3) z_3 - a_{12} b_2 z_1 z_2 + (a_1 b_{23} - a_{12} b_3) z_1 z_3 \\
p_2^\alpha &= S(f_1 + a, f_2) - a_2 \cdot f_2 - A \cdot f_3 \\
&= -(a_2 b_0 + c_0 A) - (a_{12} b_0 + c_1 A) z_1 - a_2 b_2 z_2 + ((a_0 + a) b_{23} - a_2 b_3 - b_3 A) z_3 - a_{12} b_2 z_1 z_2 \\
p_3^\alpha &= S(f_1 + a, f_2) - a_2 \cdot f_2 - A \cdot f_3 + b_2 \cdot (f_1 + a) \\
&= ((a_0 + a) b_2 - a_2 b_0 - c_0 A) + (a_1 b_2 - a_{12} b_0 - c_1 A) z_1 + ((a_0 + a) b_{23} - a_2 b_3 - b_3 A) z_3
\end{aligned}$$

The algebraic constraints in this example are $a_2 b_3 - (a_0 + a) b_{23} = 0$ at step 1, $(a_2 b_3 - b_3 A) - (a_0 + a) b_{23} = 0$ at step 2 and 3 and $(a_2 b_0 + c_0 A) - (a_0 + a) b_2 = 0$ at step 3. These occur with probability 0.

The proof of Theorem 1.1 follows then directly from Propositions 3.1, 4.3. Thus we have proved that if there exists a dimer covering then there exists a PRODSAT solution w.h.p..

5. CONVERSE STATEMENT

We prove Proposition 2.3. The proof relies on Hall's marriage theorem [15] stated below and on the Macaulay resultant of a system of polynomials [16]. For a system of homogeneous polynomial equations of the same number of equations and variables with coefficients in an algebraically closed field (here \mathbb{C}), the Macaulay resultant is a polynomial of the coefficients which vanishes if and only if the system of equations has a common non-zero solution. For more details on the resultant and its property we refer to [17, Chap 3. §2].

Theorem 5.1 (Hall's marriage theorem). *For a bipartite graph $(V, E) = (A \cup B, E)$, the following conditions are equivalent.*

- *There is a perfect matching of A into B .*
- *For each $S \subseteq A$, the inequality $|S| \leq |N(S)|$ holds where $N(S)$ denotes the neighboring nodes of S in B .*

Remark 5.2. *A perfect matching 'of A into B ' is a dimer configuration which covers all nodes in A (but not necessarily all nodes of B) such that no two edges have common nodes.*

Proof of Proposition 2.3. We apply Theorem 5.1 to the factor graph G' with $A = [M']$ the set of constraint nodes and $B = [N']$ the set of variable nodes. Thus there exists a constraint-covering dimer configuration if and only if for any subset $S \subseteq [M']$ the number of variables appearing in those constraints satisfies $|S| \leq |N(S)|$. Taking the contrapositive, if G' has no constraint-covering dimer configuration, there must exist a subset $S \subset [M']$ with $|N(S)| < |S|$. One can find a subset $S' \subseteq S$ with $|S'| = |N(S)| + 1$ constraints which contains all the variables of $N(S)$. This set S' corresponds to a system of $|N(S)| + 1$ polynomial equations of the form 10 with $|N(S)|$ variables. Now we show that this overdetermined system of equations does not admit a solution which implies that the full system cannot admit a solution.

Take the polynomials corresponding to S' , with variables relabeled as $z_1, \dots, z_{|N(S)|}$, and make them homogeneous by introducing an additional variable z_0 , as follows

$$(17) \quad z_0^k \sum_{(i_1, \dots, i_k) \in \{0,1\}^k} \phi_{i_1 \dots i_k}^m \left(\frac{z_{m_1}}{z_0} \right)^{i_1} \dots \left(\frac{z_{m_k}}{z_0} \right)^{i_k}, \quad m \in S'$$

Suppose now that the system of original equations has a common solution $(z_1^*, \dots, z_{|N(S)|}^*)$. Then the system of homogeneous equations also has a common solution $(z_0 = 1, z_1^*, \dots, z_{|N(S)|}^*)$

and this solution is not the zero solution (since $z_0 = 1$). Therefore the Macaulay resultant of the homogeneous system must vanish. However this resultant itself is a polynomial in the variables $\{\phi_{m_1, \dots, m_k}^m, m \in S'\}$ and is an holomorphic function. The zero locus of an holomorphic function has measure 0 [13] and therefore the Macaulay resultant does not vanish with probability 1. Hence with probability 1 the system of equations cannot have a common solution. \square

6. SIMULATIONS

In this section, we investigate two issues in order to better understand the nature of the PRODSAT phase and its possible transition towards the ENTSAT phase. For $k \geq 8$ it is established that the ENTSAT phase exists but this is still open for lower k . The discussion in this section applies to any k but we run simulations for $k = 3$, as they are too costly in practice for higher values.

6.1. Dimer coverings and dimension of solution space. Theorem 1.1 establishes a precise connection between the presence of a dimer covering and the existence of a PRODSAT solution. It is therefore of interest to further investigate if the *structure* of the interaction graph can provide insights about the dimension of the solution space of H_F .

We first gather a few observations about $\dim \ker H_F$. There are two sources of randomness in k-QSAT: the interaction graph and the choice of the projectors. For a fixed interaction graph, let us consider the corresponding Hamiltonian where the coefficients of the projectors are to be thought as *indeterminates*. The $2^N \times 2^N$ Hamiltonian matrix H_F (3) is sparse when it is represented in the computational basis since the projectors are k -local. Then the determinants of the $s \times s$ submatrices are polynomials in the indeterminates. Two situations may arise. These polynomials may be equal to a trivially ‘null polynomial’ or to a bona fide non-trivial polynomial. Let S be the largest s such that there exists an $s \times s$ submatrix M_{H_F} whose determinant is a non-trivial polynomial. Over the choices of the random projectors, the determinant of M_{H_F} will vanish on a set of measure 0. Thus the rank of H_F will take the value S with probability 1. Therefore, for any fixed interaction graph, we have $\dim \ker H_F = 2^N - S$ with probability 1 over the choices of the random projectors.⁴ Note also that by the definition of S , for any given instance of the random projectors, $\text{rank} H_F \leq S$ necessarily so that $2^N - S \leq \dim \ker H_F$.

We would like to compute the ‘generic’ value of $\dim \ker H_F$ which is $2^N - S$. This is not easy in general. Nevertheless, by the above remarks, it is certainly upper bounded by $\dim \ker H_F$ for *separable projectors* (i.e., $|\Phi^m\rangle$ is a product state). This is interesting because it is easier to compute $\dim \ker H_F$ for separable projectors, at least for a few simple graphs. It is not clear a priori when this upper bound is an equality because separable projectors form a set of measure zero in the space of all projectors, but numerical simulations suggest that this is so for the graphs reviewed. Figure 2 and Table 6.1 show a set of graphs and corresponding recurrence relations for $\dim \ker H_F$ (denoted r_m) and also for the number of dimer coverings (denoted d_m).

These intriguing relations unfortunately do not seem to clearly demonstrate a general link between the number of dimer coverings and the dimension of the null space. Within this limited set of graphs, for a given graph type, we observe either $r_m \geq d_m$ or $r_m \leq d_m$ for all m . We have not found a universal relation between r_m and d_m beyond these inequalities.

⁴This is nothing other than the content of the geometrization theorem [9].

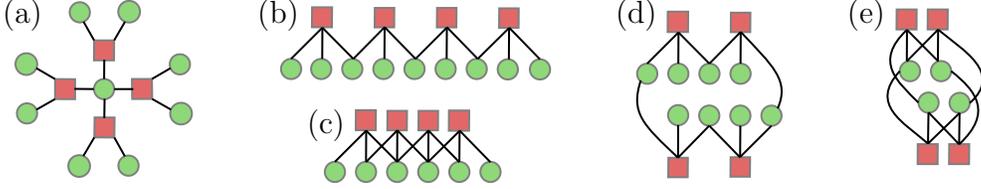


FIGURE 2. Different patterns for $k = 3, m = 4$. (a) Sunflower (b) Loose chain (c) Strong chain (d) Loose cycle (e) Strong cycle. For the strong chain and the strong cycle, each qubit is connected to k clauses, except for the boundary qubits. The sunflower is constructed around one central qubit.

Graph	Dimension of the kernel		Dimer covering	
	Initial values	r_m	Initial values	d_m
Sunflower [†]	$r_1 = 7$	$3r_{m-1} + 3^{m-1}$	$d_1 = 3$	$2^{m-1}(m+2)$
Loose chain*	$r_1 = 7, r_2 = 24$	$4r_{m-1} - 2r_{m-2}$	$d_1 = 3, d_2 = 8$	$3d_{m-1} - d_{m-2}$
Loose cycle*	$r_2 = 12, r_3 = 40$	$4r_{m-1} - 2r_{m-2}$	$d_2 = 3, d_3 = 18$	$3d_{m-1} - d_{m-2}$
Strong chain**	$r_1 = 7, r_2 = 12$	$r_{m-1} + r_{m-2} + 1$	$d_1 = 3, d_2 = 7, d_3 = 14$	$2d_{m-1} - d_{m-3} + 1$
Strong cycle**	$r_4 = 8, r_5 = 12$	$r_{m-1} + r_{m-2} - 1$	$d_4 = 9, d_5 = 13, d_6 = 20$	$2d_{m-1} - d_{m-3}$

TABLE 1. Recurrence relations for patterns in Fig. 2 with $k = 3$ and m the number of clauses. Regarding the dimension of the null space: † the recurrence relation is proved in [18] for all projectors (including non-separable ones); * the two recurrence relations are proved in Appendix D for separable projectors - numerical simulations give the dimension of the null space equal to this upper bound; ** are deduced by numerical simulation.

For example, we have $r_m = d_m - 1$ in the case of the strong cycle and $r_m = d_m + m + 3$ in the case of the strong chain.

For $k = 2$, the only satisfiable graphs are the tree and the cycle (it is easy to check that two intersecting cycles are not satisfiable [9]). In that case, it is known that there is a gap between $\dim \ker H_F = N + 1$ of a tree and $\dim \ker H_F = 2$ of a cycle. However this linear growth of the gap does not seem to persist for $k = 3$. Indeed, in solving the recurrence relations, we observe that the dimension of the null space for all patterns grows exponentially in m (this exponential growth just follows from the order 2 relations).

6.2. PRODSAT basis. We now wish to discuss *how much entanglement is present* in the PRODSAT phase by comparing the dimension of the space generated by the product solutions with that of the full solution space.

A k -QSAT instance is PRODSAT if it is satisfied by a product state. However, this does not imply that all the solutions to the problem are product states. Indeed, the (normalized) sum of two different product states is still a solution to the problem and is likely to be entangled. For a given instance of random k -QSAT, the space generated by all the PRODSAT solutions is referred to as the *PRODSAT space*. A basis of the full solution space, $\ker H_F$, is said to be a *fully PRODSAT basis*, if all the vectors of the basis are product states. Let $\dim \text{PRODSAT}$ denote the dimension of the PRODSAT space.

An interesting question is the following: Is it true that the kernel space admits a fully PRODSAT basis? While we do not directly study an ENTSAT phase in this paper, this question is clearly motivated by the harder issue of how a PRODSAT phase potentially disappears in favor of an ENTSAT phase.

Here we address this question in the following restricted setting of finite sizes with $N = M$ and $k = 3$. Note that although $M/N = 1$, we are dealing here with finite size, so there exist instances with dimer coverings which are therefore PRODSAT. In particular, for $M = N = 5, 6$ it is known that *all* graphs have dimer coverings.

Even in the restricted setting $N = M$ and $k = 3$, it is not easy to compare $\dim \text{PRODSAT}$ and $\dim \ker H_F$, and here this is done only for moderate sizes up to $N = M = 11$. Indeed, to obtain $\dim \ker H_F$ we use exact diagonalization to count the zero eigenvalues of the Hamiltonian which costs roughly $O(2^{3N})$ operations. At the same time, the computation of the PRODSAT solutions can be achieved through Buchberger's algorithm which requires a substantial runtime even for moderate sizes. Instead, we will rely on the BKK theorem (6.2) to obtain only the number of PRODSAT solutions of Eq. 10. Before stating the theorem, we need to recall the following:

Definition 6.1. *The Newton polytope of a polynomial $f = \sum_{\alpha \in \Gamma} c_\alpha x^\alpha$, $\Gamma \subset \mathbb{Z}^n$ is the polytope formed by the convex hull of the set of all $\alpha \in \Gamma$. For polytopes P_1, \dots, P_n , the Mixed Volume $MV_n(P_1, \dots, P_n)$ is the coefficient of the monomial $\lambda_1 \dots \lambda_n$ in the polynomial $f(\lambda_1, \dots, \lambda_n) = \text{Vol}_n(\lambda_1 P_1 + \dots + \lambda_n P_n)$ where the $+$ represents the Minkowski sum. Figure 3 is an example of these definitions. For a k -QSAT instance, we denote by MV the mixed volume of the polytopes associated with the polynomials in Eq. 10.*

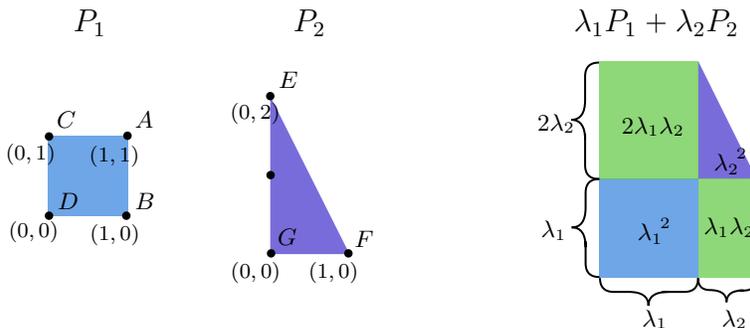


FIGURE 3. Example of Mixed Volume Computation. The Newton polytope P_1 of the polynomial $f_1 = Axy + Bx + Cy + D$ is the square with vertices $\{(1, 1), (1, 0), (0, 1), (0, 0)\}$. For $f_2 = Ey^2 + Fx + G$, it is a triangle with vertices $\{(0, 2), (1, 0), (0, 0)\}$. The decomposition of $\text{Vol}_n(\lambda_1 P_1 + \lambda_2 P_2) = \lambda_1^2 + 3\lambda_1 \lambda_2 + \lambda_2^2$ is represented on the figure. Then the mixed volume is 3.

Theorem 6.2 (BKK theorem [19]). *Let f_1, \dots, f_n be Laurent polynomials over \mathbb{C} ,*

$$(18) \quad f_i = \sum_{\alpha \in \Gamma_i} c_\alpha x^\alpha \quad c_\alpha \in \mathbb{C}, \quad \Gamma_i \subset \mathbb{Z}^k$$

with finitely many common zeroes in $(\mathbb{C}^)^n$. Let P_i be the Newton polytope of f_i . Then the number of common zeroes of the f_i in $(\mathbb{C}^*)^n$ is upper bounded by the mixed volume $MV_n(P_1, \dots, P_n)$. For generic choices of coefficients in f_i 's, the number of common solutions equals $MV_n(P_1, \dots, P_n)$.*

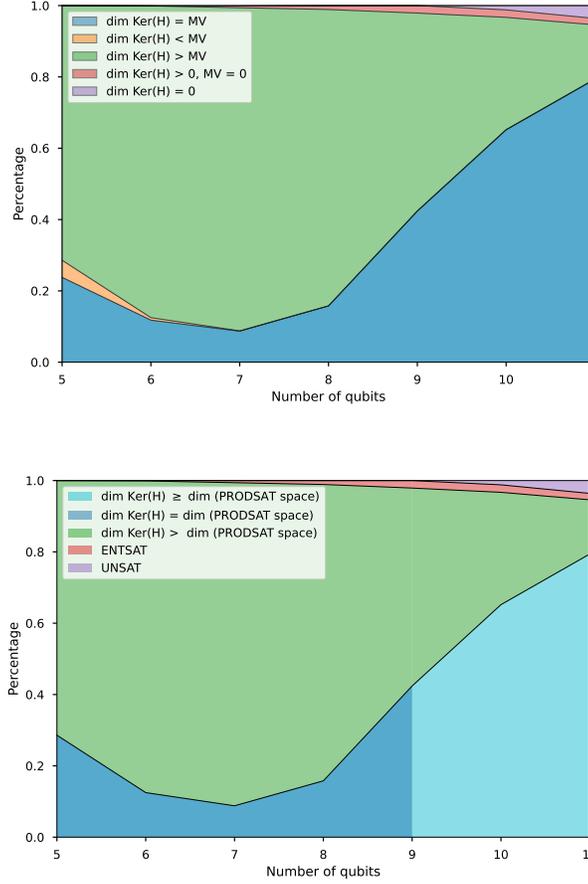


FIGURE 4. Comparison among $\dim \ker H_F$, $\dim \text{PRODSAT}$ and MV . On the left, $\dim \ker H_F$ and MV are compared. On the right, we reinterpret the results in terms of PRODSAT space, UNSAT, and ENTSAT instances. For blue and green, we use $\dim \text{PRODSAT} \leq MV$. Red and purple follow from the definition of the phases. Orange results are joined with blue since we always have $\dim \ker H_F \geq \dim \text{PRODSAT}$. For $N = 5, 6$ all the instances with $M = N$ are computed (resp. 252 and 38500). For $7 \leq N \leq 10$, 5000 instances are sampled uniformly. For $N = 11$, only 500 instances are used.

Remark 6.3. For a given set of equations of the form 10, the corresponding mixed volume does not depend on the coefficients of projectors, but only on the monomials. Hence, for k -QSAT, the mixed volume only depends on the interaction graph. Regarding the complexity of computing the mixed volumes of polytopes, it is at least $\#P$ -hard [20].

Since the product states obtained by substituting the $\{z_i\}$ solutions of Eq. 10 in Eq. 8 could be linearly dependent, the mixed volume only gives an upper bound on $\dim \text{PRODSAT}$,

(19)
$$\dim \text{PRODSAT} \leq MV.$$

When we can compute the $\{z_i\}$ with Buchberger's algorithm, we can check whether this inequality is tight or not by looking for linear dependencies between PRODSAT solutions. Three situations can arise:

- $MV < \dim \ker H_F$. Then $\dim \text{PRODSAT} \leq MV < \dim \ker H_F$ so the basis is not fully PRODSAT.
- $MV = \dim \ker H_F$. Then $\dim \text{PRODSAT} \leq MV = \dim \ker H_F$ so we cannot conclude if the basis is fully PRODSAT or not.
- $MV > \dim \ker H_F$. Then $\dim \text{PRODSAT} \leq \dim \ker H_F < MV$ so there must be linear dependencies among PRODSAT solutions. We cannot conclude if the basis is fully PRODSAT or not.

Figure 4 shows the percentage of instances for which these scenarios occur for $M = N$ between 5 and 11. For increasing N , we observe an increase in the proportion of instances with $MV = \dim \ker H_F$, which is somewhat unexpected (blue region). In particular, up to $N \leq 9$, we can check that $\dim \text{PRODSAT} = \dim \ker H_F$ so the basis is indeed fully PRODSAT. Unfortunately, it is difficult to assess if this is still true for $N = 10, 11$ but the trend in the figure suggests this might be so. This finding may seem rather surprising as one might have expected that the trend of the share of the green region increasing, observed for $N = 5, 6, 7$, would continue with fully PRODSAT basis becoming rarer. We also find that for $N \geq 7$, there appear a small fraction of instances for which $\dim \ker H_F > 0$ and $MV = 0$. This corresponds to the existence of ENTSAT instances. For $N \geq 9$, there also appear a fraction of UNSAT instances. These results may point towards a picture of coexisting fractions of fully PRODSAT and non-fully PRODSAT instances in the large size limit $N, M \rightarrow +\infty, M/N = 1$ for a random ensemble of instances *conditioned* on the existence of dimer coverings.

APPENDIX A. LEAF REMOVAL PROCESS (LR)

In this appendix, we give the proof of Lemma 2.1 for completeness. The proof is algorithmic and based on two ingredients, namely a leaf removal process on factor graphs and Bravyi's transfer matrix.

We begin with a brief review of the leaf removal (LR) process on a factor graph. We first delete isolated (degree-zero) vertices. Next, we choose a unary (degree-one) variable v_1 at random and delete it along with its sole neighbor $a \in \partial v_1$. By doing so, we also remove the other $k - 1$ edges of a connected to $v_2, \dots, v_k \in \partial a$. Such removal possibly makes some or all of v_2, \dots, v_k isolated or unary variables. Delete the isolated variables once again and then start again at an unary variable to delete further on as described before. We iterate this process until we cannot find any more isolated or unary variables. The process concludes with a subgraph where each variable is connected to at least two checks while all the checks are still connected to k variables. This (possibly empty) subgraph is referred to as the 2-core of the hypergraph G (equivalently, core or hypercore).

The study of the 2-core was done in [21] where the existence of a threshold $\alpha_{\text{lr}}(k)$ is proven, below which the 2-core is empty and above which it is not empty w.h.p.. More precisely, let $G_{N,p=\alpha/N^{k-1}}^k$ be the underlying k -uniform hypergraph where each $\binom{n}{k}$ possible edges appear with probability p , we have the following lemma:

Lemma A.1. [21, Theorem 1] *Define*

$$\alpha_{\text{lr}}(k) = \min_{x>0} \frac{(k-1)!x}{(1-e^{-x})^{k-1}}.$$

(1) *For any $\alpha < \alpha_{\text{lr}}$, $G_{N,p=\alpha/N^{k-1}}^k$ has no non-empty 2-core w.h.p.*

(2) For $\alpha > \alpha_{lr}$, $G_{N,p=\alpha/N^{k-1}}^k$ has a 2-core of size $\beta(\alpha)N + o(N)$ w.h.p., with $\beta(\alpha) = 1 - e^{-x} - e^{-x}x$, where x is the greatest solution of

$$\alpha = \frac{(k-1)!x}{(1-e^{-x})^{k-1}}.$$

We note that the construction of $G_{N,p=\alpha/N^{k-1}}^k$ is a bit different from $G_{N,M}^k$ but the two random hypergraph models are mutually contiguous meaning that any events that happen w.h.p. in $G_{N,p=\alpha/N^{k-1}}^k$ also happen w.h.p. in $G_{N,M}^k$ and vice versa.

The second ingredient needed is Bravyi's transfer matrix [1]. As described above, we remove certain vertices and edges according to LR. An inherent reason for removing them is that the removed constraints should be easily satisfied by the removed variables. We show how to implement this idea here (see [1] for $k = 2$).

Lemma A.2. [1, for $k = 2$] For all projectors $|\Phi^m\rangle\langle\Phi^m|$ and any selected variable node m_i involved in constraint $m \equiv \{m_1, \dots, m_k\}$, we can construct a transfer matrix T of size $2 \times 2^{k-1}$ such for that given any product state $|\chi_{m_1}\rangle \otimes \dots \otimes |\chi_{m_{i-1}}\rangle \otimes |\chi_{m_{i+1}}\rangle \dots \otimes |\chi_{m_k}\rangle$, the constraint is satisfied

$$\langle\Phi^m|\chi_{m_1}\rangle \otimes \dots \otimes |\chi_{m_k}\rangle = 0$$

for

$$|\chi_{m_i}\rangle \propto T|\chi_{m_1}\rangle \otimes \dots \otimes |\chi_{m_{i-1}}\rangle \otimes |\chi_{m_{i+1}}\rangle \otimes \dots \otimes |\chi_{m_k}\rangle$$

The proportionality sign indicates that the state still has to be normalized resulting in a non-linear relation.

Proof. For the ease of notations, without loss of generality, let m_k be the selected variable node. For the construction of the transfer matrix it is convenient to select the variable m_k of constraint m . The input state now being $|\chi_{m_1}\rangle \otimes \dots \otimes |\chi_{m_{k-1}}\rangle$. Set $|\chi_{m_j}\rangle = \alpha_j|0\rangle + \beta_j|1\rangle$. In order to satisfy the constraint, we want

$$(20) \quad \langle\Phi|(\alpha_1|0\rangle + \beta_1|1\rangle) \otimes \dots \otimes (\alpha_k|0\rangle + \beta_k|1\rangle) = 0.$$

We expand this relation over the computational basis states $|i_1, \dots, i_k\rangle$, $i_j \in \{0, 1\}$. Defining

$$\gamma_j := \begin{cases} \alpha_j & \text{if } i_j = 0 \\ \beta_j & \text{if } i_j = 1 \end{cases}$$

we can express (20) as follows

$$(21) \quad \alpha_k \left[\sum_{i_1, \dots, i_{k-1} \in \{0,1\}} \gamma_1 \dots \gamma_{k-1} \langle\Phi|i_1 \dots i_{k-1}0\rangle \right] + \beta_k \left[\sum_{i_1, \dots, i_{k-1} \in \{0,1\}} \gamma_1 \dots \gamma_{k-1} \langle\Phi|i_1 \dots i_{k-1}1\rangle \right] = 0.$$

Therefore α_k and β_k can be found from the linear operation

$$(22) \quad \begin{bmatrix} \alpha_k \\ \beta_k \end{bmatrix} \propto \begin{bmatrix} \langle\Phi|0 \dots 01\rangle & \dots & \langle\Phi|1 \dots 11\rangle \\ -\langle\Phi|0 \dots 00\rangle & \dots & -\langle\Phi|1 \dots 10\rangle \end{bmatrix} \left(\begin{bmatrix} \alpha_1 \\ \beta_1 \end{bmatrix} \otimes \dots \otimes \begin{bmatrix} \alpha_{k-1} \\ \beta_{k-1} \end{bmatrix} \right),$$

and the resulting state can be normalized afterwards. The transfer matrix T has size $2 \times 2^{k-1}$ and is given by

$$(23) \quad T = \begin{bmatrix} \langle\Phi|0 \dots 01\rangle & \dots & \langle\Phi|1 \dots 11\rangle \\ -\langle\Phi|0 \dots 00\rangle & \dots & -\langle\Phi|1 \dots 10\rangle \end{bmatrix}$$

where the first (resp. second) row contains all 2^{k-1} ‘binary sequences’ of the form $|i_1, \dots, i_{k-1}, 1\rangle$ (resp. $|i_1, \dots, i_{k-1}, 0\rangle$). \square

We are now ready to explain the Algorithm (given in Table 1) behind the proof of 2.1. Let D be the set of pairs $\{v, a_v\}$ of variables v of degree one removed in LR together with its unique adjacent clause a_v . We order D chronologically, i.e., we let

$$D = \{\{v_1, a_{v_1}\}, \{v_2, a_{v_2}\}, \dots, \{v_L, a_{v_L}\}\}$$

where v_i is the i -th removed degree-one variables. For $\alpha < \alpha_{\text{r}}(k)$, LR ends w.h.p. with an empty 2-core and Algorithm 1 is a ‘reconstruction procedure’ which yields a product state solution $|\Psi\rangle$. Without loss of generality, we can assume that the initial graph G is connected, i.e., G has no isolated variable nodes (as we can always assign an arbitrary qubit state to isolated variable nodes if they are present). In a nutshell, starting from the last deleted check node, Algorithm 1 recursively assigns values to the set of variables connected to a clause using the transfer matrix T of Lemma A.2. We use the notation, T^{a_v} for the matrix T corresponding to the projector $|\Phi^{a_v}\rangle\langle\Phi^{a_v}|$ associated to clause a_v . We note that when a variable node w connected to a_v is already revealed in step 3 of the algorithm, we only reveal the edge connecting w and a_v .

Algorithm 1: Reconstruction Algorithm

Input: The ordered set D

Output: A product state $|\Psi\rangle = |\chi_1\rangle \otimes \dots \otimes |\chi_N\rangle \in \mathbb{C}^{\otimes N}$

```

1 begin
2   for  $i=N$  to  $1$  do
3     Reveal all  $k$  variables connected to  $a_{v_i}$ ;
4     for each variable  $w \neq v_i$  do
5       if  $w$  is not assigned any qubit then
6         | Assign an arbitrary qubit  $|\chi_w\rangle$  to  $w$  ;
7       else
8         | Set  $|\chi_w\rangle$  to be the qubit corresponding to  $w$  ;
9     | Set  $|\chi_{v_i}\rangle = T^{a_{v_i}} \cdot \prod_{w \neq v_i}^{\otimes} |\chi_w\rangle$ ;

```

Proof of Lemma 2.1. By Lemma A.1, for $\alpha < \alpha_{\text{r}}(k)$ the LR ends with an empty core w.h.p. so the set D contains all of the constraints nodes. This ensures that the outputted product state $|\Psi\rangle$ of Algorithm 1 has the correct length N , i.e., every variable has been assigned a qubit. Moreover, Lemma A.2 ensures that at each step of the algorithm, the projector $|\Phi^m\rangle\langle\Phi^m|$ is satisfied. While at the beginning of Algorithm 1 all the variables connected to the last constraint node in D have not been assigned any value yet, as the algorithm runs we need to make sure that the variable v connected to the check a_v has no qubits states attached to it yet (otherwise we would almost certainly get a contradiction when using the transfer matrix). This is indeed the case as shown in Claim A.3 below. Thus $|\Psi\rangle$ is a valid PRODSAT solution of the k -QSAT instance. Finally, as $M = \alpha N$ and there are at most k variables that are not assigned values for steps 3 to 8 of Algorithm 1, the complexity of Algorithm 1 is $O(N)$. \square

Claim A.3. For $\{v, a_v\} \in D$, the variable v is assigned a qubit only when the check a_v is revealed.

Proof. Suppose LR ends at a time $T \geq 0$ and suppose in the reconstruction Algorithm 1, we reveal a_v at some time $0 \leq t \leq T$. Hence, during the LR, v has degree one at time $T-t$. Now, assume that v would already be assigned a qubit state at time t . So, v must be connected to another clause b which was already revealed before a_v (during reconstruction), say at a time $s < t$. Thus, b must have been removed in LR at time $T-s > T-t$. Therefore, at time $T-t$, v is connected to both a_v and b and has degree at least 2, a contradiction. \square

Remark A.4. For $\alpha \geq \alpha_{\text{lr}}(k)$, LR ends with a non-empty core. As explained in the main text, if there is a zero energy product state $|\Psi'\rangle$ on the core, we can apply a reconstruction procedure similar to Algorithm 1 to recover the full product state $|\Psi\rangle$. Steps 5 and 6 of Algorithm 1 must be adapted so that when w belongs to the core then $|\chi_w\rangle$ is assigned the corresponding factor in $|\Psi'\rangle$ and the for loop in step 2 will run for a number $L < N$ steps as the set D does not contain all the constraints. This process can be used for $\alpha_{\text{lr}}(k) \leq \alpha < \alpha_{\text{dc}}(k)$.

APPENDIX B. MULTIVARIATE COMPLEX ANALYSIS RESULTS

To prove Proposition 3.1 we rely on results from complex analysis of multivariate functions. Before stating these results we need to define *isolated* and *simple* zeros of mappings $f : \mathbb{C}^n \rightarrow \mathbb{C}^n$.

Definition B.1. A vector \mathbf{a} is an *isolated zero* if it is the only solution of $f(\mathbf{z}) = \mathbf{0}$ in some small enough neighborhood of \mathbf{a} .

For example the map $f_1(z_1, z_2) = z_1^2(z_2 - 1)$, $f_2(z_1, z_2) = z_2^2(z_1 - 3)$ has two isolated zeros $(z_1, z_2) = (0, 0)$, $(z_1, z_2) = (3, 1)$. On the other hand the mapping $f_1(z_1, z_2) = z_1^2 + z_2^2$, $f_2(z_1, z_2) = z_1 + iz_2$ has a family of zeros $(z_1, z_2) = (\mu, i\mu)$ for any $\mu \in \mathbb{C}$, and none of these are isolated.

Definition B.2. A zero \mathbf{a} is *simple* if it is isolated and if Jacobian satisfies $\det J_f(\mathbf{a}) \neq 0$.

For the first example above one checks that $\det J_f(z_1, z_2) = 4z_1z_2(z_1 - 3)(z_2 - 1) - z_1^2z_2^2$. Thus $(0, 0)$ is not simple and $(3, 1)$ is simple. For the second example the zeros are not isolated and thus they are not simple. At the same time, the determinant of Jacobian vanishes. This is not a coincidence as one can show that a necessary condition to have $\det J_f(\mathbf{a}) \neq 0$ is that \mathbf{a} is isolated. This follows from the local inverse theorem B.6 below. This means that in the definition of a simple zero we can in fact drop the condition of being isolated.

Proposition B.3. [22, Proposition 2.1] Suppose that the mapping $\mathbf{z} \mapsto f(\mathbf{z})$ is holomorphic in a domain $D \subset \mathbb{C}^n$. Suppose the closure of a neighborhood $U_{\mathbf{a}} \subset D$ of a zero \mathbf{a} of the mapping does not contain other zeros (so \mathbf{a} is isolated). Then there exists $\varphi > 0$ such that for almost all $\zeta \in B(\mathbf{0}, \varphi)$ (w.r.t Lebesgue measure), the mapping $\mathbf{z} \mapsto f(\mathbf{z}) - \zeta$ has only simple zeros in $U_{\mathbf{a}}$. The number of simple zeros depends neither on ζ nor on the choice of the neighborhood $U_{\mathbf{a}}$.

Definition B.4. The *multiplicity* of the zero \mathbf{a} in the multivariate mapping $\mathbf{z} \mapsto f(\mathbf{z})$ is defined as the number of zeros in $U_{\mathbf{a}}$ of the perturbed mapping $\mathbf{z} \mapsto f(\mathbf{z}) - \zeta$.

Proposition B.5. [22, Proposition 2.2] The multiplicity of a simple zero is equal to 1.

Applying Propositions B.3, B.5 to the mapping $f_1(z_1, z_2) = z_1^2(z_2 - 1)$, $f_2(z_1, z_2) = z_2^2(z_1 - 3)$ we see that the simple zero $(3, 1)$ has multiplicity one since the perturbed mapping develops

“one branch” of simple zeros $(z_1, z_2) \approx (1 + \frac{\zeta_1}{9}, 3 + \zeta_2)$, whereas $(0, 0)$ has higher multiplicity as many branches of simple zeros $(z_1, z_2) \approx (\pm i\zeta_1^{1/2}, \pm \frac{i}{\sqrt{3}}\zeta_2^{1/2})$ appear.

In our application we need to show that $\mathbf{0}$ is an isolated zero of F_{sqr} . For this we rely on the following local inverse theorem:

Theorem B.6. [23, Theorem 5.5] *Consider an holomorphic map $f : U \subset \mathbb{C}^n \mapsto f(U) \subset \mathbb{C}^n$. Suppose $\mathbf{a} \in U$. Then f is biholomorphic in some small enough neighborhood of \mathbf{a} if and only if $\det J_f(\mathbf{a}) \neq 0$. (Biholomorphic means that the map $f : U \mapsto f(U)$ is a bijection and its inverse $f^{-1} : f(U) \mapsto U$ is also holomorphic.)*

APPENDIX C. DEFINITION AND PROPERTIES OF THE BUCHBERGER ALGORITHM

To describe the Buchberger algorithm, we need to define operations on multivariate polynomials. We start with a multivariate polynomial division.

C.1. Multivariate Division Algorithm. Let f be a multivariate polynomials in $K[X_1, \dots, X_n]$. We can write $f = \sum_{\alpha} a_{\alpha} x^{\alpha}$ with $\alpha = (\alpha_1, \dots, \alpha_n)$ in $\mathbb{Z}_{\geq 0}^n$ where α_i denotes the exponent of the i th variable such that x^{α} is the monomial $x^{\alpha} = X_1^{\alpha_1} \dots X_n^{\alpha_n}$ and the coefficients a_{α} are in K .

For univariate polynomials in X , the common ordering of the monomials is the degree ordering,

$$(24) \quad \dots > X^n > X^{n-1} > \dots > X > 1.$$

This notion can be extended to multivariate monomials.

Definition C.1 ([14]). *A monomial ordering $>$ on $K[X_1, \dots, X_n]$ is a relation $>$ on the set of monomial $x^{\alpha}, \alpha \in \mathbb{Z}_{\geq 0}^n$ satisfying*

- (1) $>$ is a total ordering on $\mathbb{Z}_{\geq 0}^n$
- (2) If $\alpha > \beta$ then for all $\gamma \in \mathbb{Z}_{\geq 0}^n$, $\alpha + \gamma > \beta + \gamma$.
- (3) $>$ is a well-ordering on $\mathbb{Z}_{\geq 0}^n$. This means that every nonempty subset of $\mathbb{Z}_{\geq 0}^n$ has a smallest element under $>$.

We also need some additional definitions to describe the multivariate division.

Definition C.2. *Let f, g be two multivariate polynomials in $K[X_1, \dots, X_n]$ and $<$ be a monomial ordering on $K[X_1, \dots, X_n]$.*

- The multidegree of f is $\text{multideg}(f) = \max(\alpha \in \mathbb{Z}_{\geq 0}^n | a_{\alpha} \neq 0)$ (the maximum is taken with respect to $<$).
- The leading coefficient of f is $LC(f) = a_{\text{multideg}(f)} \in K$.
- The leading monomial of f is $LM(f) = x^{\text{multideg}(f)}$.
- The leading term of f is $LT(f) = LC(f) \cdot LM(f)$.
- The least common multiple of the leading terms of f and g is denoted $LCM(f, g)$ and

$$LCM(f, g) = a_{\text{multideg}(f)} b_{\text{multideg}(g)} x^{\gamma}$$

with $\gamma_i = \max(\text{multideg}(f)_i, \text{multideg}(g)_i)$.

- The S -polynomial of f and g is

$$S(f, g) = \frac{LCM(f, g)}{LT(f)} \cdot f - \frac{LCM(f, g)}{LT(g)} \cdot g.$$

Theorem C.3 (Multivariate Division Algorithm 2 in $K[X_1, \dots, X_n]$ [14]). *Let $>$ be a monomial ordering on $\mathbb{Z}_{\geq 0}^n$, and let $F = (f_1, \dots, f_s)$ be an ordered s -tuple of polynomials in $K[X_1, \dots, X_n]$. Then every $p \in K[X_1, \dots, X_n]$ can be written as*

$$(25) \quad p = q_1 f_1 + \dots + q_s f_s + r$$

where $q_i, r \in K[X_1, \dots, X_n]$, and either $r = 0$ or r is a linear combination, with coefficients in K , of monomials, none of which is divisible by any of $LT(f_1), \dots, LT(f_s)$. We call r a remainder of p on division by F . Furthermore, if $q_i f_i \neq 0$, then

$$\text{multideg}(f) \geq \text{multideg}(q_i f_i).$$

Algorithm 2 makes it possible to compute such decomposition.

The r and q_i polynomials depend on the ordering of the (f_i) and on the monomial ordering. They are not uniquely characterized with the condition that r is not divisible by $LT(f_1), \dots, LT(f_s)$

Algorithm 2: Multivariate Division Algorithm

Input: $f_1, \dots, f_s, p \in K[X_1, \dots, X_n]$

Output: $q_1, \dots, q_s, r \in K[X_1, \dots, X_n]$

1 **begin**

2 $q_1 \leftarrow 0, \dots, q_s \leftarrow 0, r \leftarrow 0;$

3 **while** $p \neq 0$ **do**

4 division_occurred is false; $\triangleright p + f_1 q_1 + \dots + f_s q_s + r$ is a loop invariant

5 **while** $i < s$ and not division_occurred **do**

6 **if** $LT(f_i)$ divides $LT(p)$ **then**

7 $q_i \leftarrow q_i + LT(p)/LT(f_i);$

8 $p \leftarrow p - LT(p)/LT(f_i) \cdot f_i;$ \triangleright Remove the leading monomial

 division_occurred is true;

9 **if** not division_occurred **then**

10 $r \leftarrow r + LT(p);$

11 $p \leftarrow p - LT(p);$

Example C.4 (Multivariate Division). *Let's divide $p = xy^2 + 1$ by $(f_1 = xy + 1, f_2 = y + 1)$ using lexicographic ordering with $x > y$.*

(1) $LT(p) = xy^2$ which is divisible by $LT(f_1) = xy$. Then q_1 is updated to $q_1 = y$ and p is updated to $p = xy^2 + 1 - xy^2 - y = -y + 1$.

(2) $LT(p) = -y$ which is only divisible by $LT(f_2) = y$. Then q_2 is updated to $q_2 = -1$ and p is updated to $p = -y + 1 + y + 1 = 2$.

(3) $LT(p) = 2$ which is neither divisible by $LT(f_1)$ nor $LT(f_2)$ so r is updated to $r = 2$ and $f = 0$. The algorithm ends with $q_1 = y, q_2 = 1, r = 2$.

If the division is performed by changing the order of the polynomial $(f_1 = y + 1, f_2 = xy + 1)$, it gives another reminder:

(1) $LT(p) = xy^2$ which is divisible by $LT(f_1) = y$. Then q_1 is updated to $q_1 = xy$ and p is updated to $p = xy^2 + 1 - xy^2 - xy = -xy + 1$.

- (2) $LT(p) = -xy$ which is divisible by $LT(f_1)$. Then q_1 is updated to $q_1 = xy - x$ and p is updated to $p = -xy + 1 + xy + x = x + 1$.
- (3) $LT(p) = x$ which is neither divisible by $LT(f_1)$ nor $LT(f_2)$ so r is updated to $r = x + 1$ and $f = 0$. The algorithm ends with $q_1 = xy - x, q_2 = 0, r = x + 1$.

C.2. Gröbner basis and Buchberger Algorithm. An ideal I of a ring R is an additive subgroup of the ring such that for all $x \in I, r \in R$, we have $rx \in I$. The smallest ideal generated by a set S of elements in R is denoted as $\langle S \rangle = \{rx | r \in R \text{ and } x \in S\}$. S is called a basis of $\langle S \rangle$.

Definition C.5 (Gröbner basis). Given an ideal I of $K[X_1, \dots, X_n]$, $<$ a monomial ordering for $K[X_1, \dots, X_n]$ and a finite subset $G \subset I$, we say G is a Gröbner basis for the ordering $<$ if

$$\langle LT(G) \rangle = \langle LT(I) \rangle.$$

Here $\langle LT(G) \rangle$ means the ideal generated by the leading terms of the polynomials in G .

Corollary C.6. For a fixed monomial order, every ideal I has a Gröbner basis. Furthermore, any Gröbner basis is a generating set of I .

Gröbner bases have nice properties. Running the multivariate division algorithm with a Gröbner basis will not change the remainder regardless of the chosen order of the polynomials in the basis. Indeed, if the Buchberger outputs two different remainders r and r' for the division of a polynomial f with a Gröbner basis $G = \{g_1, \dots, g_s\}$, then there exist $g, g' \in \langle G \rangle$ such that $f = g + r = g' + r'$. Thus $r - r' = g' - g \in \langle G \rangle$ so $LT(r - r') \in \langle LT(g_1), \dots, LT(g_s) \rangle$ by the definition of a Gröbner basis but from Theorem C.3 neither r nor r' has monomials divisible by any of the $LT(g_1), \dots, LT(g_s)$. This implies that $r = r'$.

The Buchberger algorithm (3) returns a Gröbner basis.

Algorithm 3: Buchberger Algorithm [14]

Input: $F = (f_1, \dots, f_s)$

Output: A Gröbner basis G

```

1 begin
2    $G \leftarrow F$  ;
3   repeat
4      $G' \leftarrow G$ ;
5     for each pair  $g_i, g_j \in G$  with  $i \neq j$  do
6        $S_{i,j} \leftarrow S(g_i, g_j)$ ;
7        $r \leftarrow$  Multivariate Division Algorithm ( $G, S_{i,j}$ );
8       if  $r \neq 0$  then
9          $G \leftarrow G \cup \{r\}$ ;
10  until  $G = G'$ ;

```

Example C.7. Let us construct the Gröbner basis for the two polynomials of Example C.4, ($f_1 = xy + 1, f_2 = y + 1$) with lexicographic ordering.

- (1) $S(f_1, f_2) = xy + x - xy - 1 = x - 1$ whose leading term is divisible neither by $LT(f_1)$ nor $LT(f_2)$. So, $f_3 = x - 1$ can be added to the basis.

- (2) $S(f_1, f_3) = xy + 1 - xy + y = y + 1 = f_2$. The remainder of the division is 0.
(3) $S(f_2, f_3) = xy + x - xy + y = x - y = f_1 + f_3$. The remainder of the division is 0.
All pairs have been examined and the algorithm ends.

One can verify that running the algorithm with (f_1, f_2, f_3) yields the same remainder regardless of the order of the polynomials.

Buchberger algorithm generates numerous intermediate polynomials with total degrees that can be pretty large. The basis output by Buchberger algorithm can be simplified.

Definition C.8 (Reduced Gröbner basis [14]). For an ideal $I \subseteq K[X_1, \dots, X_n]$, a finite subset $G \subset I$ is a reduced Gröbner basis of I for the order $<$ if

- $LC(g) = 1$ for all $g \in G$,
- for all $g \in G$ no monomials of g lie in $\langle LT(G \setminus \{g\}) \rangle$.

Theorem C.9 ([14]). Let $I \neq \{0\}$ be a polynomial ideal. Then, for a given monomial ordering, I has a reduced Gröbner basis and the reduced Gröbner basis is unique.

Example C.10. (f_2, f_3) is the reduced basis for (f_1, f_2, f_3) .

We can construct a reduced Gröbner basis for a non-zero ideal by applying Buchberger algorithm. Then by adjusting the constants of the obtained basis G to make all the leading coefficients equal to 1 and removing any g with $LT(g) \in \langle LT(G \setminus \{g\}) \rangle$ from G , we can obtain the reduced basis because for any removed g , the resulting set $G \setminus \{g\}$ is also a Gröbner basis.

Even using a reduced Gröbner basis, Buchberger algorithm takes a huge amount of storage. The degree of the polynomials in the reduced Gröbner basis is bounded by $2(d^2/2 + d)^{2^{n-2}}$ [24] where d is the total degree of the (f_i) for example $d = k$ for k -QSAT. Today Faugere's algorithms are the fastest algorithms to compute Gröbner basis **F4**, **F5**. They use the same principle as Buchberger algorithm but use linear algebra to evaluate several pairs of polynomials and apply additional criteria to avoid evaluating S -polynomials that will reduce to 0.

C.3. Hilbert's Nullstellensatz. Thanks to Hilbert's Nullstellensatz, Buchberger algorithm can determine if a system of complex multivariate polynomial equations admits a common zero. Let us recall that an ideal I is a maximal ideal of a ring R if there are no other ideals contained between I and R , i.e. for any ideal J such that $I \subsetneq J$, $J = R$.

Theorem C.11 (Hilbert's Nullstellensatz (Zeros Theorem)). Let K be an algebraically closed field. Then every maximal ideal in the polynomial ring $K[X_1, \dots, X_n]$ has the form $(X_1 - a_1, \dots, X_n - a_n)$ for some $a_1, \dots, a_n \in K$.

Corollary C.12. As a consequence, a family of polynomials in $K[X_1, \dots, X_n]$ with no common zeros generates the unit ideal.

Proof. Let I be an ideal generated by $f_1, \dots, f_m \in K[X_1, \dots, X_n]$ with no common zeros. If I is contained in a maximal ideal $M = (X_1 - a_1, \dots, X_n - a_n)$ by Theorem C.11, then $(a_1, \dots, a_n) \in K^n$ is a common root of elements of I , in contradiction with the hypothesis. Since I does not lie in any maximal ideal, it must be $K[X_1, \dots, X_n]$. \square

Remark C.13. Conversely, any family of polynomials f_1, \dots, f_m in $K[X_1, \dots, X_n]$ that generates the unit ideal has no common zeros. Indeed if $\langle f_1, \dots, f_m \rangle = K[X_1, \dots, X_n]$ there exist $g_1, \dots, g_m \in K[X_1, \dots, X_n]$ such that $g_1 f_1 + \dots + g_m f_m = 1$. If f_1, \dots, f_m have a common zero, it contradicts the equality.

Corollary C.14. *A set of polynomials in an algebraically closed field has no common zeros if and only if the reduced Gröbner basis is $\{1\}$.*

Proof. A set of polynomials has no common zeros if and only if it generates the unit ideal and 1 belongs to an ideal if and only if 1 belongs to the Gröbner basis of the ideal for any monomial ordering (because $LT(1) = 1$) and thus belongs to the reduced Gröbner basis. \square

From Corollary C.14, the Gröbner basis output by Buchberger algorithm on input f_1, \dots, f_m is reduced to $\{1\}$ if and only if f_1, \dots, f_m do not have a common zero.

APPENDIX D. EVALUATING THE DIMENSION OF THE KERNEL

In Table 6.1, we give recurrence relations for $\dim \ker H_F$ with a specific interaction graph. Here we give the proof of the recurrence relations for the loose chain and the cycle (when the projectors are separable) which are not found in the literature to the best of our knowledge. These recurrences in Lemmas D.1 and D.2 yield upper bounds since we prove them only for separable projectors. However, we observe with numerical tests that they coincide with the ‘generic’ values.

Lemma D.1. *For an instance of k -QSAT with m separable projectors, the dimension of the kernel space for the loose chain interaction graphs, satisfies the recurrence relation*

$$(26) \quad r_m = 2^{k-1}r_{m-1} - 2^{k-2}r_{m-2}.$$

The initial conditions are $r_1 = 7$, $r_2 = 24$.

The initial conditions for $m = 1, 2$ are found by considering a special case of the \mathbf{d} -nosegay described in [18] where $\mathbf{d} = (0, \dots, 0)$ for $m = 1$ and $\mathbf{d} = (1, 0, \dots, 0)$ for $m = 2$. The proof of the recurrence relation is based on the arguments similar to those in [18, Lemma 5].

Proof. If the projectors are separable, then we can find a basis of the Hilbert space to decompose the two projectors at the ends of the chain as $|\alpha_m\rangle \otimes |0\rangle^{\otimes k-1}$ and the interior projectors as $|\beta_m^0\rangle \otimes |\beta_m^1\rangle \otimes |0\rangle^{\otimes k-2}$ where $|\alpha_m\rangle$ and $|\beta_m^i\rangle, i \in \{0, 1\}$ are the states of the qubits that appear in two clauses.

We can construct a basis for the solution space of the form

$$(27) \quad |\mathbf{b}\rangle = \bigotimes_i |b_i\rangle \otimes |v_{\mathbf{b}}\rangle$$

where $|b_i\rangle$ is the state of a unary qubit (a qubit whose vertex is unary) and $|v_{\mathbf{b}}\rangle$ is the state of the remaining qubits. The solutions are constructed by satisfying the clauses from one end to the other of the loose chain.

The first clause, at the beginning of the chain, can be satisfied if one of its $k - 1$ unary qubits has a state equal to $|1\rangle$. In this situation, there are $2^{k-1} - 1$ possible ways to satisfy the first clause. The remaining degree-two qubit is left unassigned. What remains to satisfy is a loose chain with $m - 1$ clauses. The subspace satisfying the new pattern is of dimension r_{m-1} .

If the states of all the unary qubits of the first clause are set to $|0\rangle^{\otimes k-1}$, then the last qubit is constrained to satisfy the clause. The Schmidt decomposition of $|v_{\mathbf{b}}\rangle$ gives $|v_{\mathbf{b}}\rangle = |q_1\rangle \otimes |v_1\rangle + |q_2\rangle \otimes |v_2\rangle$. Since $|q_1\rangle$ is orthogonal to $|q_2\rangle$ and $\langle q_1|\alpha_1\rangle = 0 = \langle q_2|\alpha_1\rangle$ (because the first clause is satisfied), $|v_{\mathbf{b}}\rangle$ is separable and $|q_1\rangle = |q_2\rangle = |\alpha_1^\perp\rangle$. To extend this partial assignment to a full solution, we can apply the same method recursively for the interior

clauses where there are only $k - 2$ unary qubits. For the last exterior clause, the degree 2 qubit is fixed so the dimension is $2^{k-1} - 1$. This gives in total

$$(28) \quad r_m = (2^{k-1} - 1)r_{m-1} + (2^{k-2} - 1) \sum_{i=2}^{m-2} r_i + 2^{k-1} - 1.$$

To obtain the announced relation (32), a few algebraic manipulations are necessary. Applying the last result to r_{m-1} , we obtain

$$(29) \quad r_{m-1} - (2^{k-1} - 1)r_{m-2} = (2^{k-2} - 1) \sum_{i=2}^{m-3} r_i + 2^{k-1} - 1$$

and combining with (28), we find

$$(30) \quad \begin{aligned} r_m &= (2^{k-1} - 1)r_{m-1} + (2^{k-2} - 1)r_{m-2} + r_{m-1} - (2^{k-1} - 1)r_{m-2} \\ &= 2^{k-1}r_{m-1} + (2^{k-2} - 2^{k-1})r_{m-2} \\ &= 2^{k-1}r_{m-1} - 2^{k-2}r_{m-2}. \end{aligned}$$

□

Lemma D.2. *For an instance of k -QSAT with m separable projectors, the dimension of the kernel space for the loose cycle interaction graphs, satisfies the recurrence relation*

$$(31) \quad s_m = 2^{k-1}s_{m-1} - 2^{k-2}s_{m-2}.$$

The initial conditions are $s_2 = 12$, $s_3 = 40$.

It is more convenient to use the notation s_m (instead of r_m used in Table 6.1) because the proof will use the previous lemma. We also need the following lemma.

Lemma D.3. *For an instance of QSAT with $m + 1$ separable projectors whose interaction graph is a 2-qubit chain starting with a k -qubit clause and $k \geq 2$, the dimension of the kernel space satisfies the recurrence relation*

$$(32) \quad t_m = (2^{k-1} - 1)m + 2^k - 1.$$

Here m is the number of 2-qubit clauses.

Proof. Recall that the dimension of the kernel space for a chain of m 2-qubit clauses is $m + 2$ [9]. The qubits of the first clause are labeled from 1 to k starting with the qubit both in the chain and in the k -qubit clause. We can find a basis of the Hilbert space where the projector of the first clause can be decomposed into $|\phi\rangle \otimes |0\rangle^{\otimes k-2}$ where $|\phi\rangle$ is a state of the first two qubits. We can construct a basis for the solution space of the form

$$(33) \quad |\mathbf{b}\rangle = \bigotimes_{i=3}^k |b_i\rangle \otimes |v_{\mathbf{b}}\rangle$$

where $|b_i\rangle$ is the state of qubit q_i and $|v_{\mathbf{b}}\rangle$ is the state of the remaining qubits. If any of the $|b_i\rangle$ are $|1\rangle$, the first clause is satisfied. What remains to satisfy is a 2-qubit chain of m clauses. The unassigned qubit q_2 accounts for 2 degrees of freedom. There are $2^{k-2} - 1$ possibilities to satisfy the first clause this way. If the qubits q_2, \dots, q_k are assigned to $|0\rangle^{\otimes k-2}$, then it remains to satisfy a 2-qubit chain of $m + 1$ clauses. Summing all contributions we find

$$(34) \quad t_m = 2(2^{k-2} - 1)(m + 2) + (m + 3) = (2^{k-1} - 1)m + 2^k - 1.$$

□

Proof of Lemma D.2. We start by looking at the interaction graphs that are loose chains composed of m clauses of k -qubit and ending with p 2-qubit clauses. Let us denote $r_{m,p}$ the dimension of the kernel space of the instances with these interaction graphs. We can show the following recurrence relation over m

$$(35) \quad r_{m,p} = 2^{k-1}r_{m-1,p} - 2^{k-2}r_{m-2,p}$$

with the same argument of the proof of Lemma D.1 and with the initialization given by $r_{m,0} = m + 2$ (chain only composed of 2-qubit clauses) and $r_{m,1} = t_m$ from Lemma D.3.

Regarding the loose cycle, we remark that fixing the value of a unary qubit in a clause c breaks the cycle into a loose chain with $m - 1$ clauses if this assignment satisfies c . There are $2^{k-2} - 1$ ways to satisfy a clause with unary qubits. If, after assigning a value to all unary qubits in c , the clause is still unsatisfied, the resulting interaction graph is a loose cycle with one 2-qubit clause. We can repeat the procedure and assign unary qubits of the next clause $c + 1$ in the cycle. If clause $c + 1$ is satisfied, the new interaction graph is a loose chain ending with one 2-qubit clause. If $c + 1$ is unsatisfied, the cycle graph now contains two 2-qubit clauses. We can iterate this procedure. At step $1 \leq p \leq m$, if the clause $c + p$ is satisfied, the cycle is broken into a loose chain ending with $p - 1$ 2-qubit clauses and if the clause $c + p$ is unsatisfied the new interaction graph is a loose cycle with p 2-qubit clauses.

After m steps, if all unary qubits are assigned and do not satisfy any of the clauses, the interaction graph is a cycle composed of only 2-qubit clauses, and the dimension of the kernel space for this graph is 2 [9]. Summing all contributions gives

$$(36) \quad s_m = (2^{k-2} - 1) \sum_{i=0}^{m-1} r_{i,m-1-i} + 2.$$

With some algebraic manipulations, we can obtain the desired relation, i.e,

$$(37) \quad s_m = (2^{k-2} - 1) \sum_{i=2}^{m-1} r_{i,m-1-i} + (2^{k-2} - 1)(r_{0,m-1} + r_{1,m-2}) + 2$$

$$(38) \quad = 2^{k-1}s_{m-1} - 2^{k-2}s_{m-2} + (2^{k-2} - 1)(r_{0,m-1} + r_{1,m-2} - 2^{k-1}r_{0,m-2}) - 2^{k-1} + 2$$

$$(39) \quad = 2^{k-1}s_{m-1} - 2^{k-2}s_{m-2}.$$

Equation (37) is obtained by applying (35) to each $r_{i,m-1-i}$. In (38), we bring out s_{m-1} and s_{m-2} . Finally, we replace in (38) $r_{0,m-1}$, $r_{0,m-2}$ and $r_{1,m-2}$ by their algebraic expressions to obtain (39). □

REFERENCES

- [1] S. Bravyi, “Efficient algorithm for a quantum analogue of 2-sat,” *Contemporary Mathematics*, vol. 536, pp. 33–48, 2011.
- [2] S. A. Cook, “The complexity of theorem-proving procedures,” in *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, ser. STOC ’71, Shaker Heights, Ohio, USA: Association for Computing Machinery, 1971, pp. 151–158, ISBN: 9781450374644. DOI: 10.1145/800157.805047. [Online]. Available: <https://doi.org/10.1145/800157.805047>.
- [3] B. Trakhtenbrot, “A survey of russian approaches to perebor (brute-force searches) algorithms,” *Annals of the History of Computing*, vol. 6, no. 4, pp. 384–400, 1984. DOI: 10.1109/MAHC.1984.10036.
- [4] D. Gosset and D. Nagaj, “Quantum 3-sat is qma₁-complete,” *SIAM Journal on Computing*, vol. 45, no. 3, pp. 1080–1128, 2016.
- [5] M. Mezard and A. Montanari, *Information, Physics, and Computation*. USA: Oxford University Press, Inc., 2009, ISBN: 019857083X.
- [6] F. Krzakala, A. Montanari, F. Ricci-Tersenghi, G. Semerjian, and L. Zdeborová, “Gibbs states and the set of solutions of random constraint satisfaction problems,” *Proceedings of the National Academy of Sciences*, vol. 104, no. 25, pp. 10 318–10 323, 2007.
- [7] J. Ding, A. Sly, and N. Sun, “Proof of the satisfiability conjecture for large k ,” *Annals of Mathematics*, vol. 196, no. 1, pp. 1–388, 2022. DOI: 10.4007/annals.2022.196.1.1. [Online]. Available: <https://doi.org/10.4007/annals.2022.196.1.1>.
- [8] O. Sattath, S. C. Morampudi, C. R. Laumann, and R. Moessner, “When a local hamiltonian must be frustration-free,” *Proceedings of the National Academy of Sciences*, vol. 113, no. 23, pp. 6433–6437, 2016.
- [9] C. R. Laumann, R. Moessner, A. Scardicchio, and S. L. Sondhi, “Random quantum satisfiability,” *Quantum Info. Comput.*, vol. 10, no. 1, pp. 1–15, Jan. 2010, ISSN: 1533-7146.
- [10] C. R. Laumann, A. M. Läuchli, R. Moessner, A. Scardicchio, and S. L. Sondhi, “Product, generic, and random generic quantum satisfiability,” *Phys. Rev. A*, vol. 81, p. 062 345, 6 Jun. 2010. DOI: 10.1103/PhysRevA.81.062345. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevA.81.062345>.
- [11] R. M. Karp and M. Sipser, “Maximum matching in sparse random graphs,” in *22nd Annual Symposium on Foundations of Computer Science (sfcs 1981)*, 1981, pp. 364–375. DOI: 10.1109/SFCS.1981.21.
- [12] C. Bordenave, M. Lelarge, and J. Salez, “Matchings on infinite graphs,” *Probability Theory and Related Fields*, vol. 157, no. 1, pp. 183–208, 2013.
- [13] R. C. Gunning and H. Rossi, *Analytic functions of several complex variables* (Prentice-Hall Series in Modern Analysis). Englewood Cliffs, New Jersey: Prentice-Hall, 1965.
- [14] D. A. Cox, J. Little, and D. O’Shea, *Ideals, Varieties, and Algorithms*. Springer, 2015.
- [15] P. Hall, “On representatives of subsets,” *Journal of the London Mathematical Society*, vol. s1-10, no. 1, pp. 26–30, 1935. DOI: <https://doi.org/10.1112/jlms/s1-10.37.26>.
- [16] F. S. Macaulay, “Some formulæ in elimination,” *Proceedings of The London Mathematical Society*, pp. 3–27, 1902.
- [17] D. A. Cox, J. Little, and D. O’Shea, *Using Algebraic geometry*. Springer, 2004.

- [18] S. Bravyi, C. Moore, and A. Russell, “Bounds on the quantum satisfiability threshold,” in *Innovations in Computer Science - ICS 2010, Tsinghua University, Beijing, China, January 5-7, 2010. Proceedings*, 2010, pp. 482–489.
- [19] D. N. Bernshtein, “The number of roots of a system of equations,” *Functional Analysis and Its Applications*, vol. 9, pp. 183–185, 1975. [Online]. Available: <https://api.semanticscholar.org/CorpusID:122772773>.
- [20] M. Dyer, P. Gritzmann, and A. Hufnagel, “On the complexity of computing mixed volumes,” *SIAM Journal on Computing*, vol. 27, no. 2, pp. 356–400, 1998. DOI: 10.1137/S0097539794278384. eprint: <https://doi.org/10.1137/S0097539794278384>. [Online]. Available: <https://doi.org/10.1137/S0097539794278384>.
- [21] M. Molloy, “Cores in random hypergraphs and boolean formulas,” *Random Structures & Algorithms*, vol. 27, no. 1, pp. 124–135, 2005.
- [22] A. Juzakov and L. A. Aizenberg, *Integral Representations and Residues in Multidimensional Complex Analysis*. AMS - Translations of Mathematical Monographs, 1983, p. 19.
- [23] C. Laurent-Thiébaud, *Holomorphic function theory in several variables: An introduction*. Springer Science & Business Media, 2010.
- [24] T. W. Dubé, “The structure of polynomial ideals and gröbner bases,” *SIAM Journal on Computing*, vol. 19, no. 4, pp. 750–773, 1990. DOI: 10.1137/0219053.